APPLICATION OF CLAIRMONT'S COLORING ALGORITHM TO THE ERRERA GRAPH

JOHN CLAIRMONT

Abstract. In a recent article I described a new alternating chain algorithm for four coloring planar graphs . Someone suggested that I test the algorithm on the Errera graph. Kempe Chains fail to four color the Errera graph but my algorithm had no problem four coloring the Errera graph.

Section 1. Introduction

In this algorithm only the main steps, though well defined, are given. The complete details of the algorithm remain to be fleshed out, but I don't think it will be difficult. For a more complete explanation see my previous paper John Clairmont, "TOWARD A NEW PROOF OF THE FOUR COLOR THEOREM," *viXra.org*, 2024, Two properties are defined Property A and property B, which are explained in the example solution of the Errera graph. And two subroutines are described subroutine 1 is applied when both properties A and B hold. Sometimes this will disrupt property B. In that case subroutine 2 is applied, so that property B is restored and both properties A and B hold, then subroutine 1 can be applied again.

Section 2. The Errera Graph

(Step through the example - don't scroll.) In the following a list of instructions is given to a computer program which then executes each instruction. Things start off slowly, The first alternating chain of length greater than one occurs at figure 83, the first example of property B failing, occurs at figure 85.

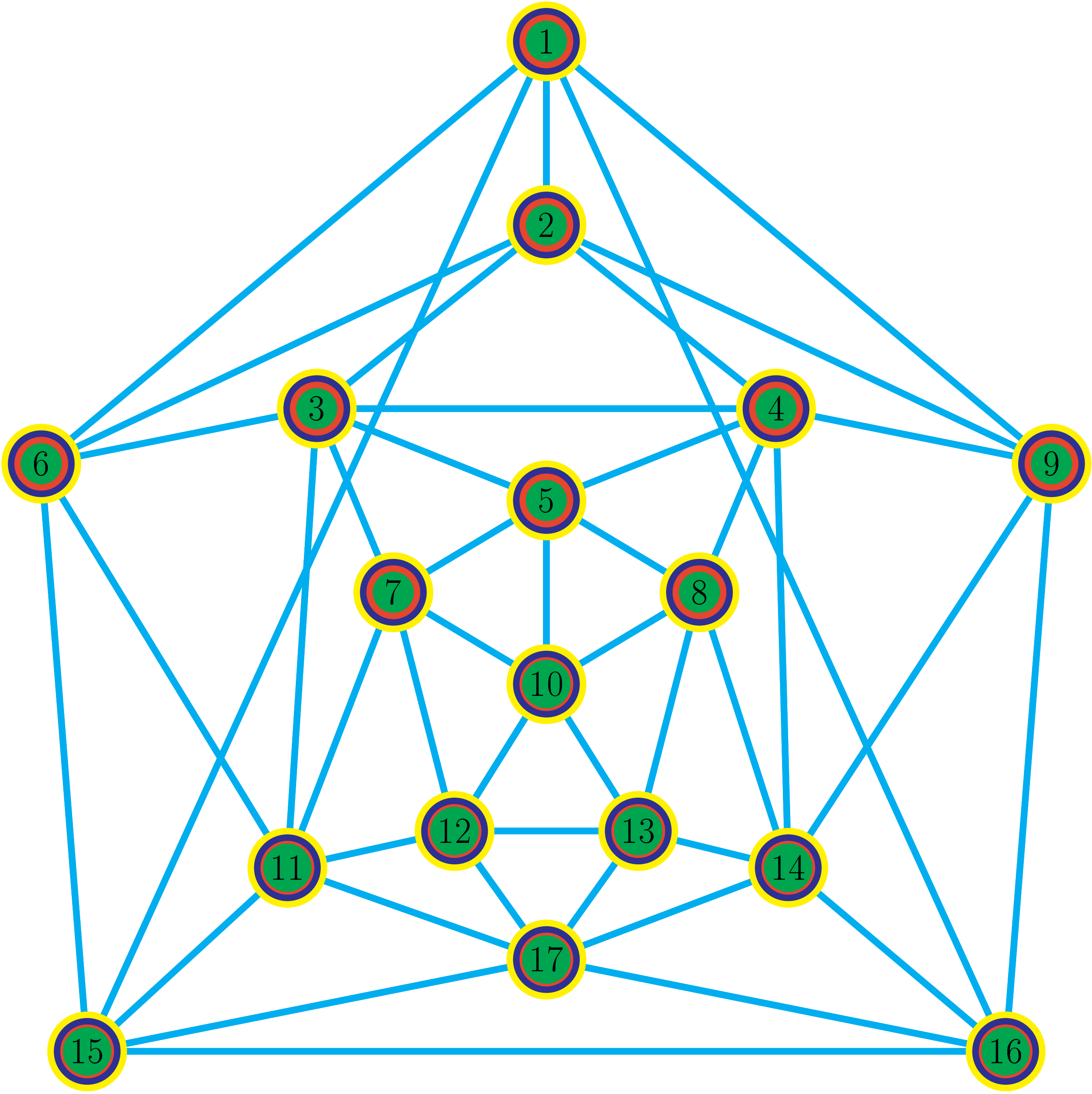


Figure 1. In the beginning a checker of each of the four colors Green, Red, Blue, and Yellow, is placed on every vertex.

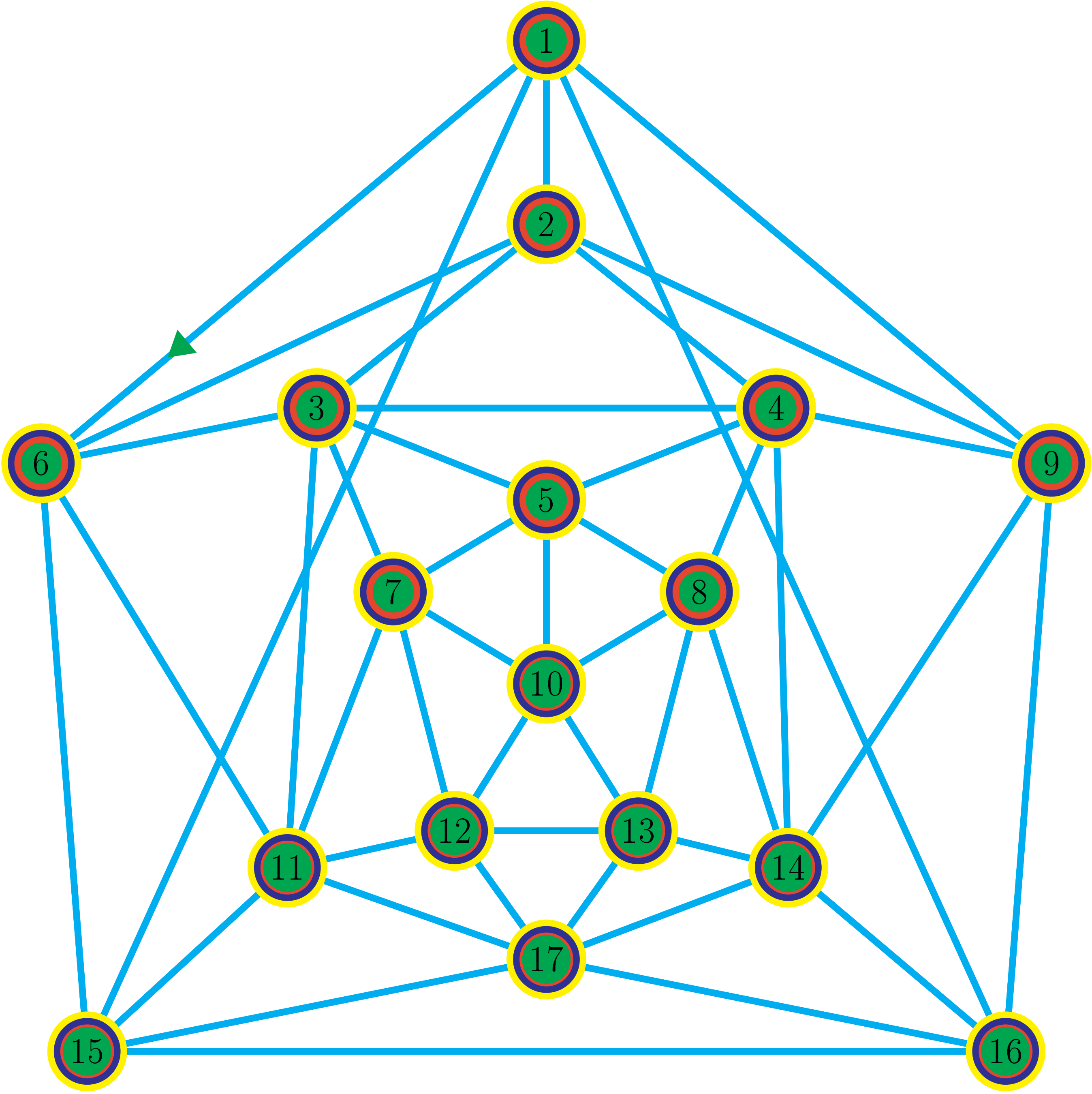


Figure 2.

instruction 1: place edge 1->6 Green DeletionArrow

When the above instruction is executed by the computer a triangular green arrowhead is placed on the directed edge 1→6. A triangular green arrowhead is used to indicate the deletion of the green checker on the vertex pointed to (vertex 6). We define triangle arrowheads as deletion arrows. Later we'll define insertion arrows (see FIGURE 82).

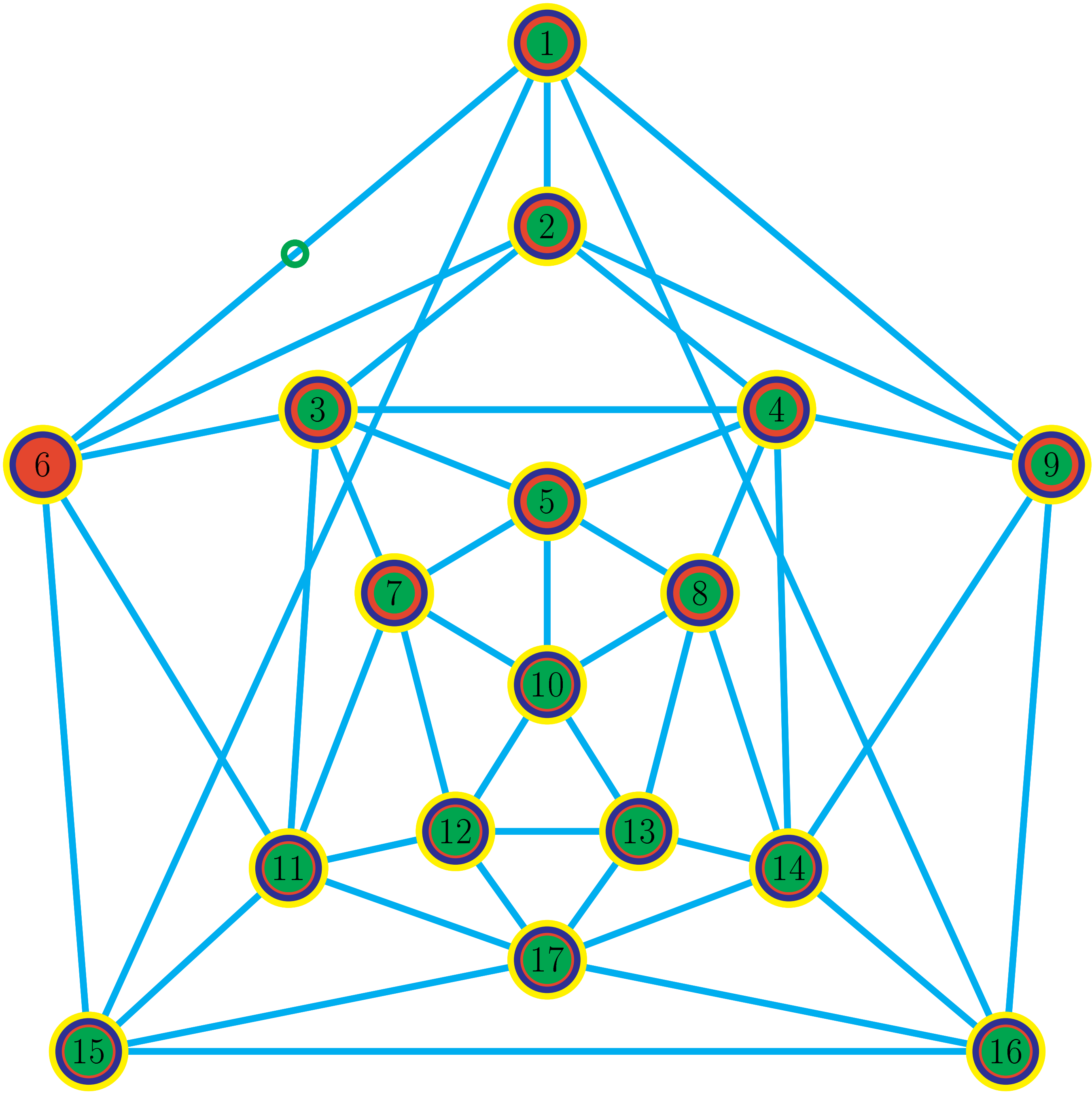


Figure 3.

instruction 2: unplace edge 1->6 Green DeletionArrow
instruction 3: place edge 1-6 Green Checker
instruction 4: unplace vertex 6 Green Checker;

When the above instructions are executed the arrowhead is removed, a Green Checker (circle) is placed(exactly in the middle) of the undirected edge 1-6 and the Green Checker is removed from vertex 6. Now the edge is conflict free wtr the color green ie there is only one green checker on vertices of this edge.

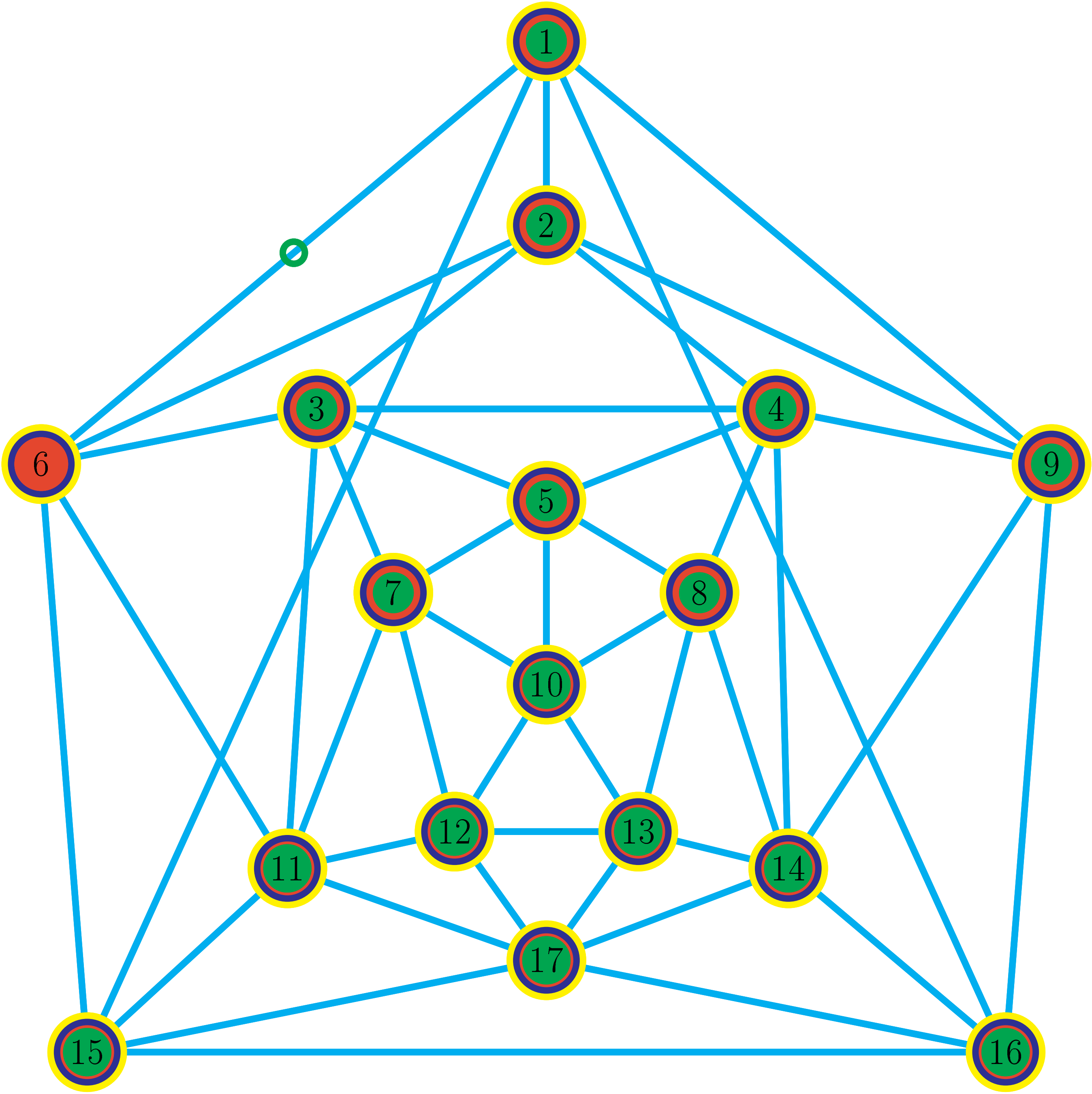


Figure 4.

Continuing in this manner we obtain a sequence of graphs with the following properties.

1: If a checker of a given color is on an edge there cannot be two checkers of the same color on vertices of that edge. Lets call this property A.

2: If a checker of a given color is on an edge there must be one and only one checker of that color on verices of that edge. Lets call this property B. See Figure 85 for an example.

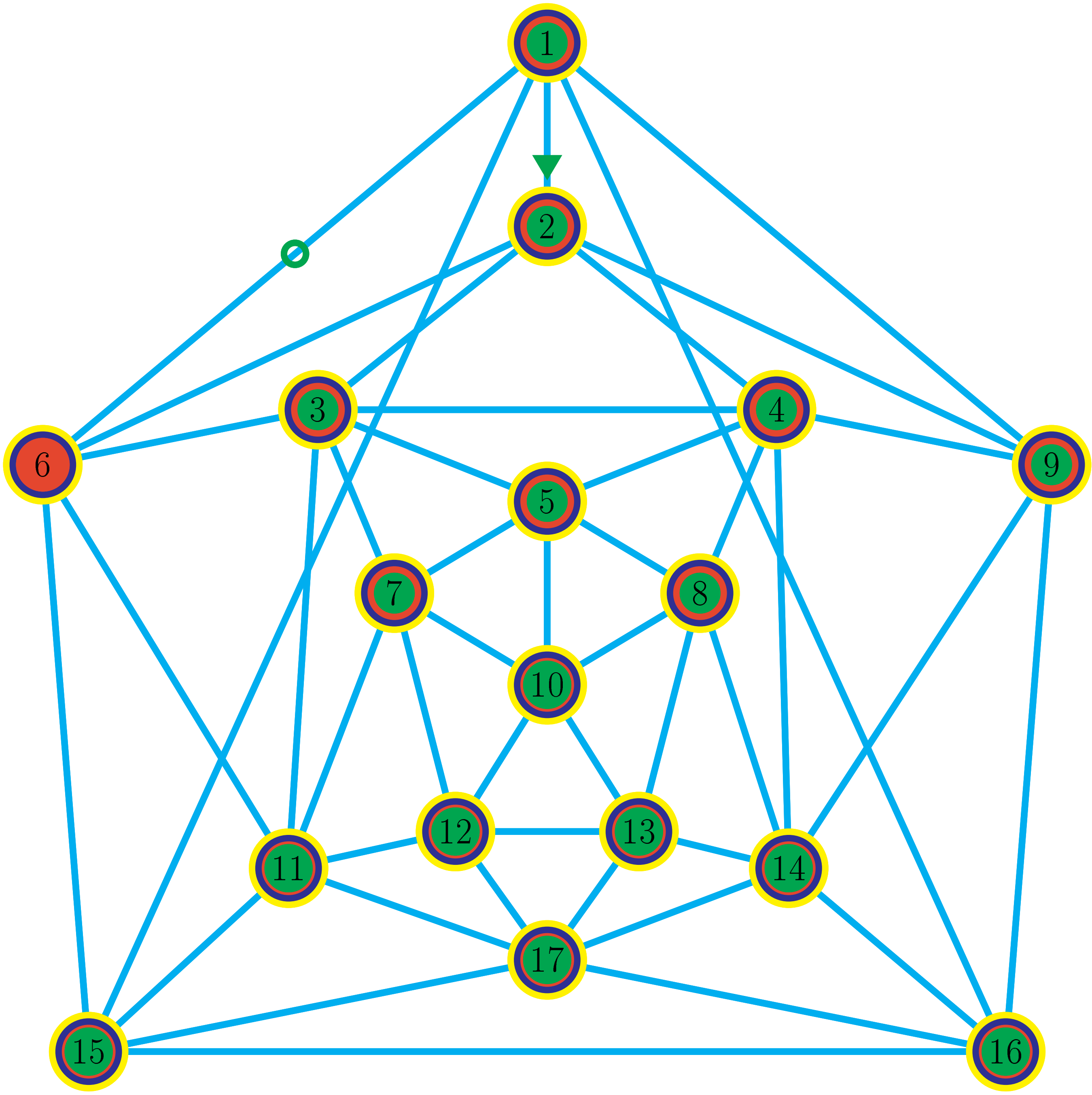


Figure 5.

instruction 5: place edge 1->2 Green DeletionArrow

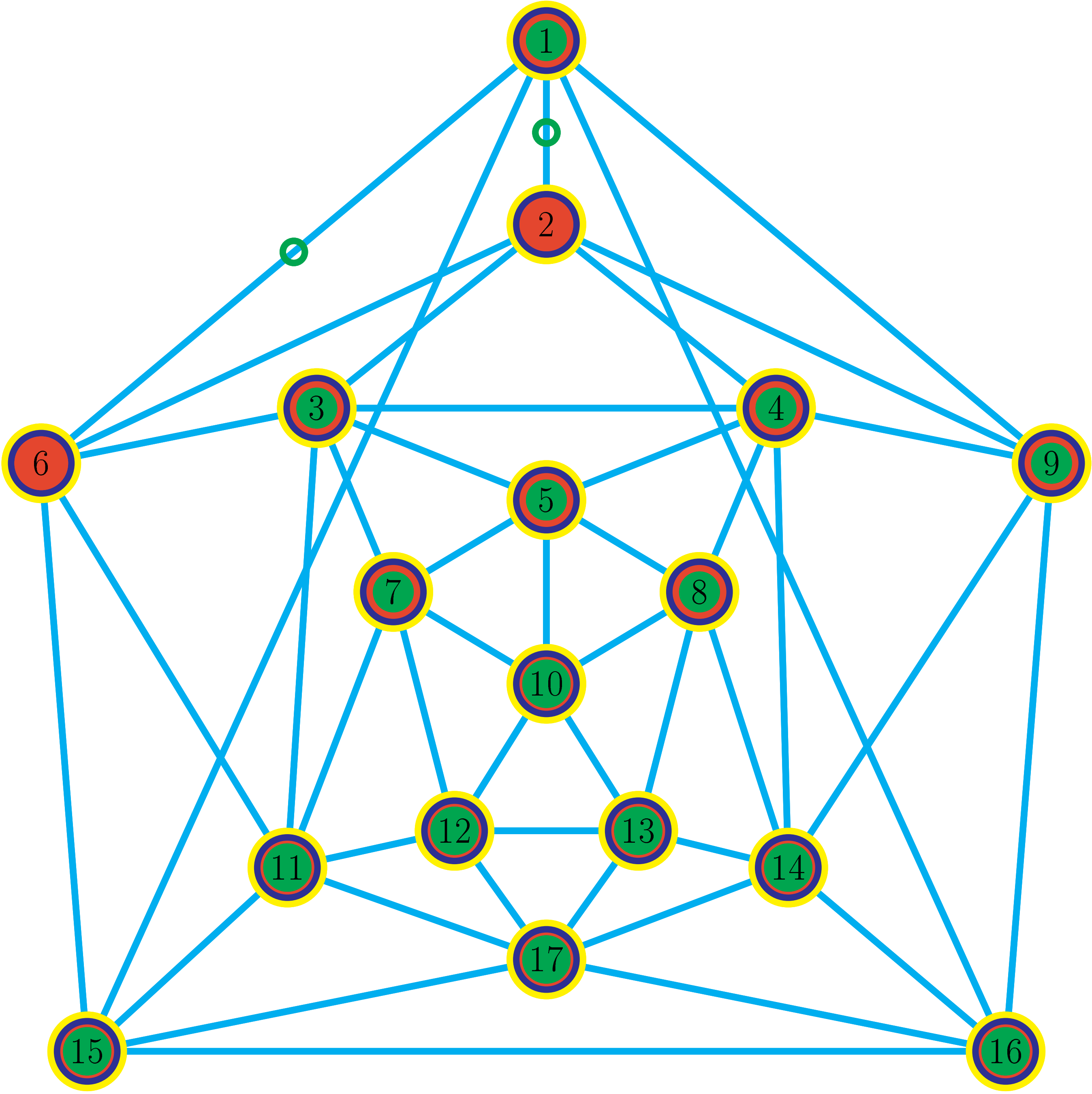


Figure 6.

instruction 6: unplace edge 1->2 Green DeletionArrow
instruction 7: place edge 1-2 Green Checker
instruction 8: unplace vertex 2 Green Checker;

The green checker has been removed from vertex 2 and a green checker has been placed on edge 1→2 and properties A and B are satisfied as they will be after each major iteration of the algorithm.

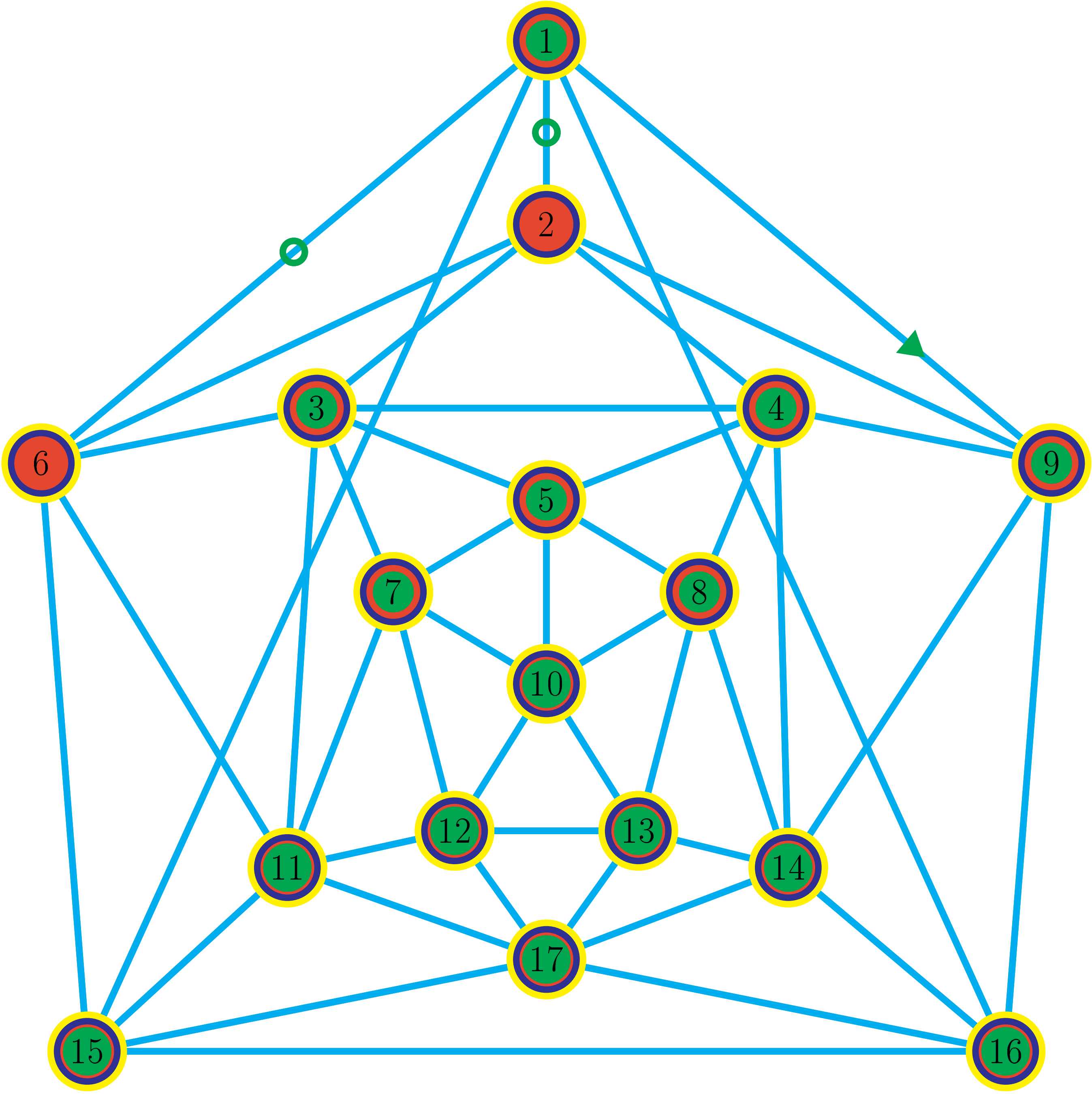


Figure 7.

instruction 9: place edge 1->9 Green DeletionArrow

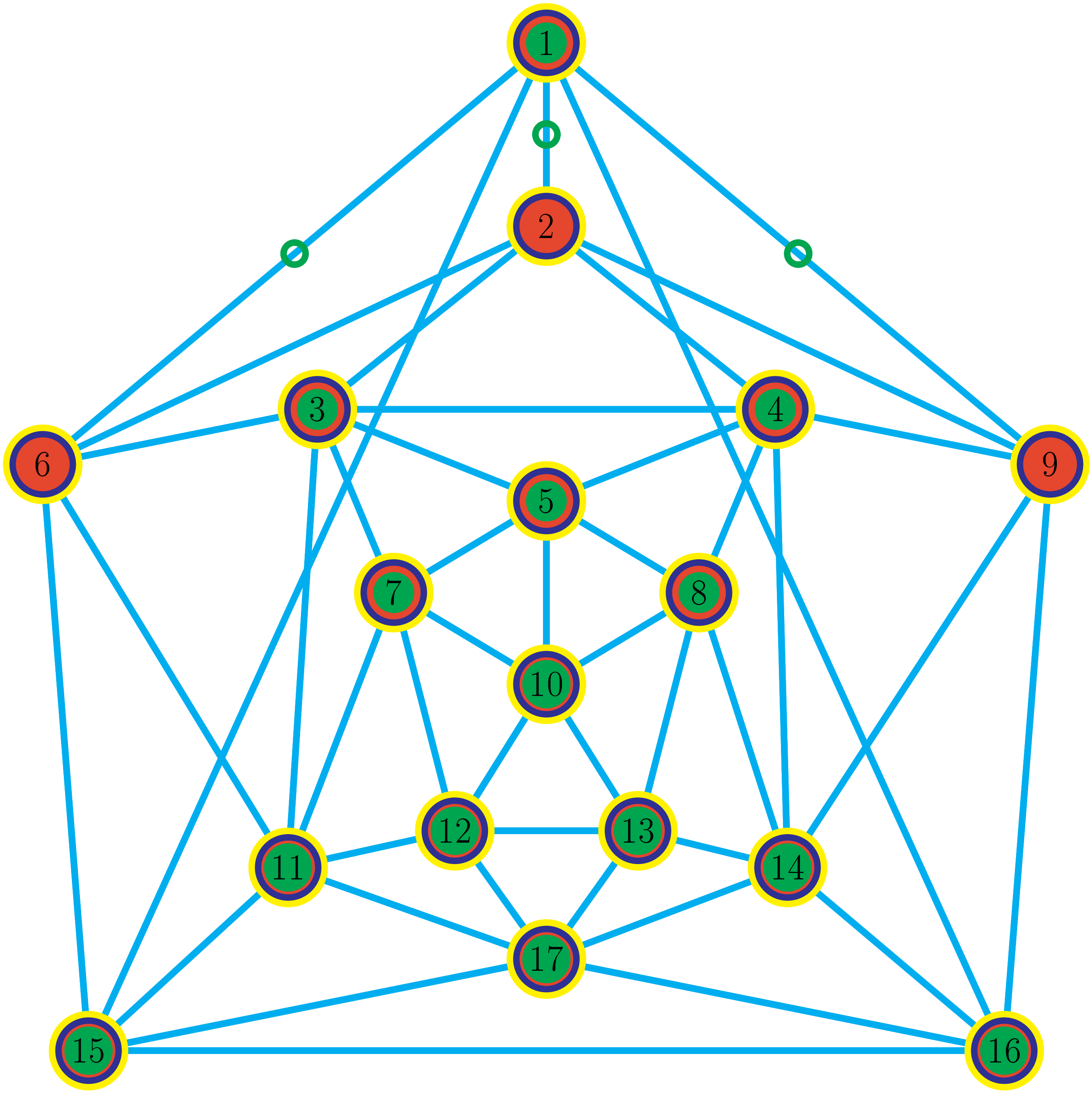


Figure 8.

instruction 10: unplace edge 1->9 Green DeletionArrow
instruction 11: place edge 1-9 Green Checker
instruction 12: unplace vertex 9 Green Checker;

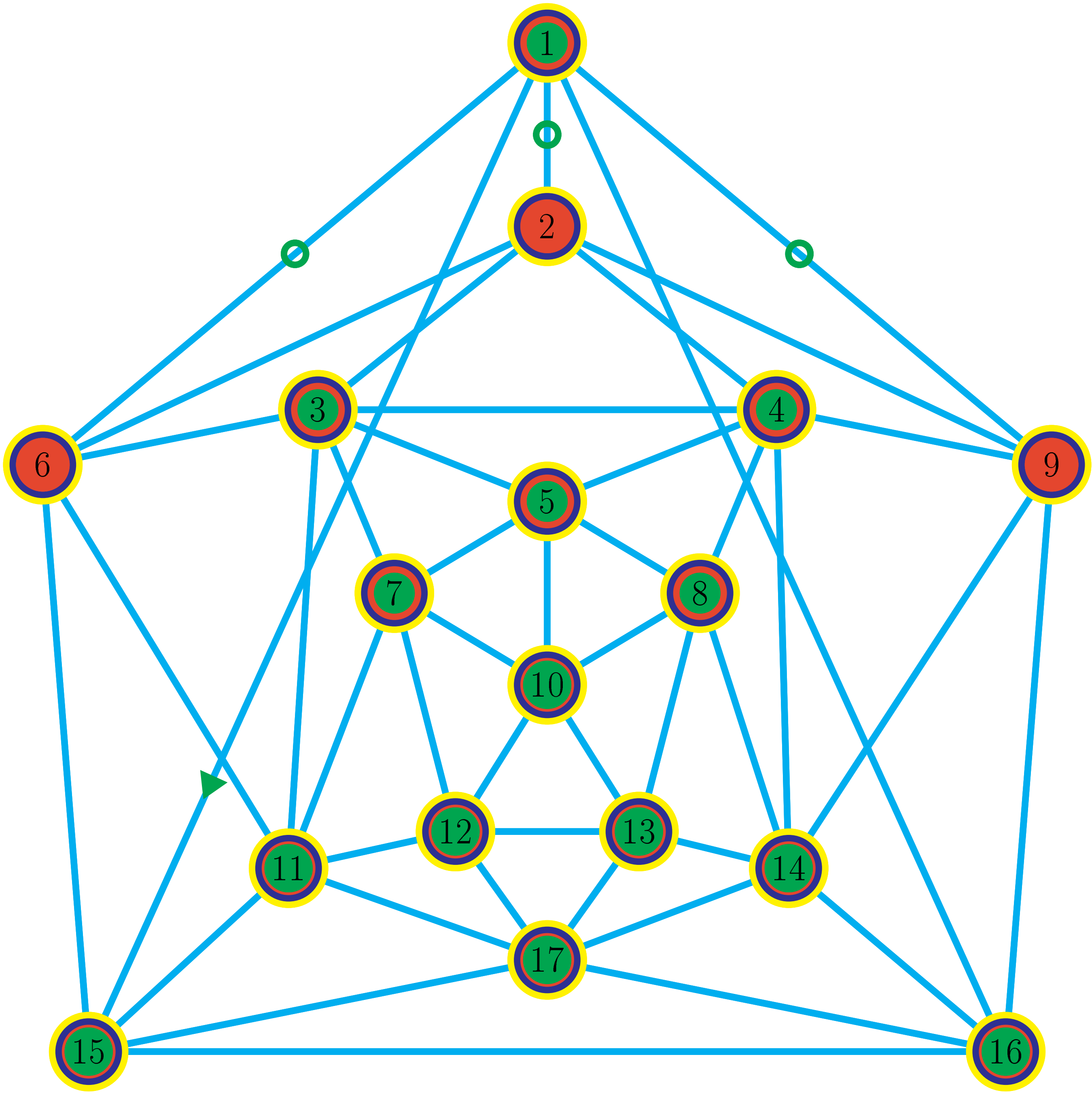


Figure 9.

instruction 13: place edge 1->15 Green DeletionArrow

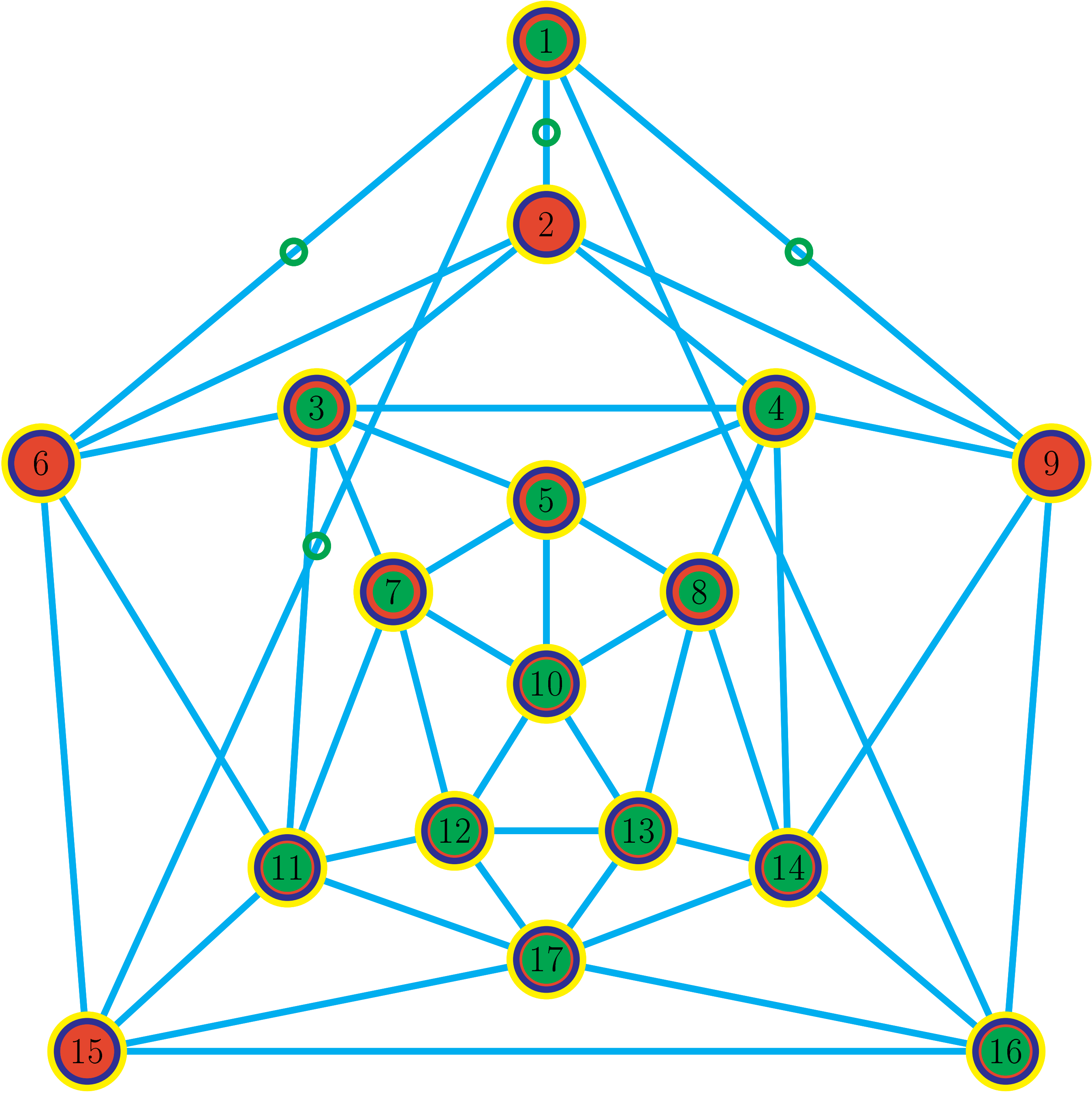


Figure 10.

instruction 14: unplace edge 1->15 Green DeletionArrow
instruction 15: place edge 1-15 Green Checker
instruction 16: unplace vertex 15 Green Checker;

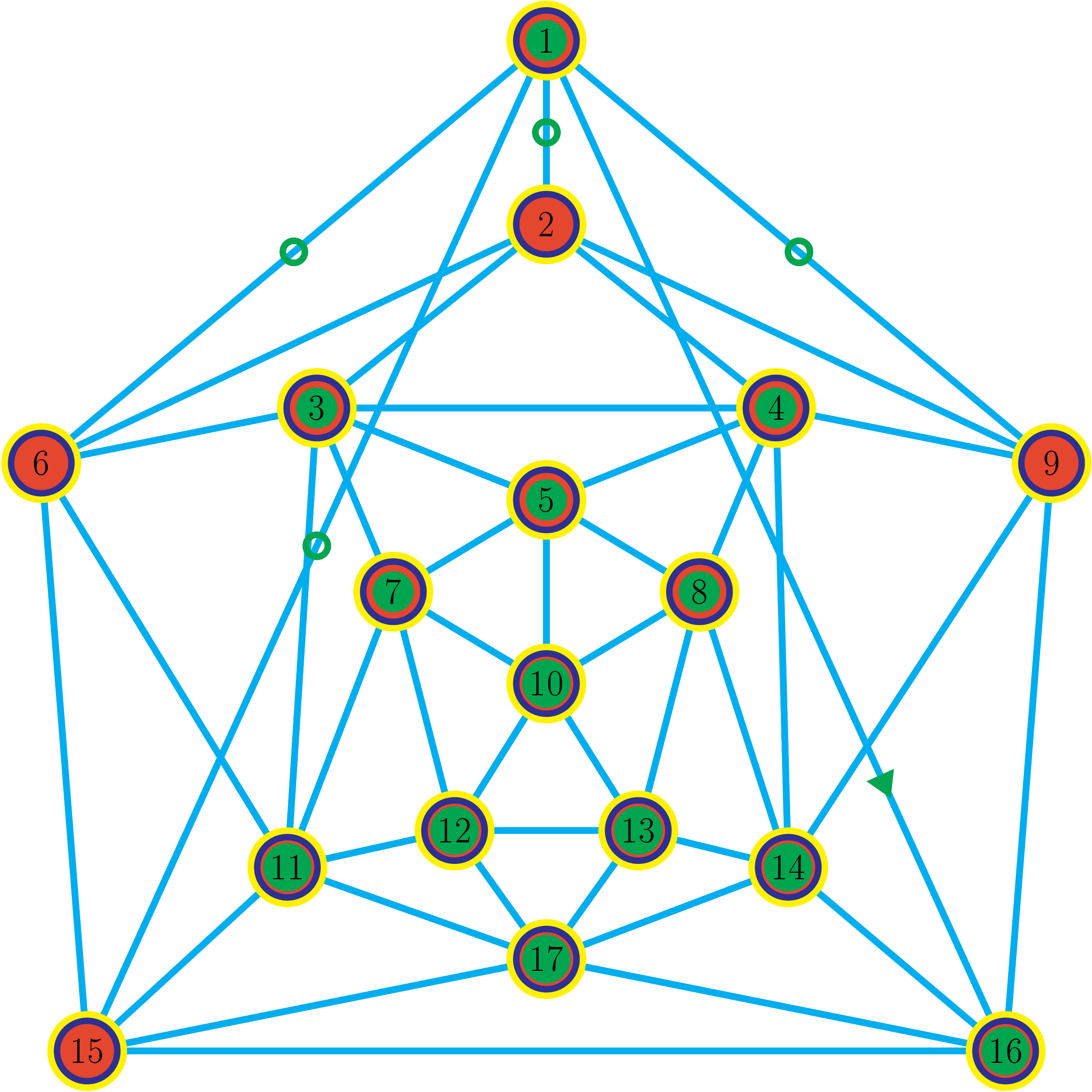


Figure 11.

instruction 17: place edge 1->16 Green DeletionArrow

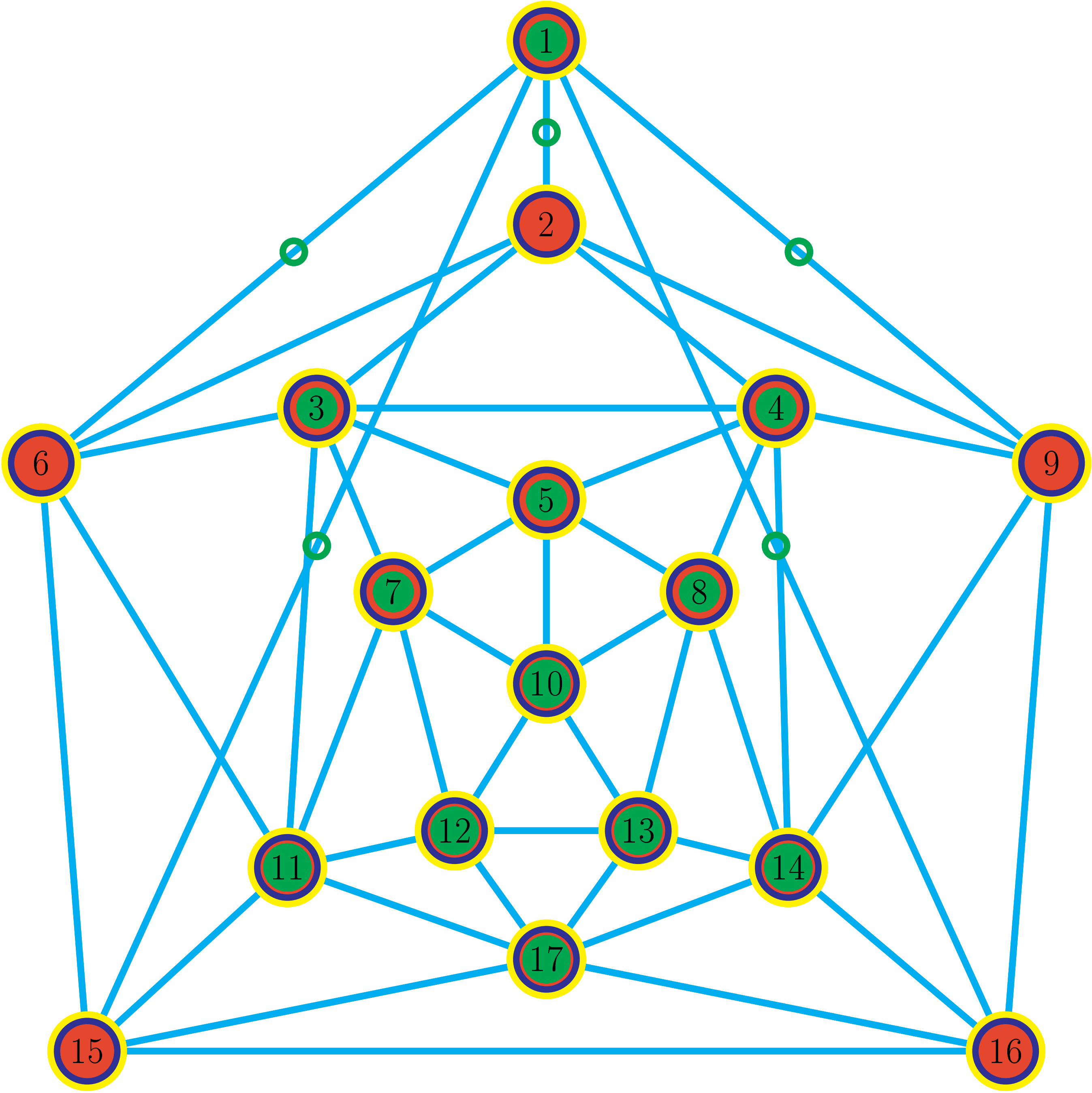


Figure 12.

instruction 18: unplace edge 1->16 Green DeletionArrow
instruction 19: place edge 1-16 Green Checker
instruction 20: unplace vertex 16 Green Checker;

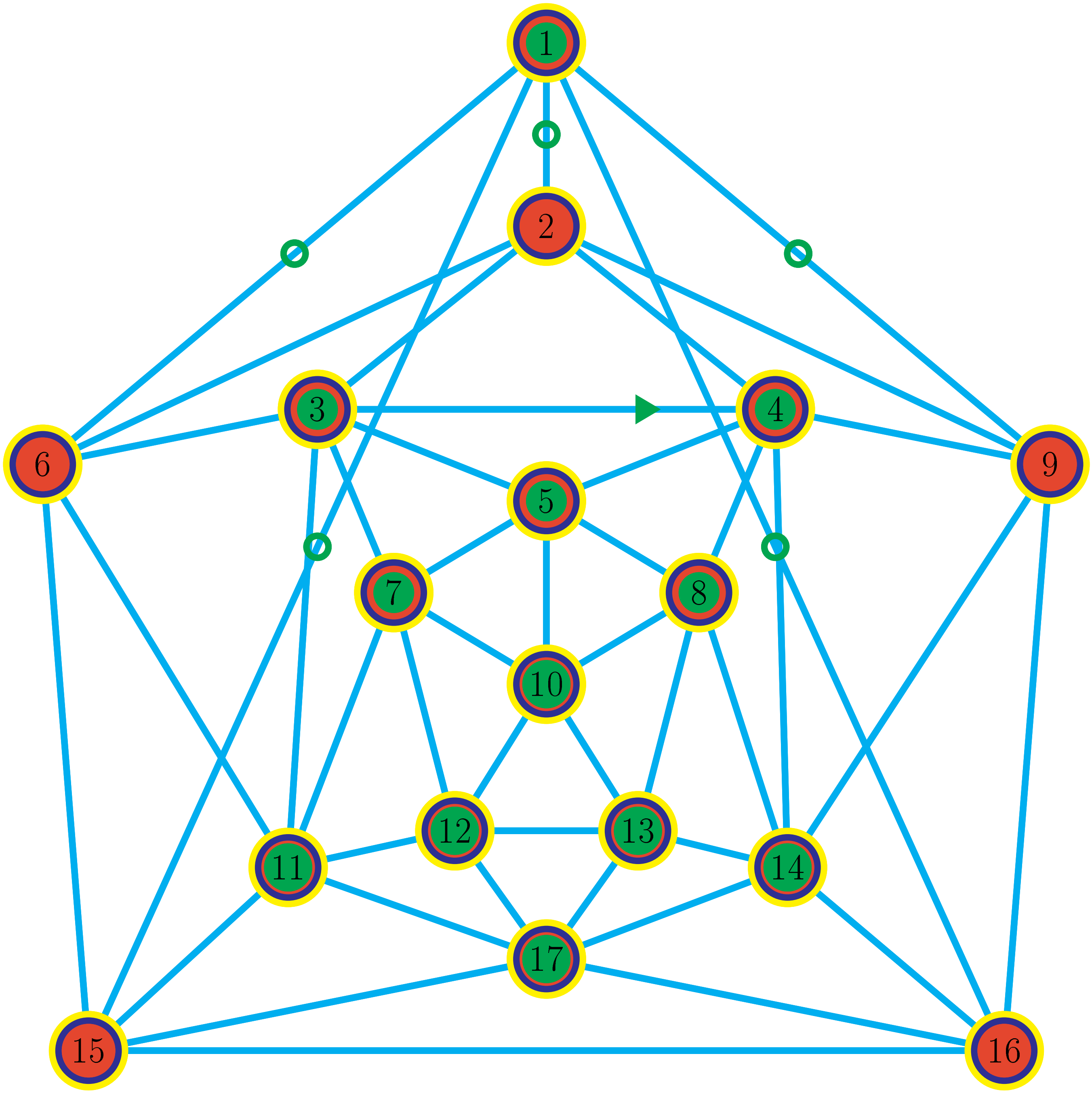


Figure 13.

instruction 21: place edge 3->4 Green DeletionArrow

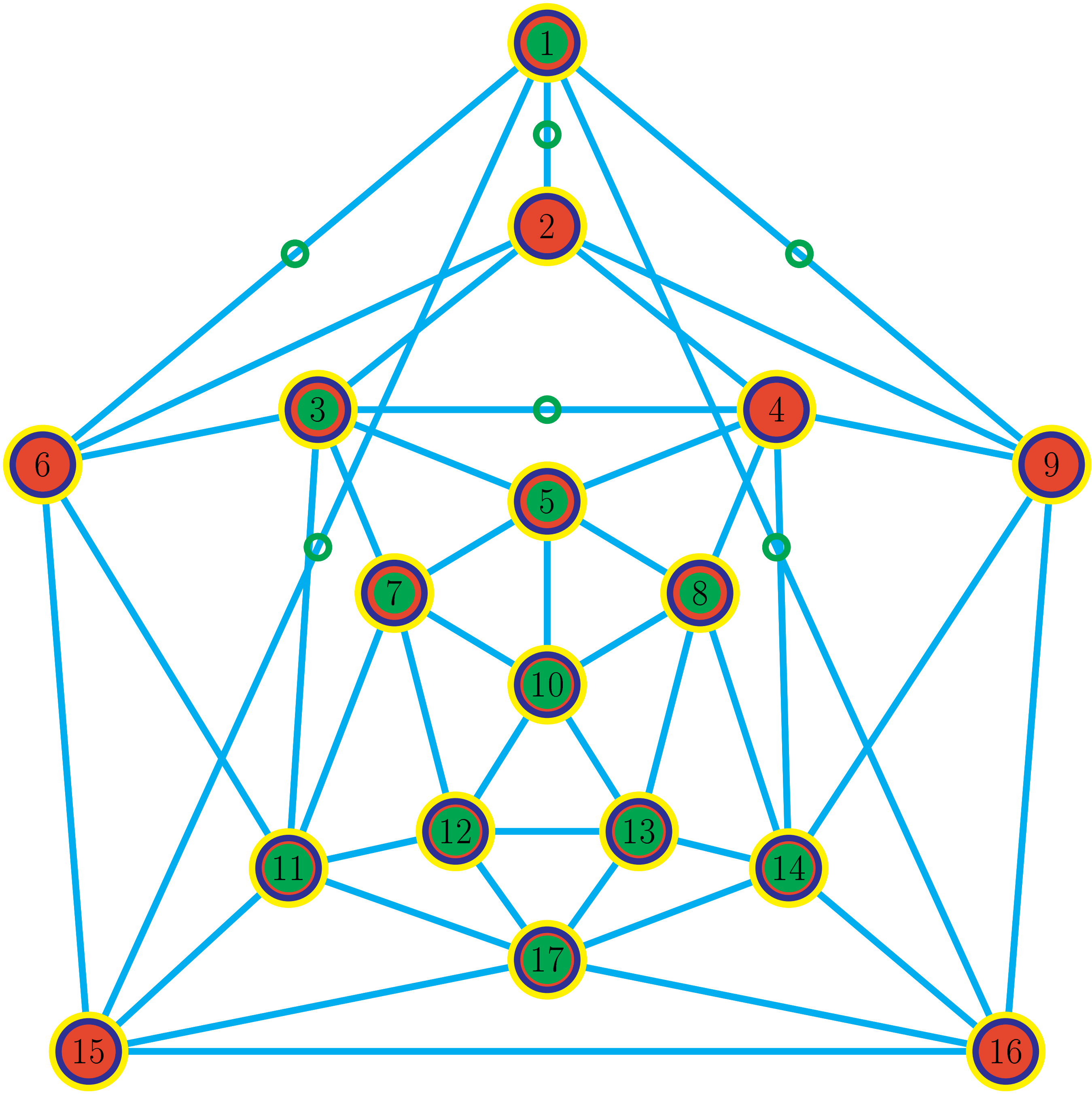


Figure 14.

instruction 22: unplace edge 3->4 Green DeletionArrow
instruction 23: place edge 3-4 Green Checker
instruction 24: unplace vertex 4 Green Checker;

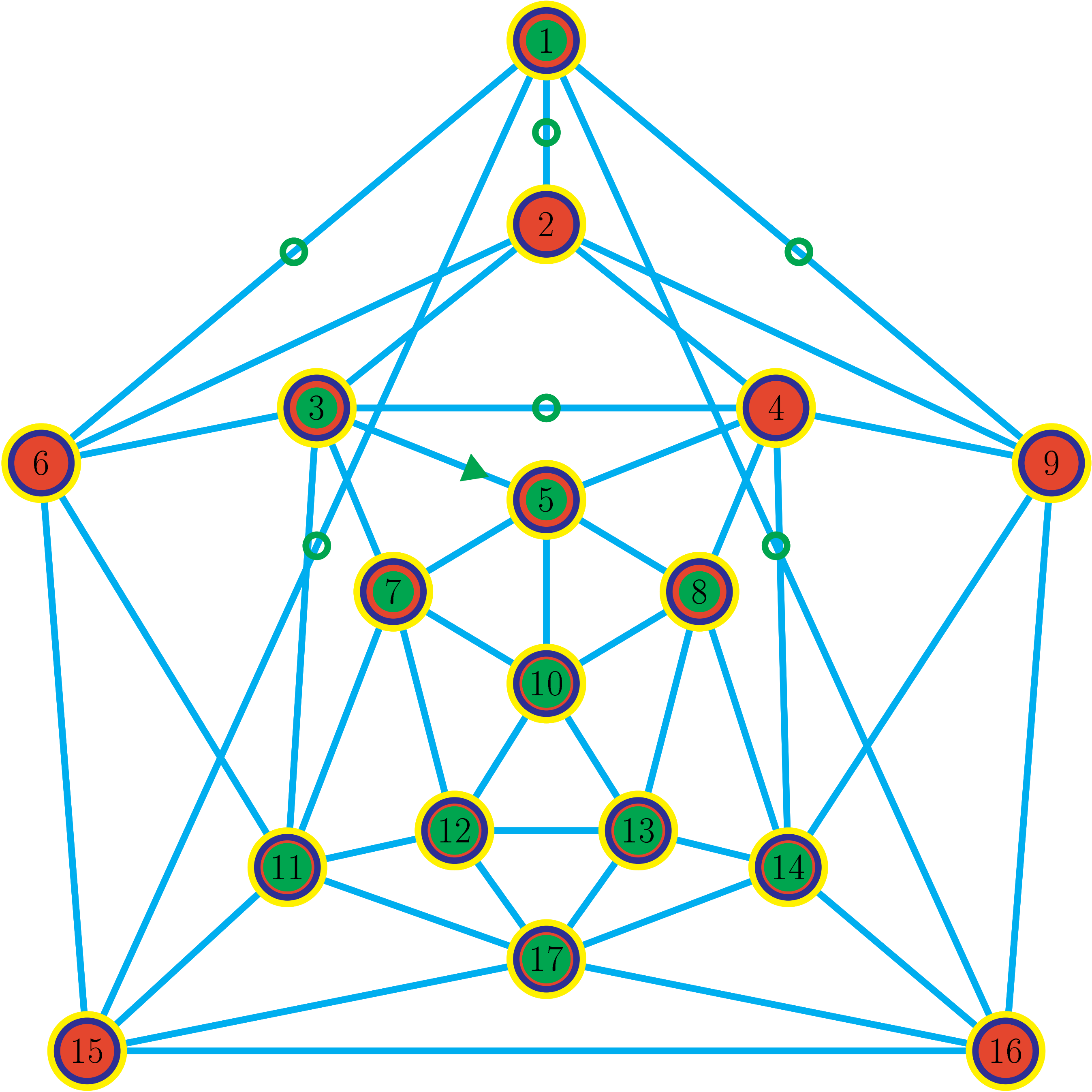


Figure 15.

instruction 25: place edge 3->5 Green DeletionArrow

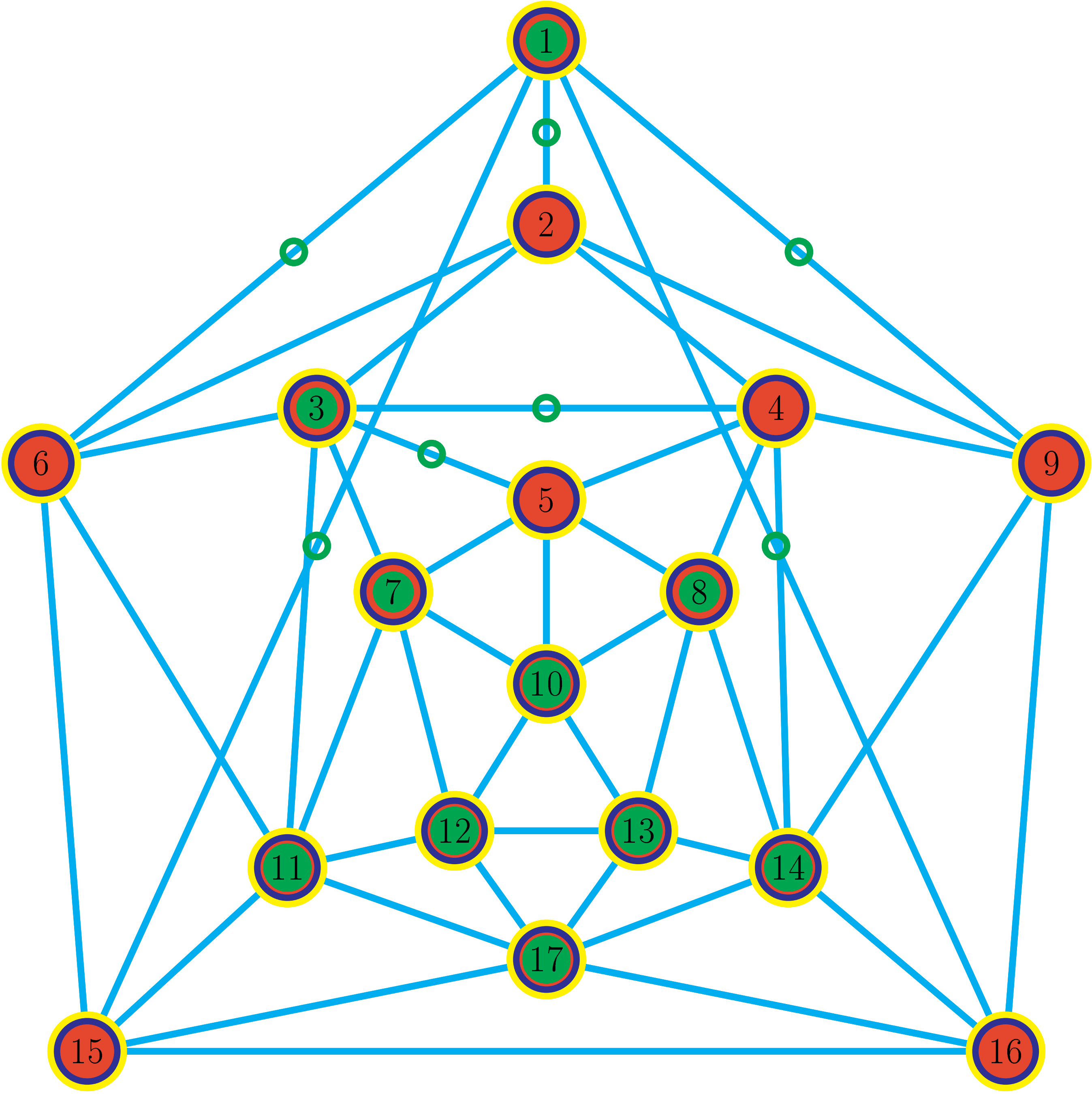


Figure 16.

instruction 26: unplace edge 3->5 Green DeletionArrow
instruction 27: place edge 3-5 Green Checker
instruction 28: unplace vertex 5 Green Checker;

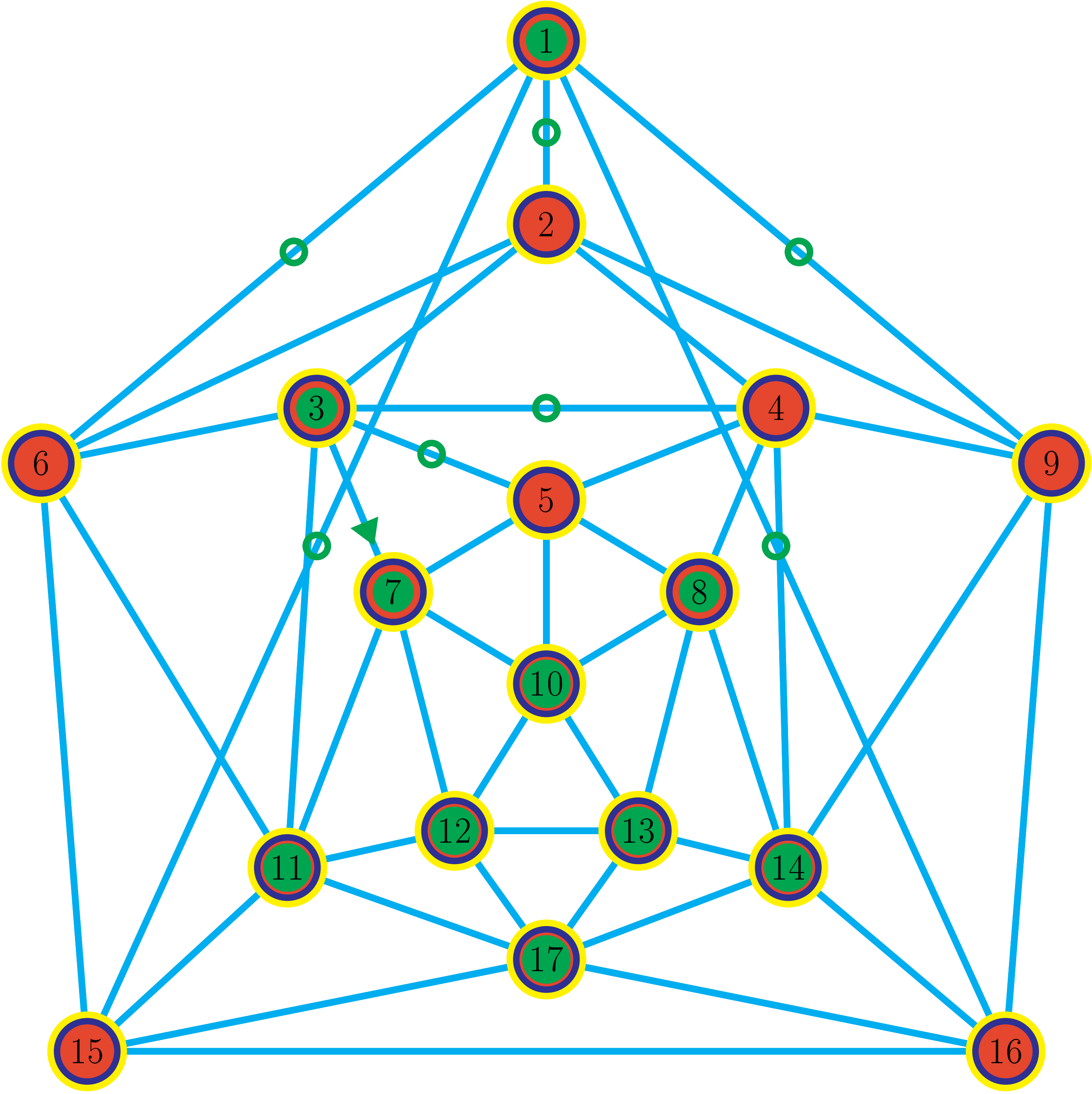


Figure 17.

instruction 29: place edge 3->7 Green DeletionArrow

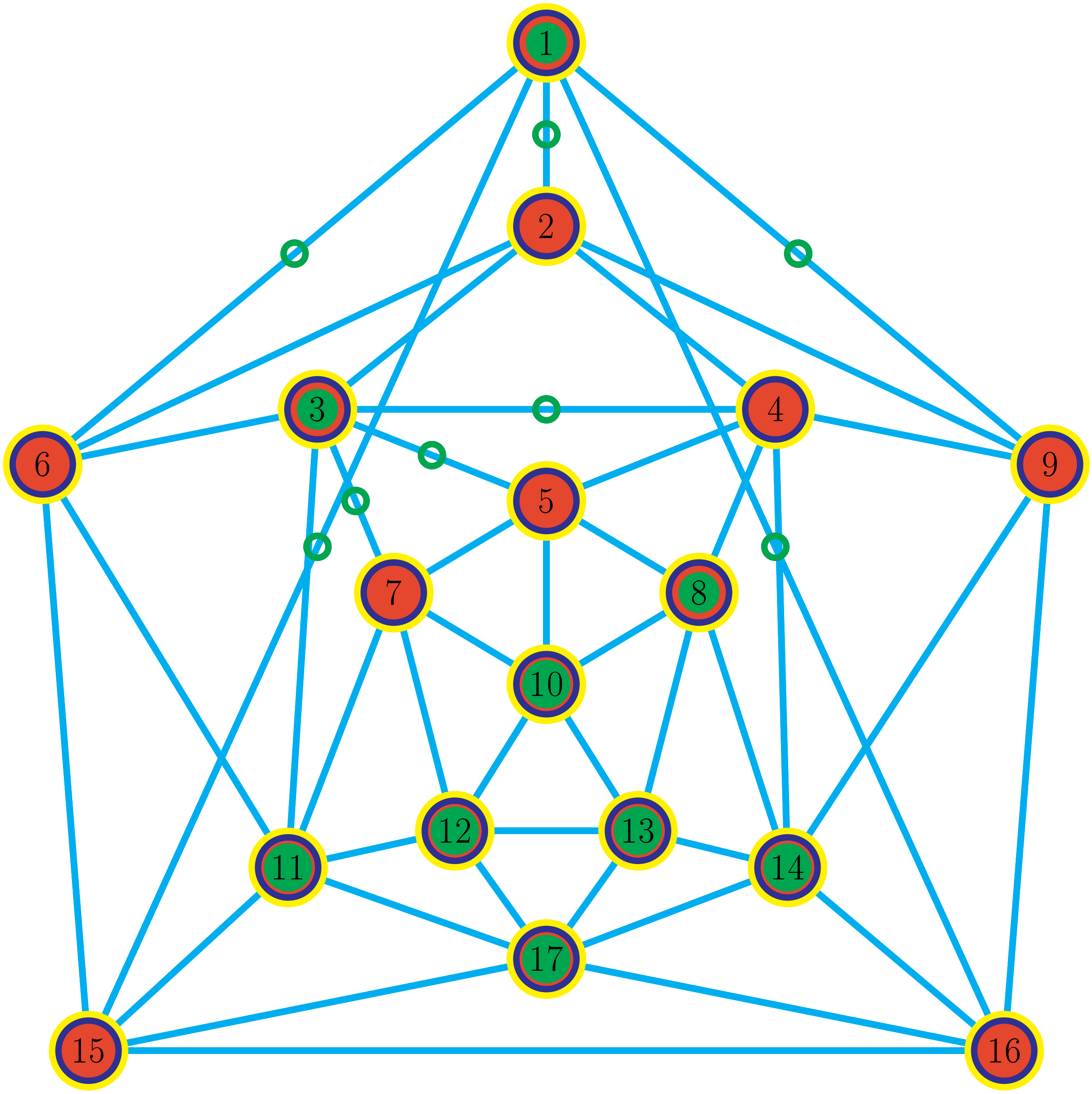


Figure 18.

instruction 30: unplace edge 3->7 Green DeletionArrow
instruction 31: place edge 3-7 Green Checker
instruction 32: unplace vertex 7 Green Checker;

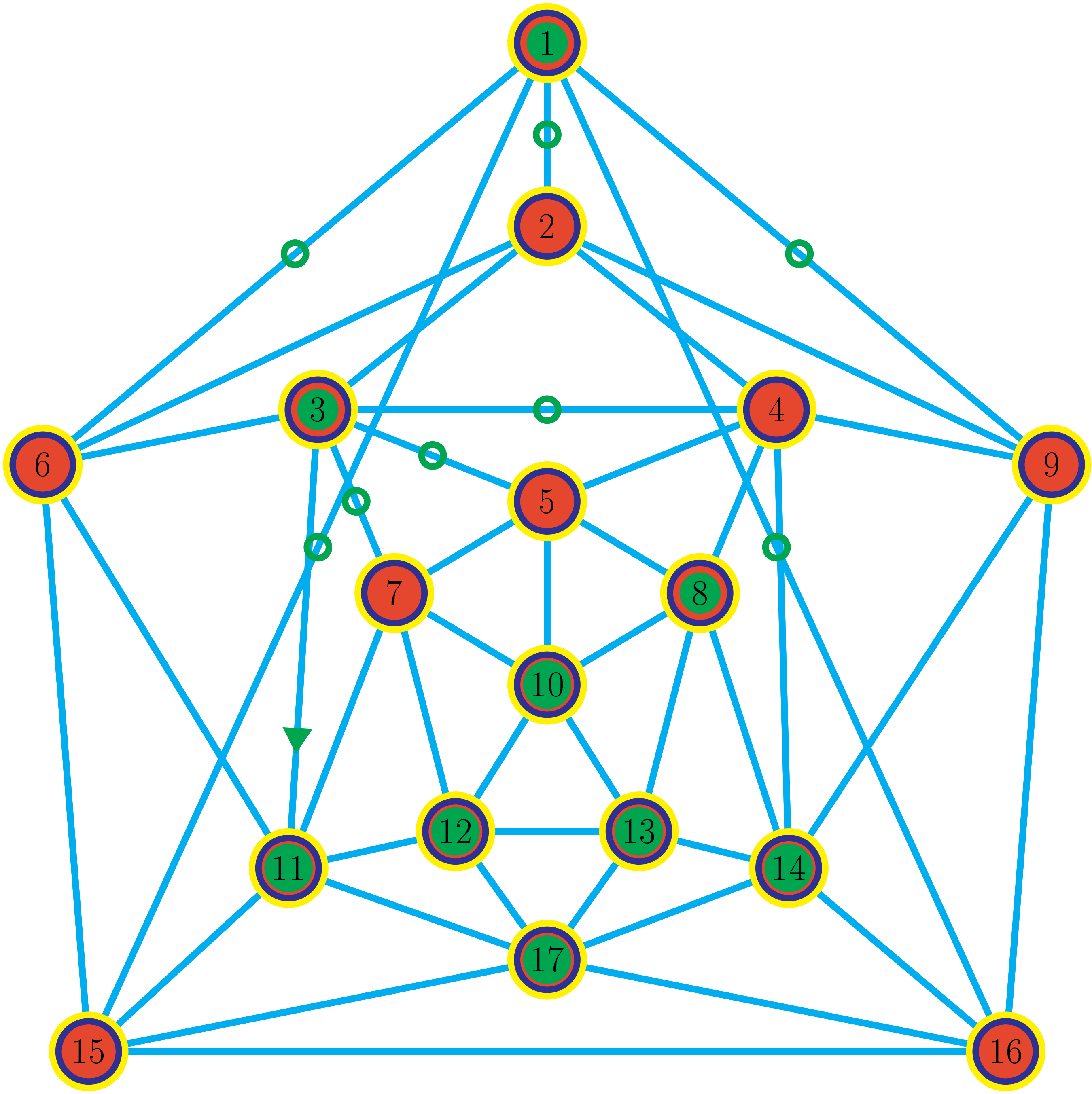


Figure 19.

instruction 33: place edge 3->11 Green DeletionArrow

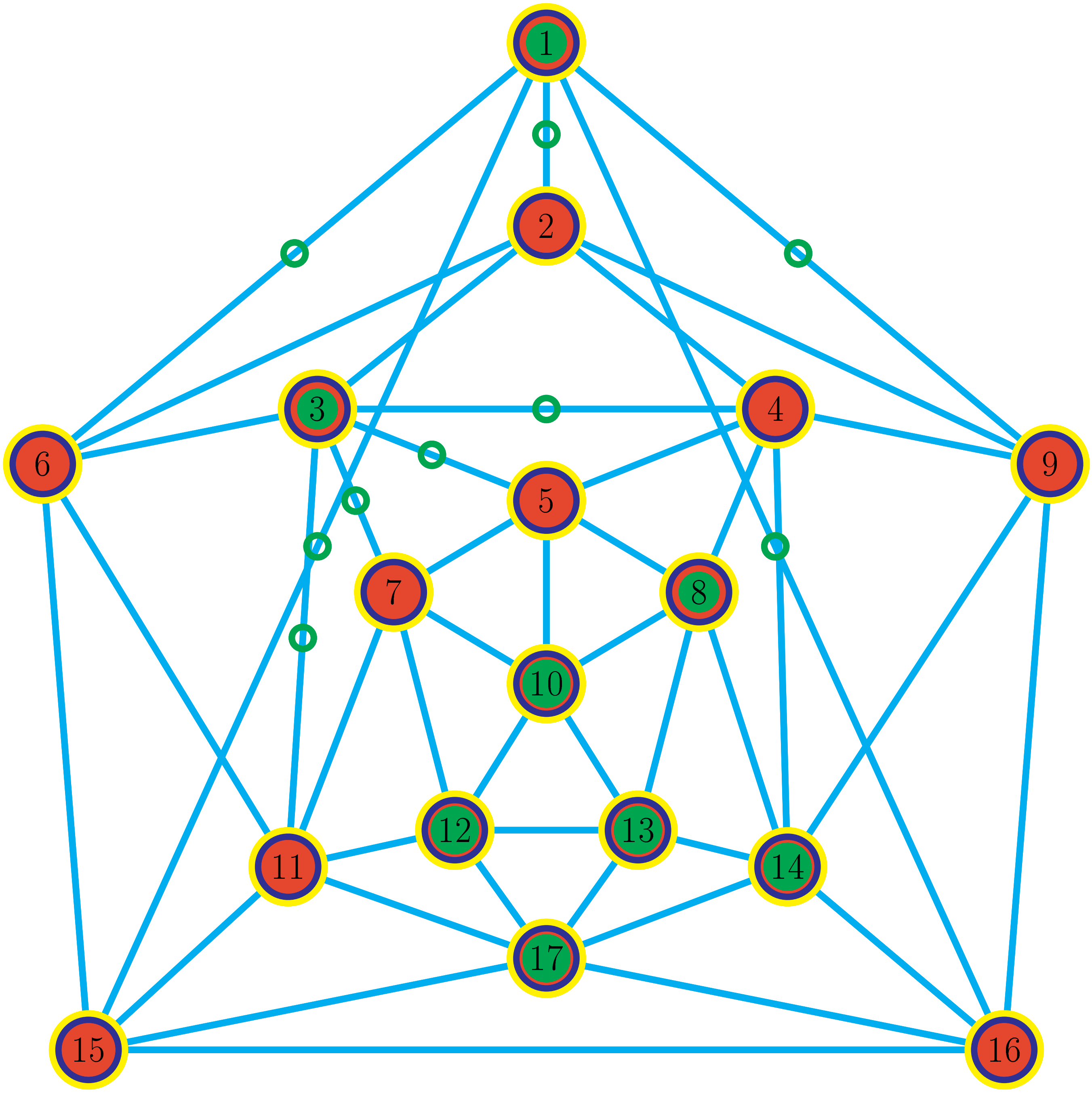


Figure 20.

instruction 34: unplace edge 3->11 Green DeletionArrow
instruction 35: place edge 3-11 Green Checker
instruction 36: unplace vertex 11 Green Checker;

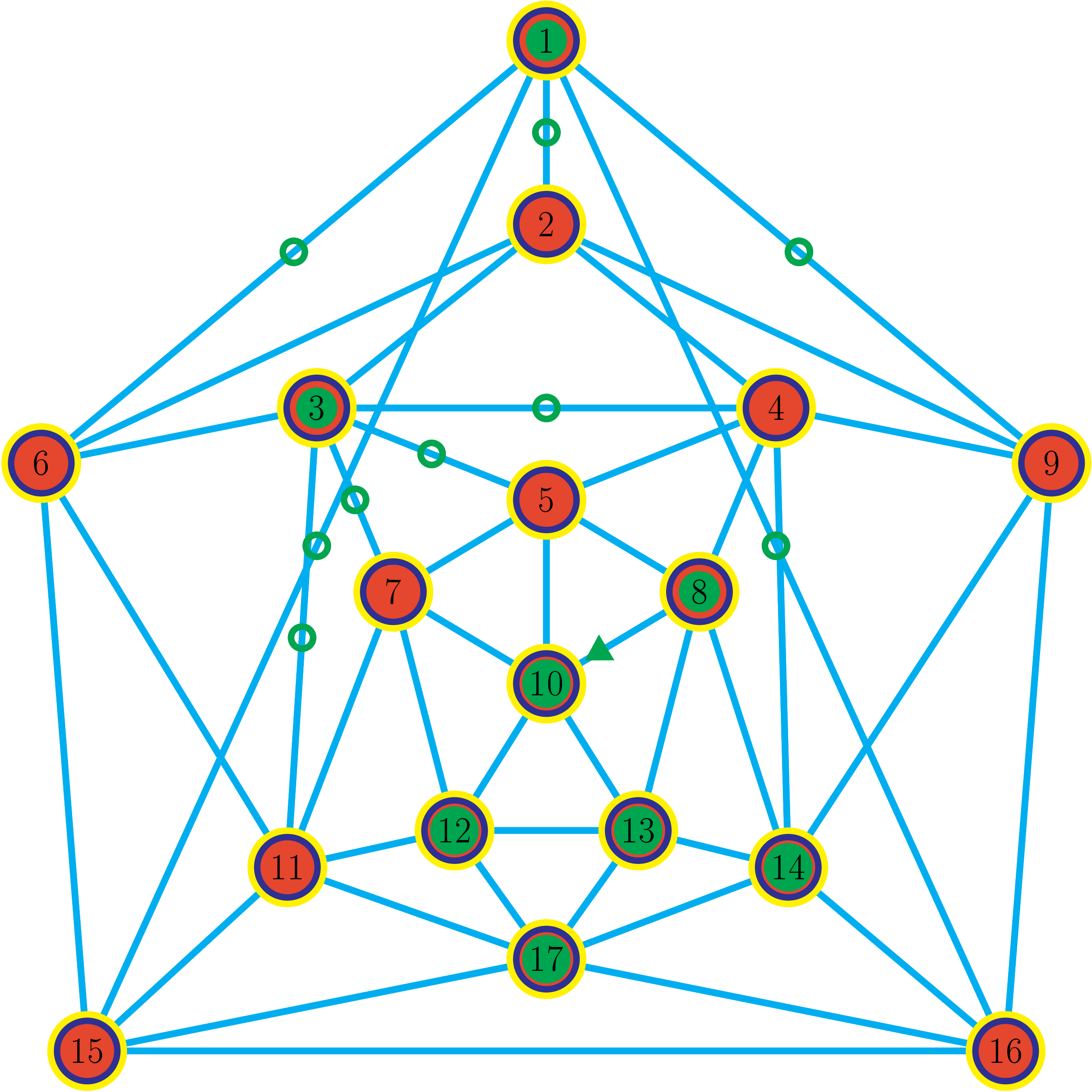


Figure 21.

instruction 37: place edge 8->10 Green DeletionArrow

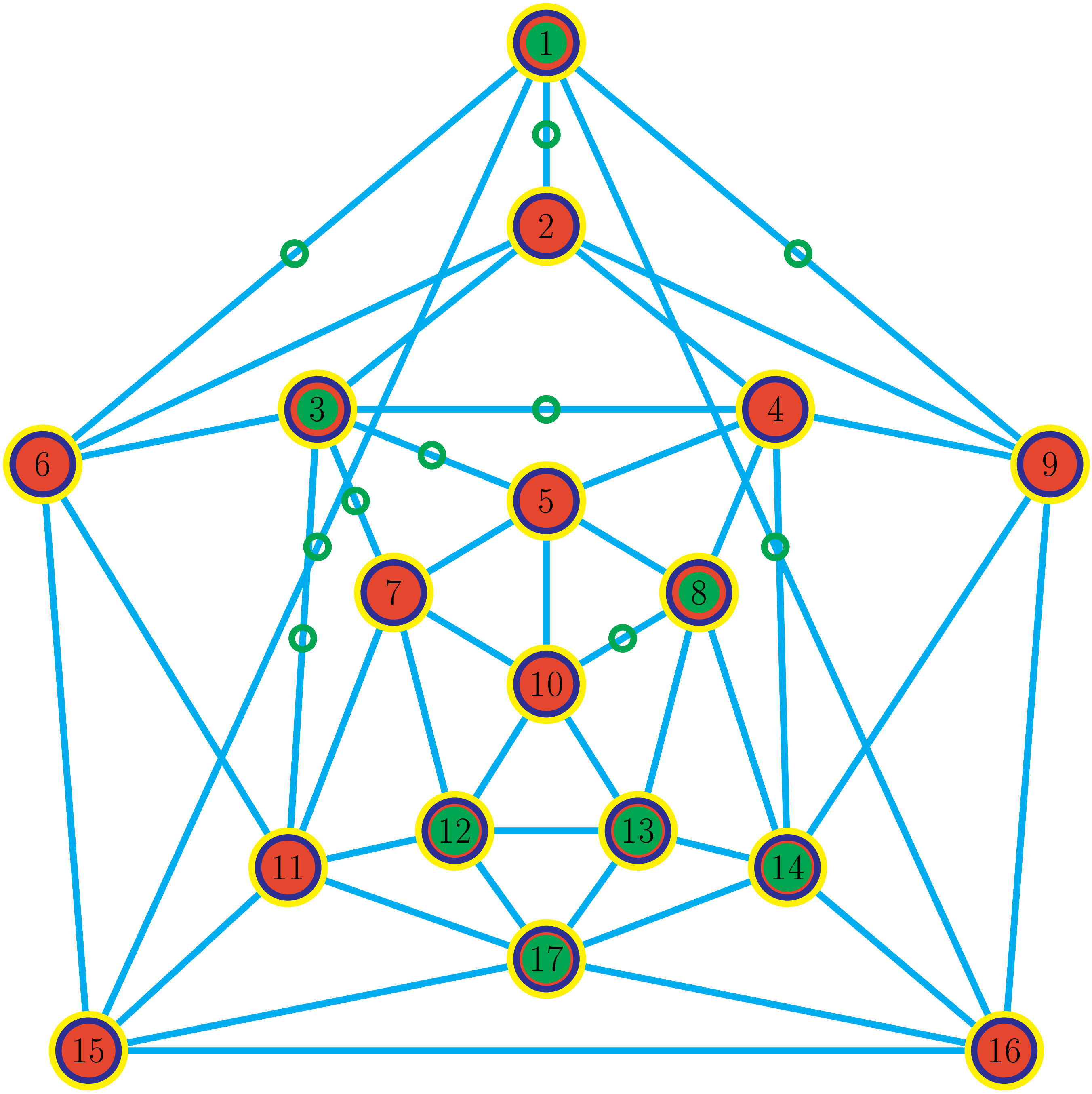


Figure 22.

instruction 38: unplace edge 8->10 Green DeletionArrow
instruction 39: place edge 8-10 Green Checker
instruction 40: unplace vertex 10 Green Checker;

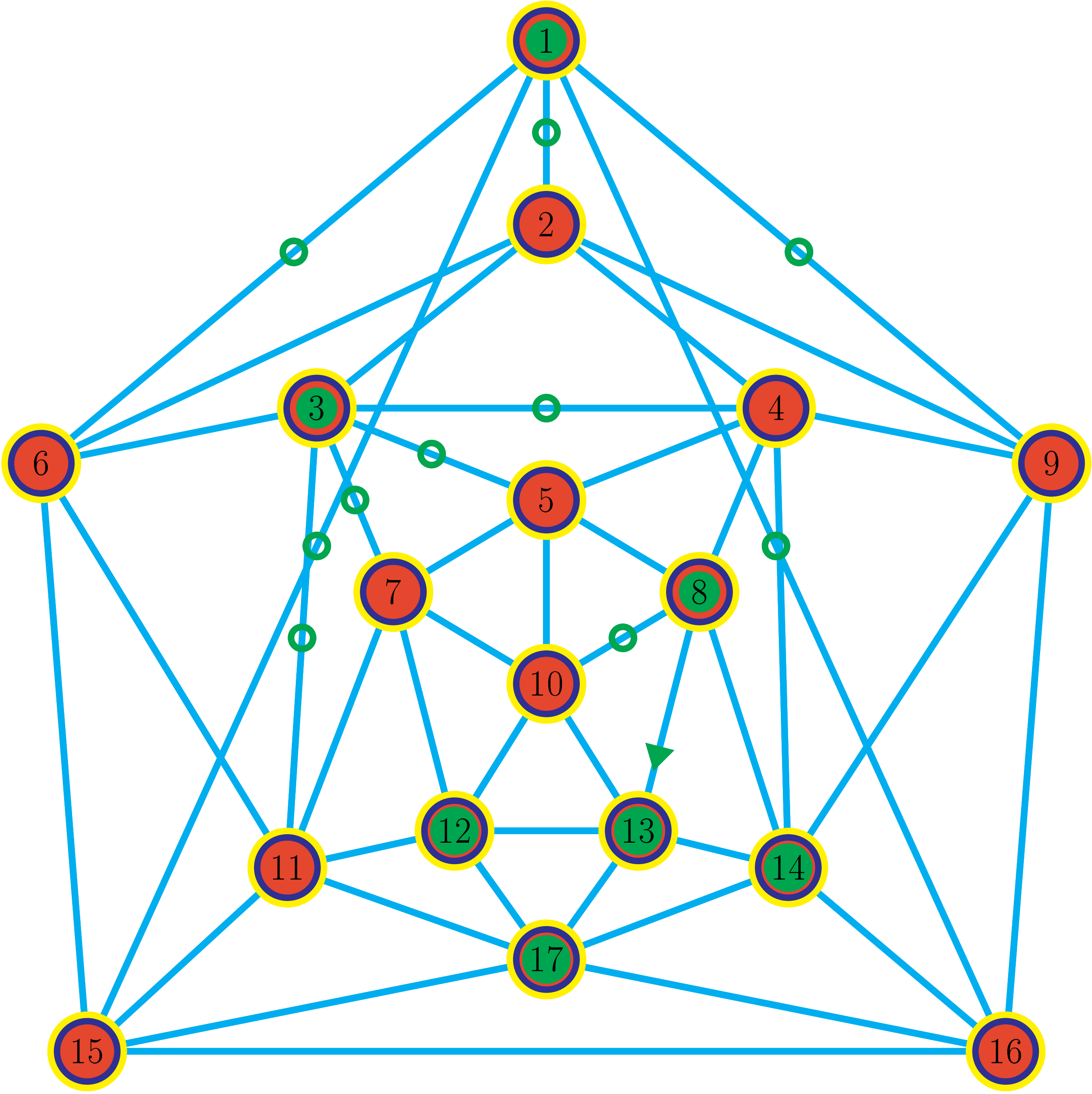


Figure 23.

instruction 41: place edge 8->13 Green DeletionArrow

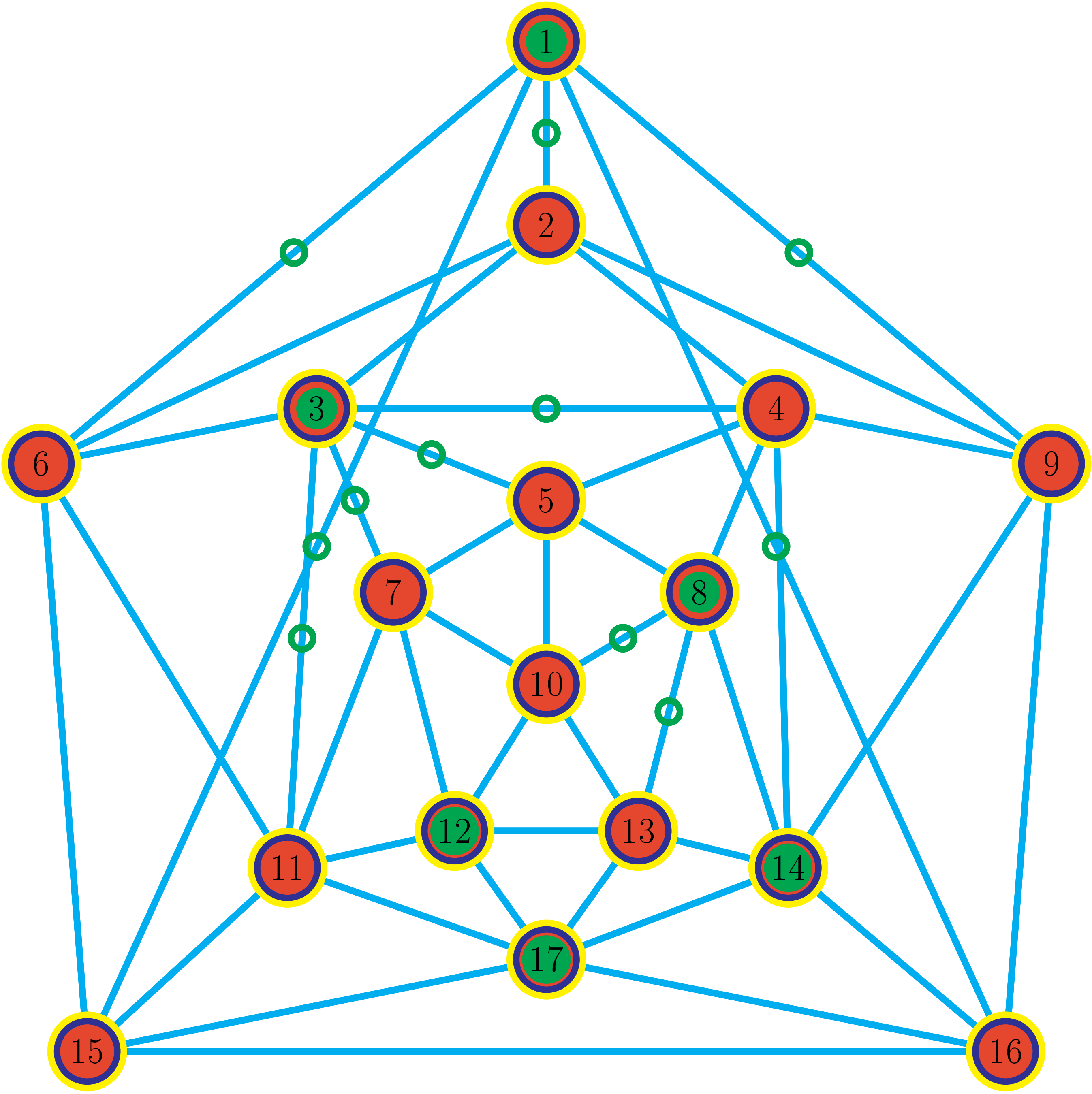


Figure 24.
instruction 42: unplace edge 8->13 Green DeletionArrow
instruction 43: place edge 8-13 Green Checker
instruction 44: unplace vertex 13 Green Checker;

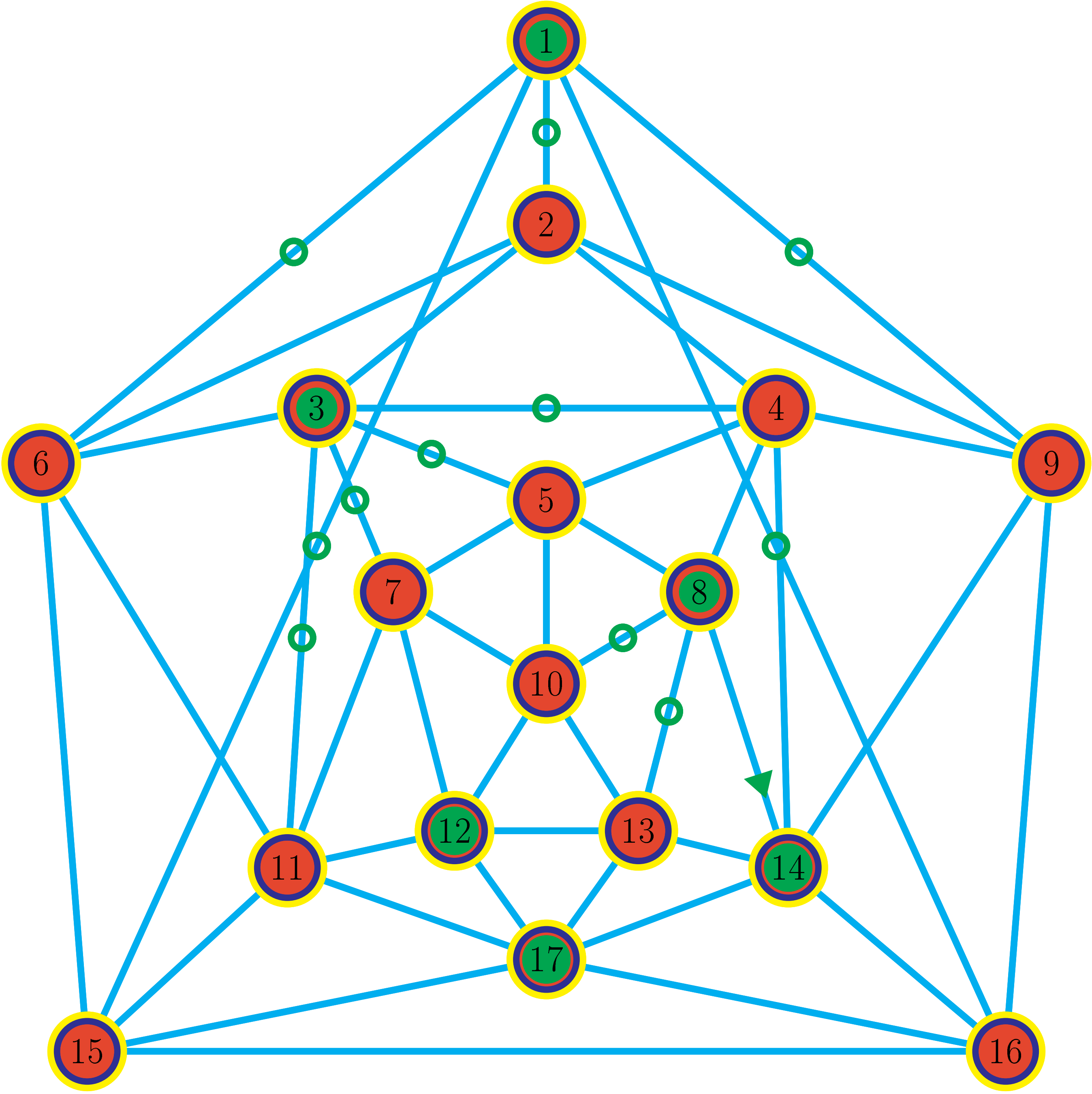


Figure 25.

instruction 45: place edge 8->14 Green DeletionArrow

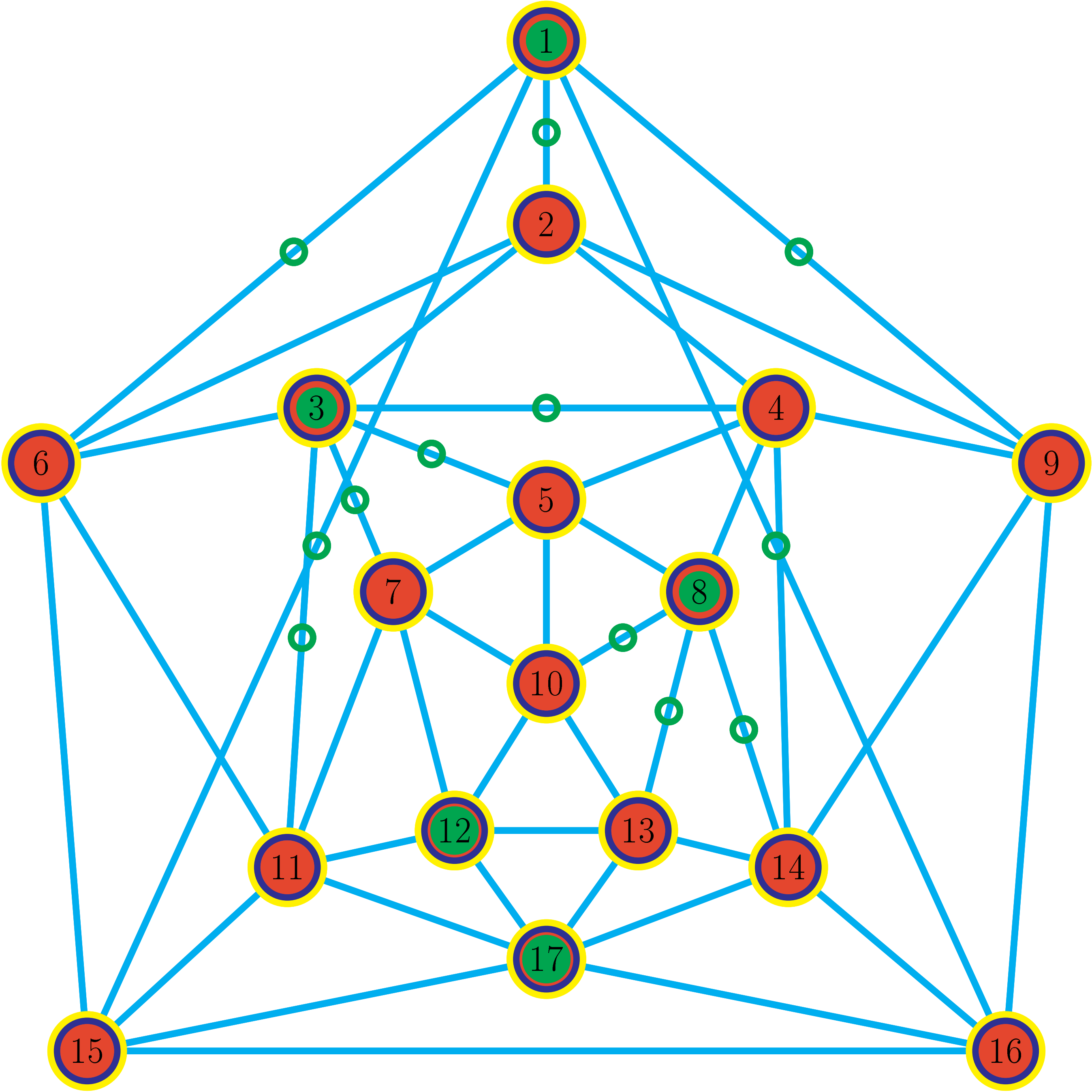


Figure 26.

instruction 46: unplace edge 8->14 Green DeletionArrow
instruction 47: place edge 8-14 Green Checker
instruction 48: unplace vertex 14 Green Checker;

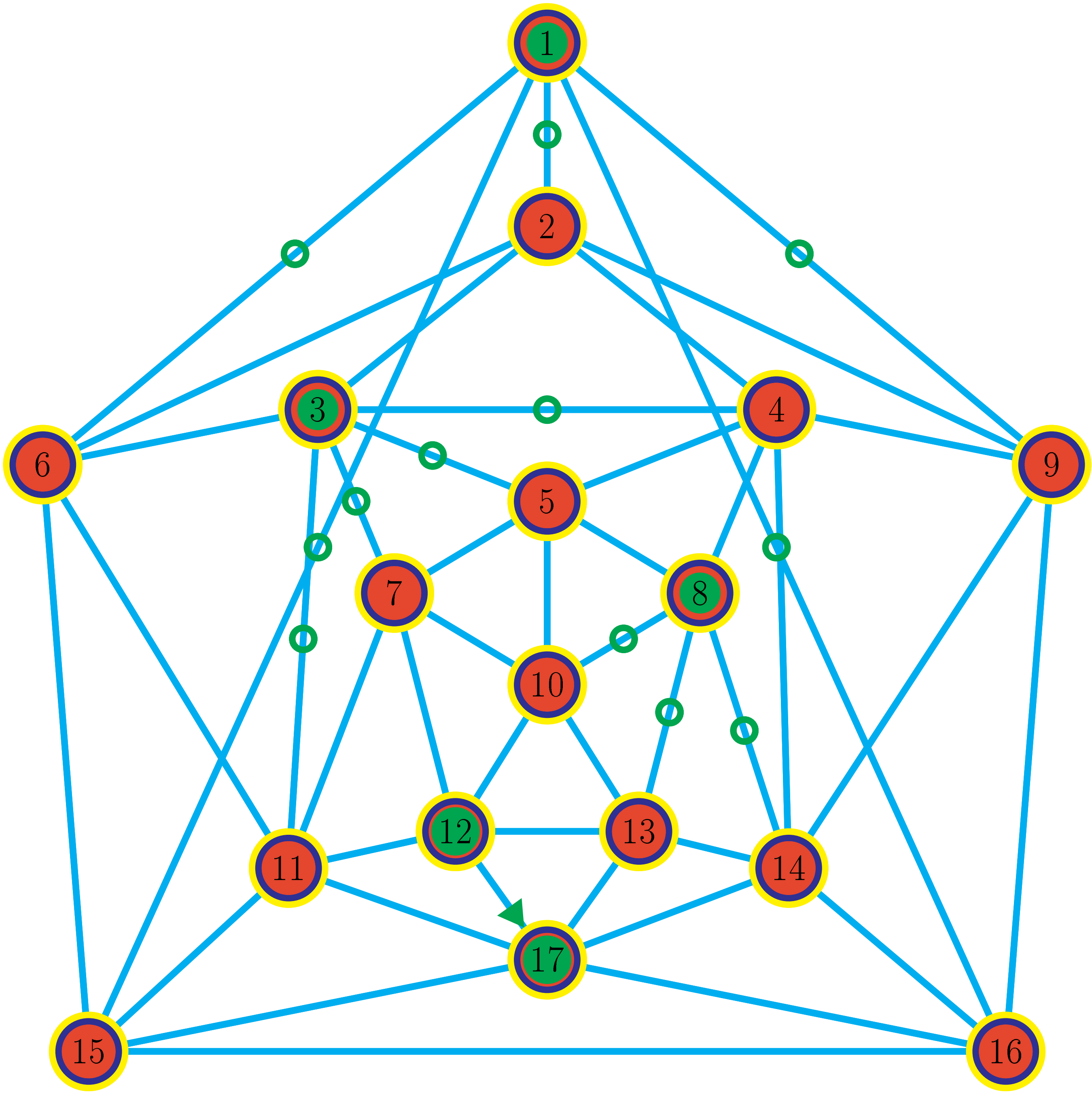


Figure 27.

instruction 49: place edge 12->17 Green DeletionArrow

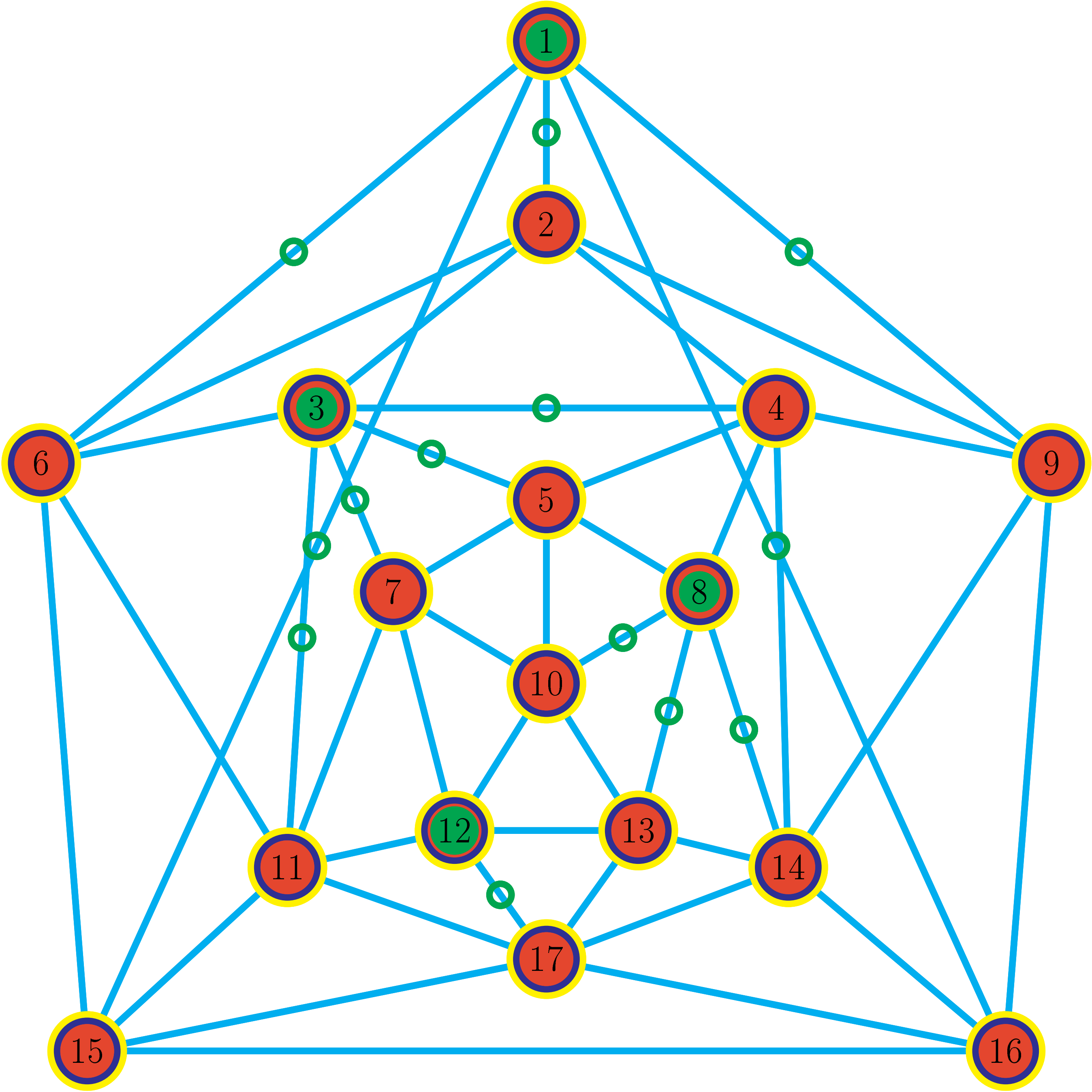


Figure 28.
instruction 50: unplace edge 12->17 Green DeletionArrow
instruction 51: place edge 12-17 Green Checker
instruction 52: unplace vertex 17 Green Checker;

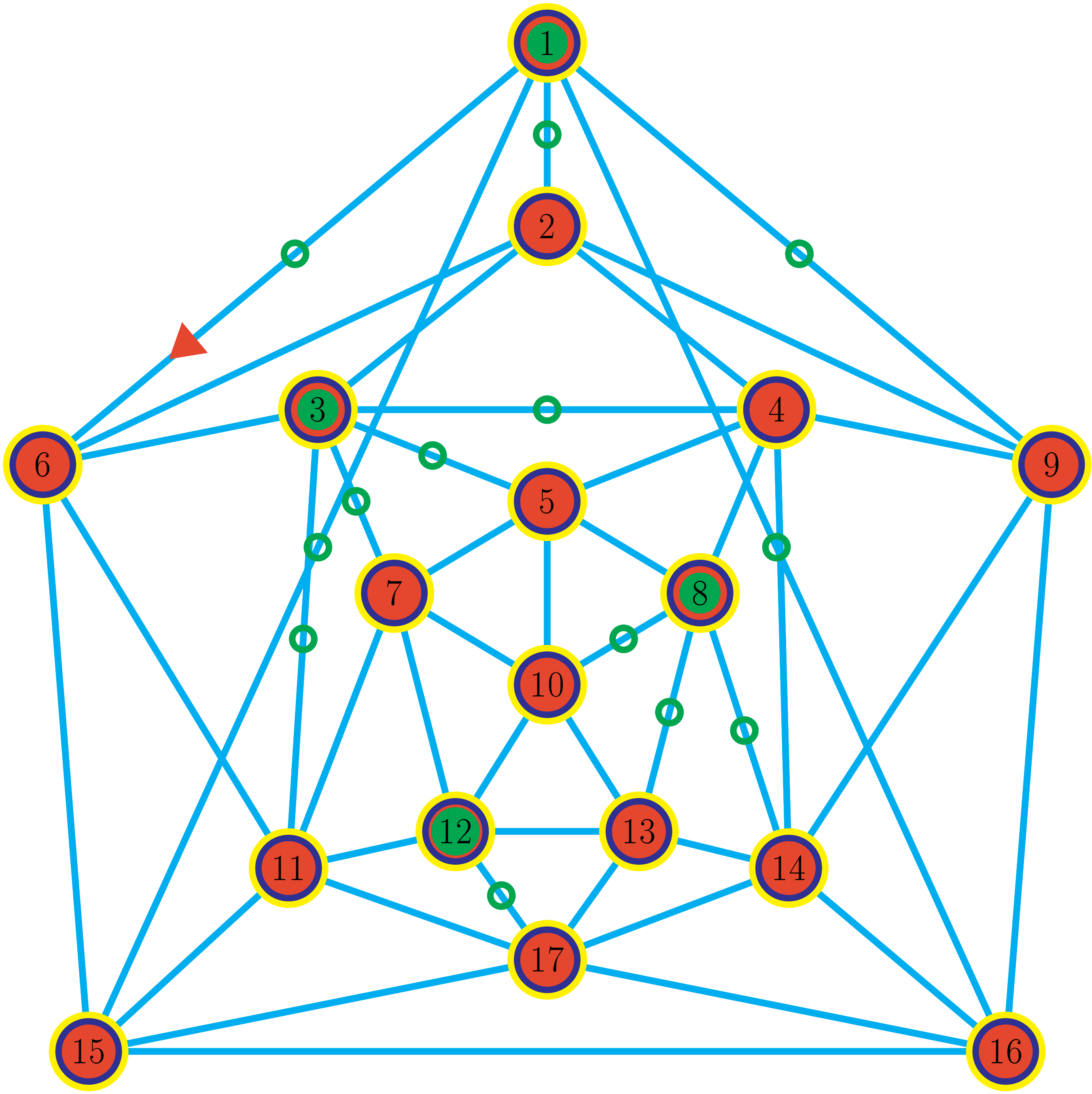


Figure 29.

instruction 53: place edge 1->6 Red DeletionArrow

When the above instruction is executed by the computer a triangular red arrowhead is placed on the directed edge 1→6. A triangular red arrowhead is used to indicate the deletion of the red checker on the vertex pointed to (vertex 6). Define triangle arrowheads as deletion arrows. Later we'll define insertion arrows.

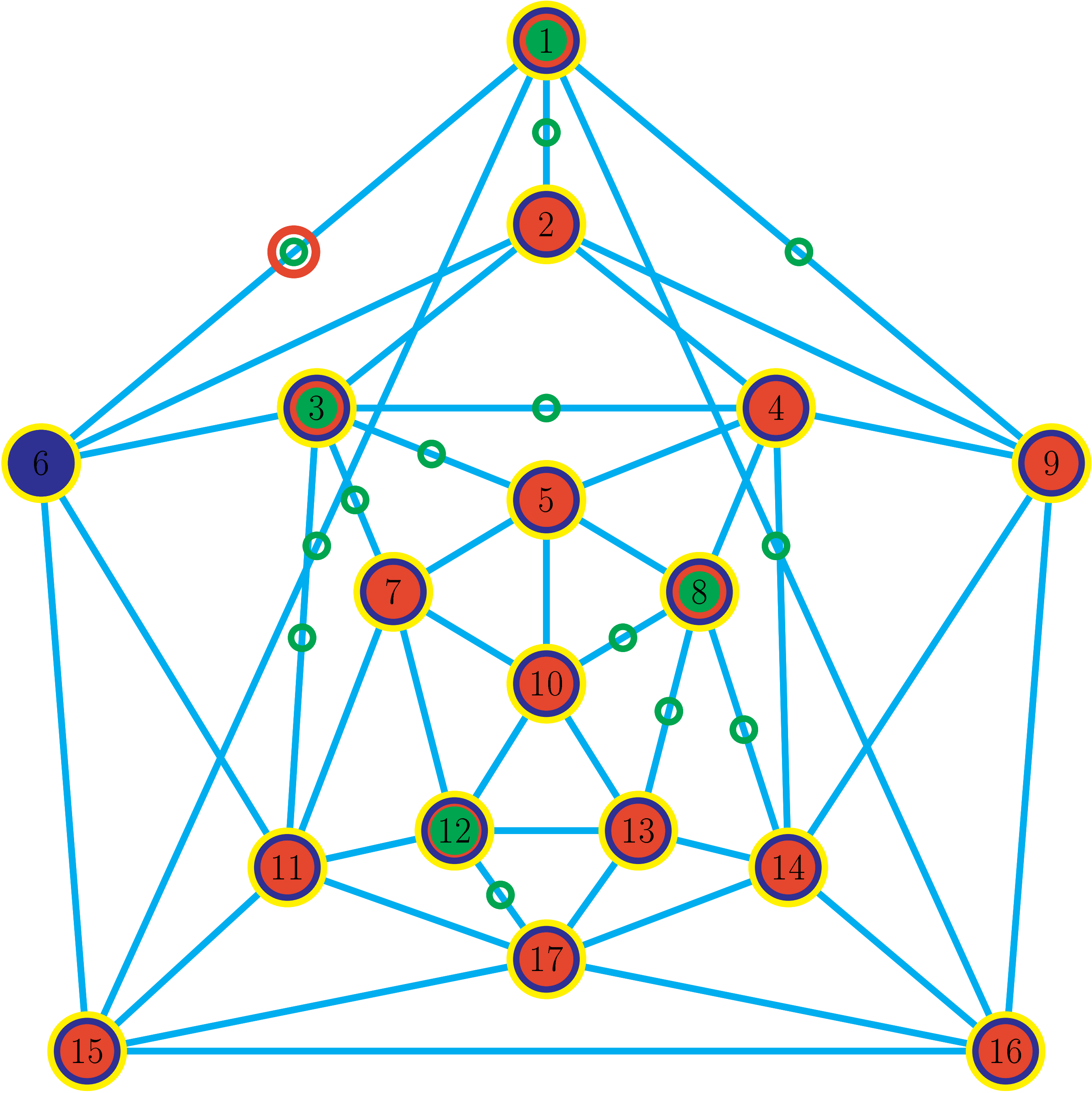


FIGURE 30.

instruction 54: unplace edge 1->6 Red DeletionArrow
instruction 55: place edge 1-6 Red Checker
instruction 56: unplace vertex 6 Red Checker;

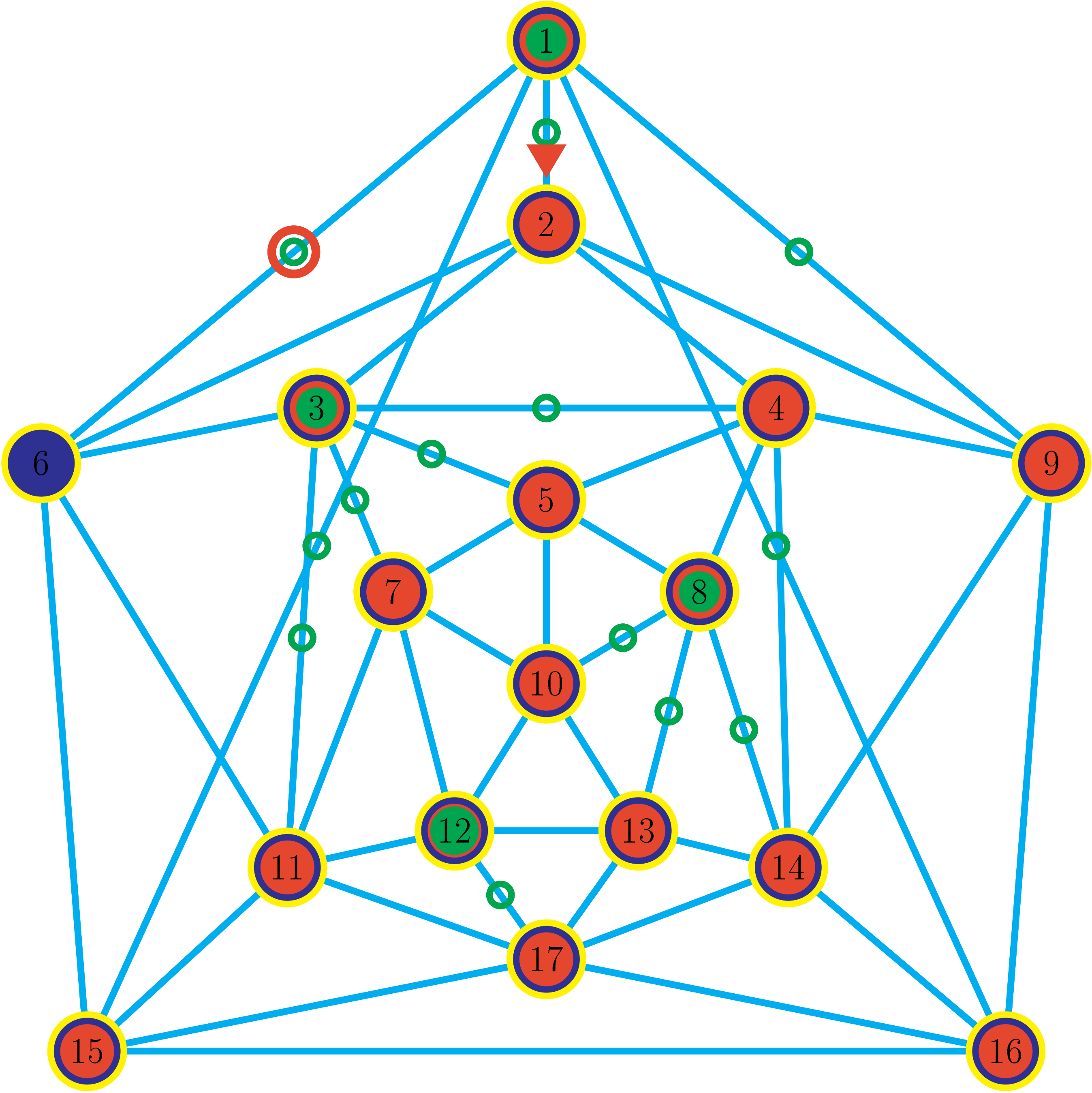


Figure 31.

instruction 57: place edge 1->2 Red DeletionArrow

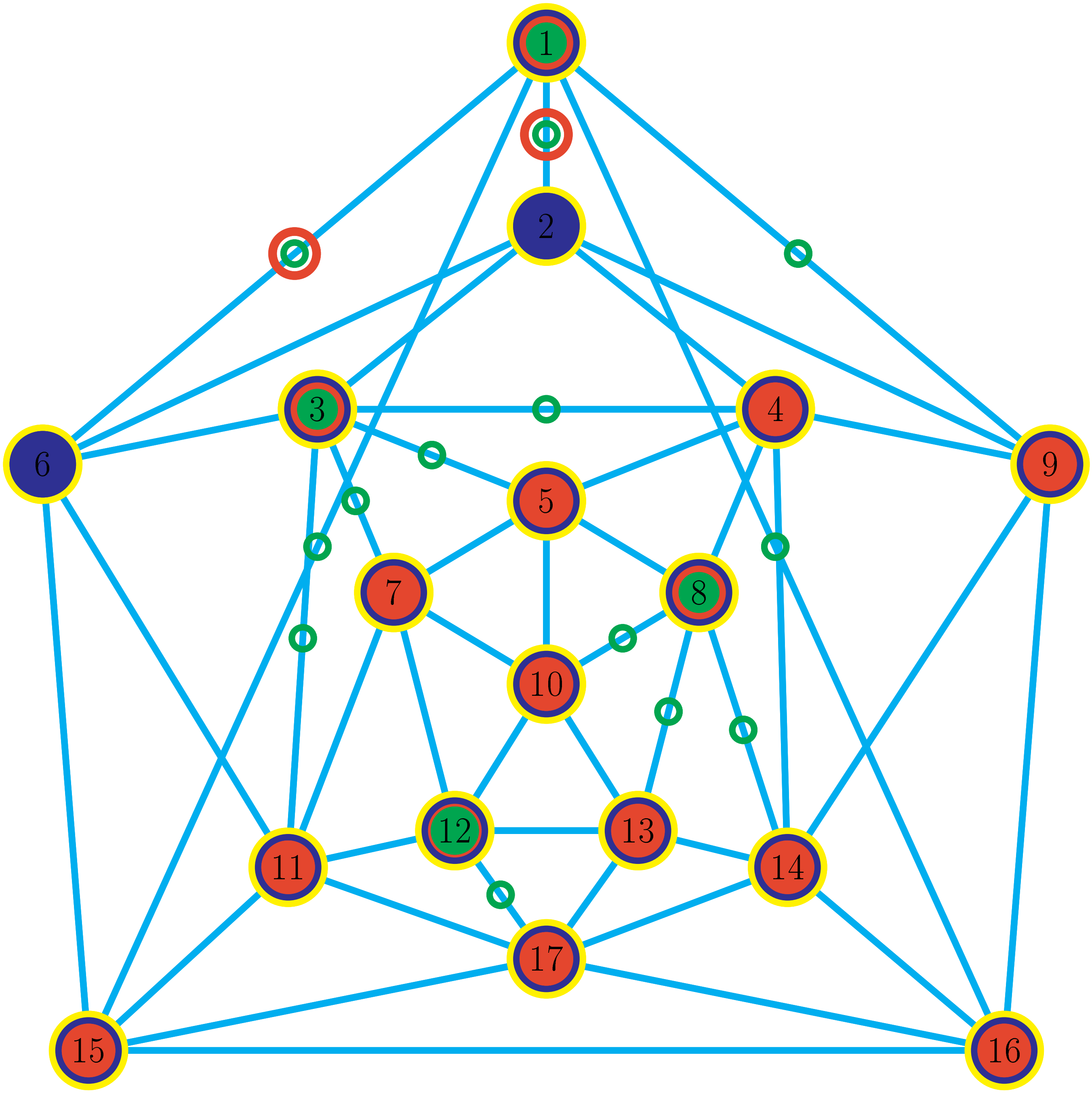


Figure 32.

instruction 58: unplace edge 1->2 Red DeletionArrow
instruction 59: place edge 1-2 Red Checker
instruction 60: unplace vertex 2 Red Checker;

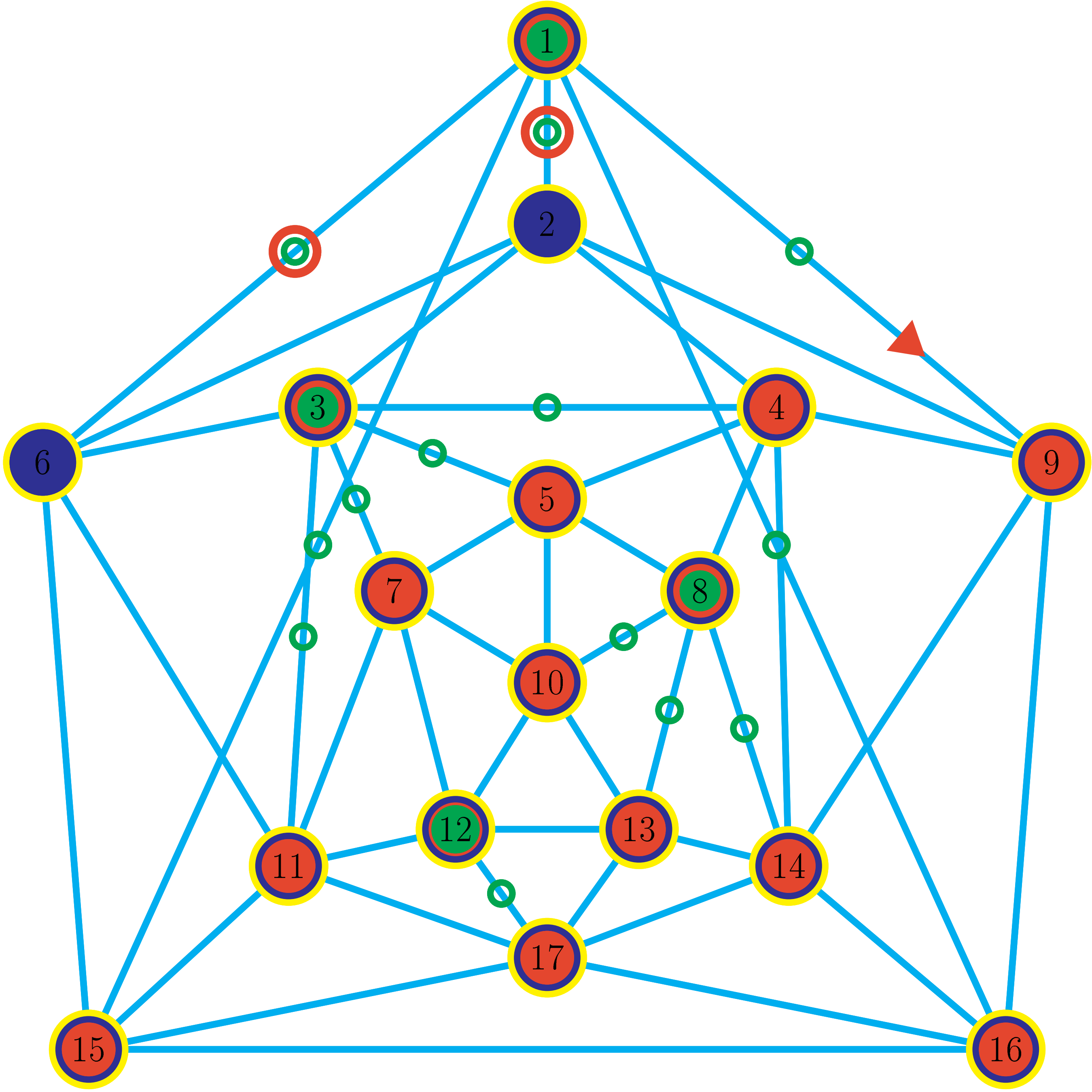


FIGURE 33.

instruction 61: place edge 1->9 Red DeletionArrow

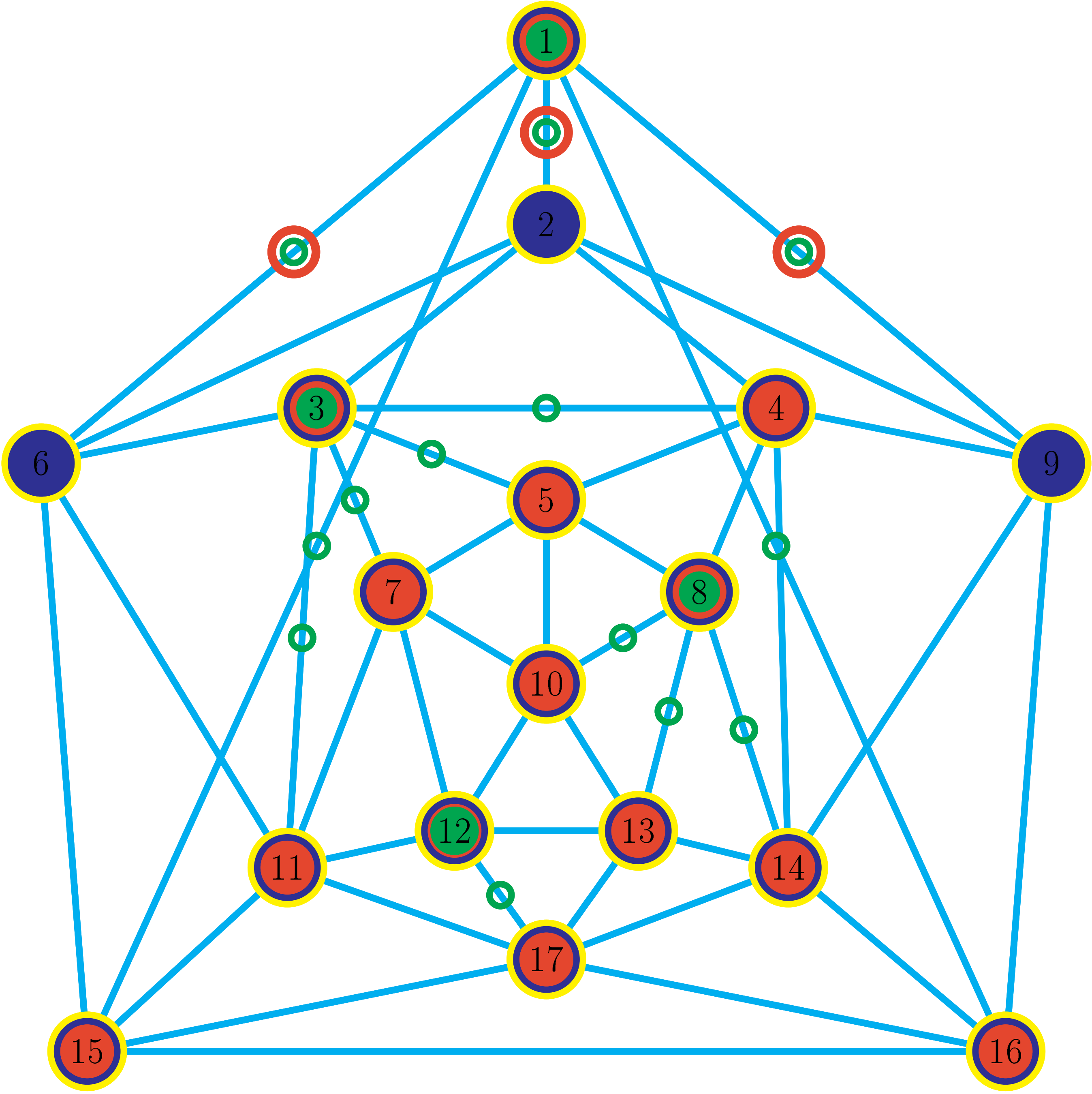


Figure 34.

instruction 62: unplace edge 1->9 Red DeletionArrow
instruction 63: place edge 1-9 Red Checker
instruction 64: unplace vertex 9 Red Checker;

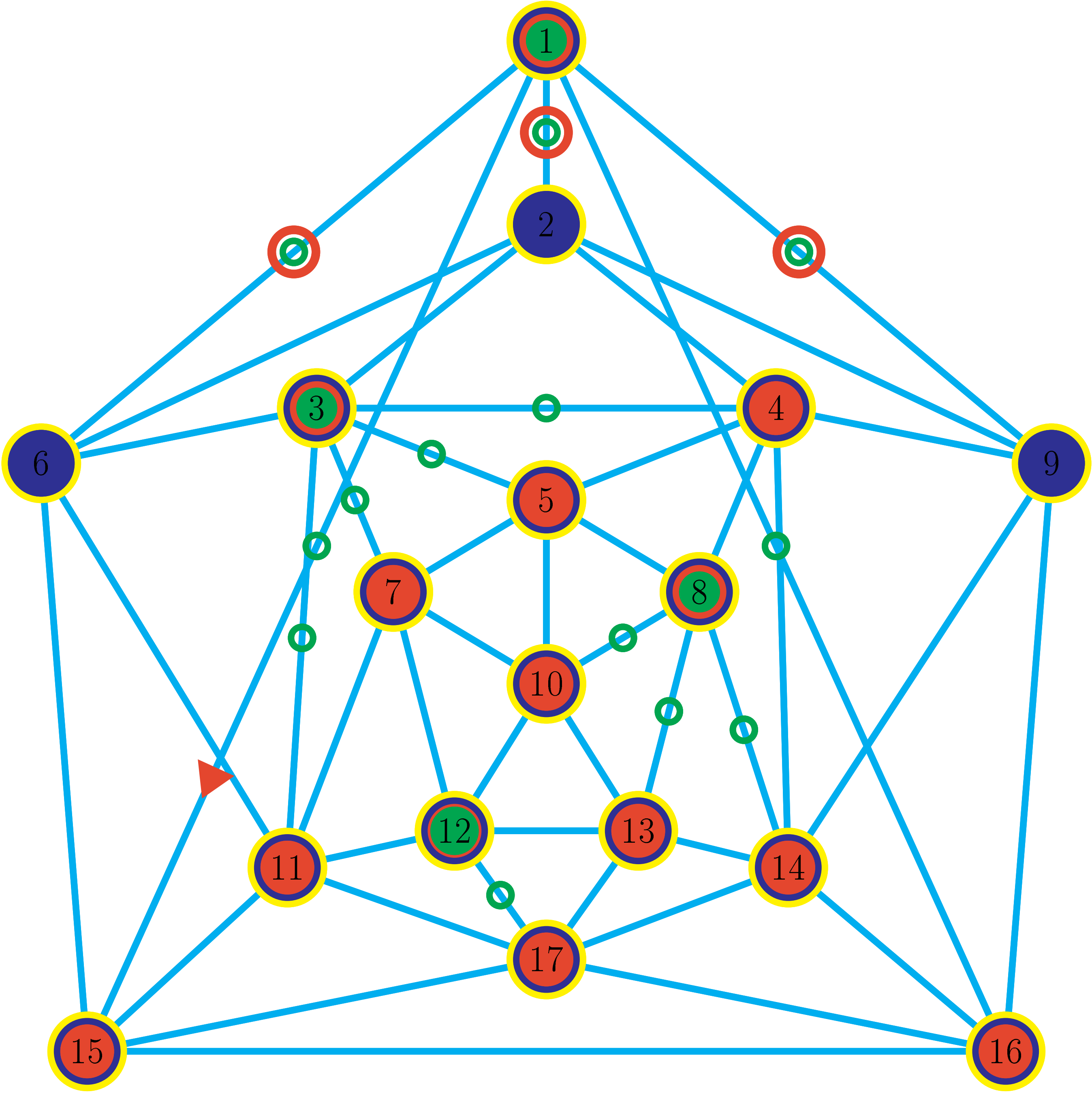


Figure 35.

instruction 65: place edge 1->15 Red DeletionArrow

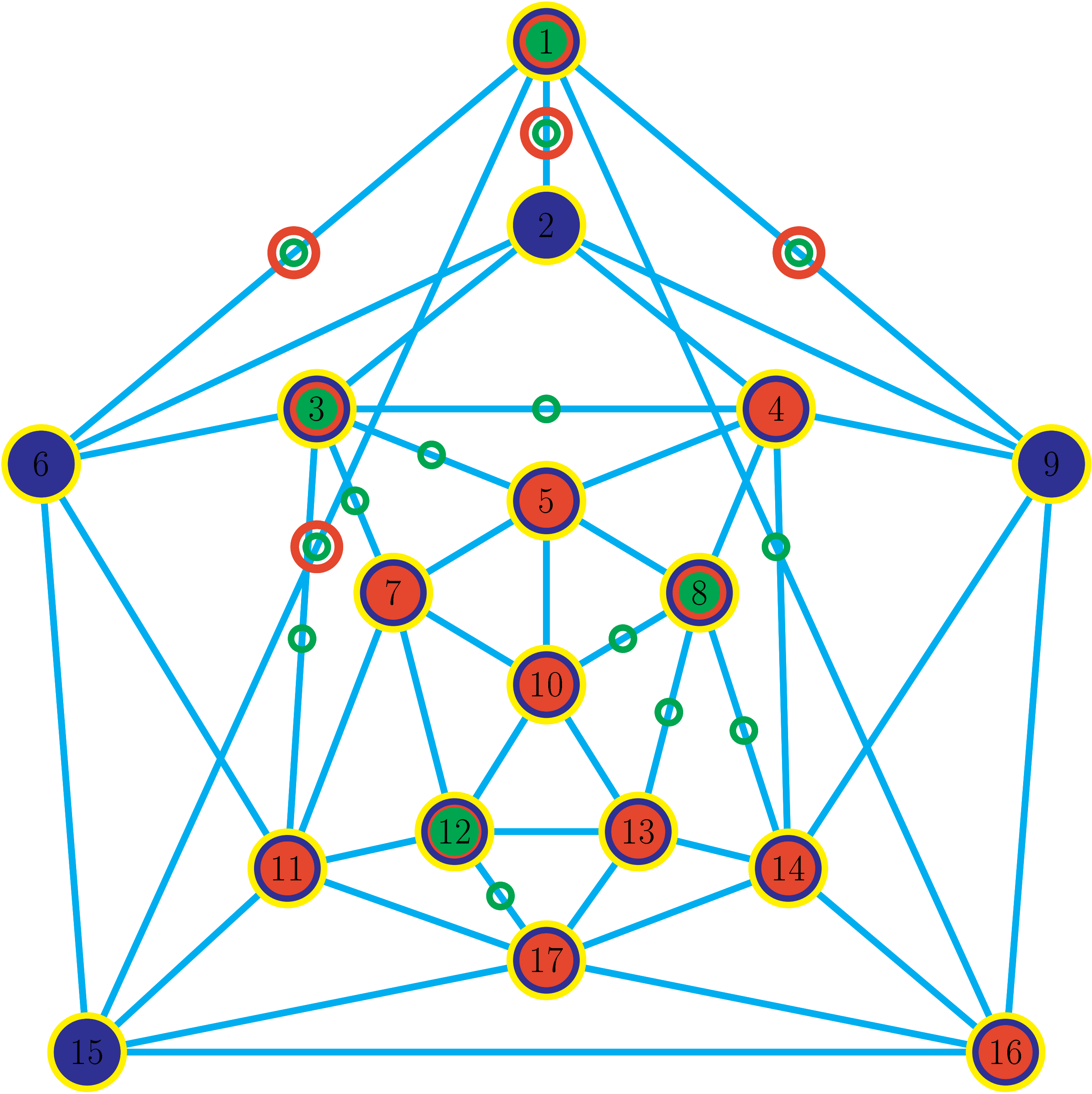


Figure 36.

instruction 66: unplace edge 1->15 Red DeletionArrow
instruction 67: place edge 1-15 Red Checker
instruction 68: unplace vertex 15 Red Checker;

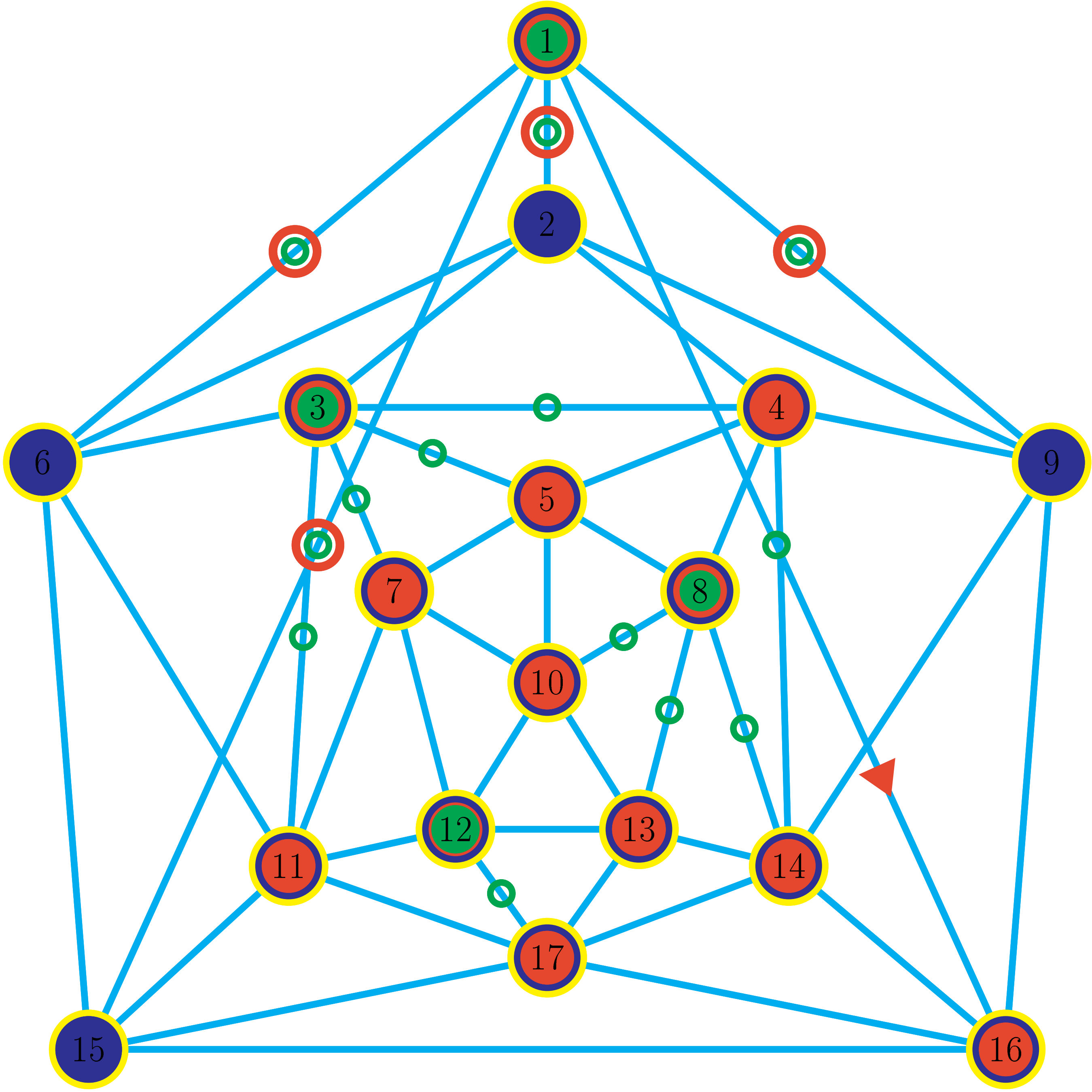


Figure 37.

instruction 69: place edge 1->16 Red DeletionArrow

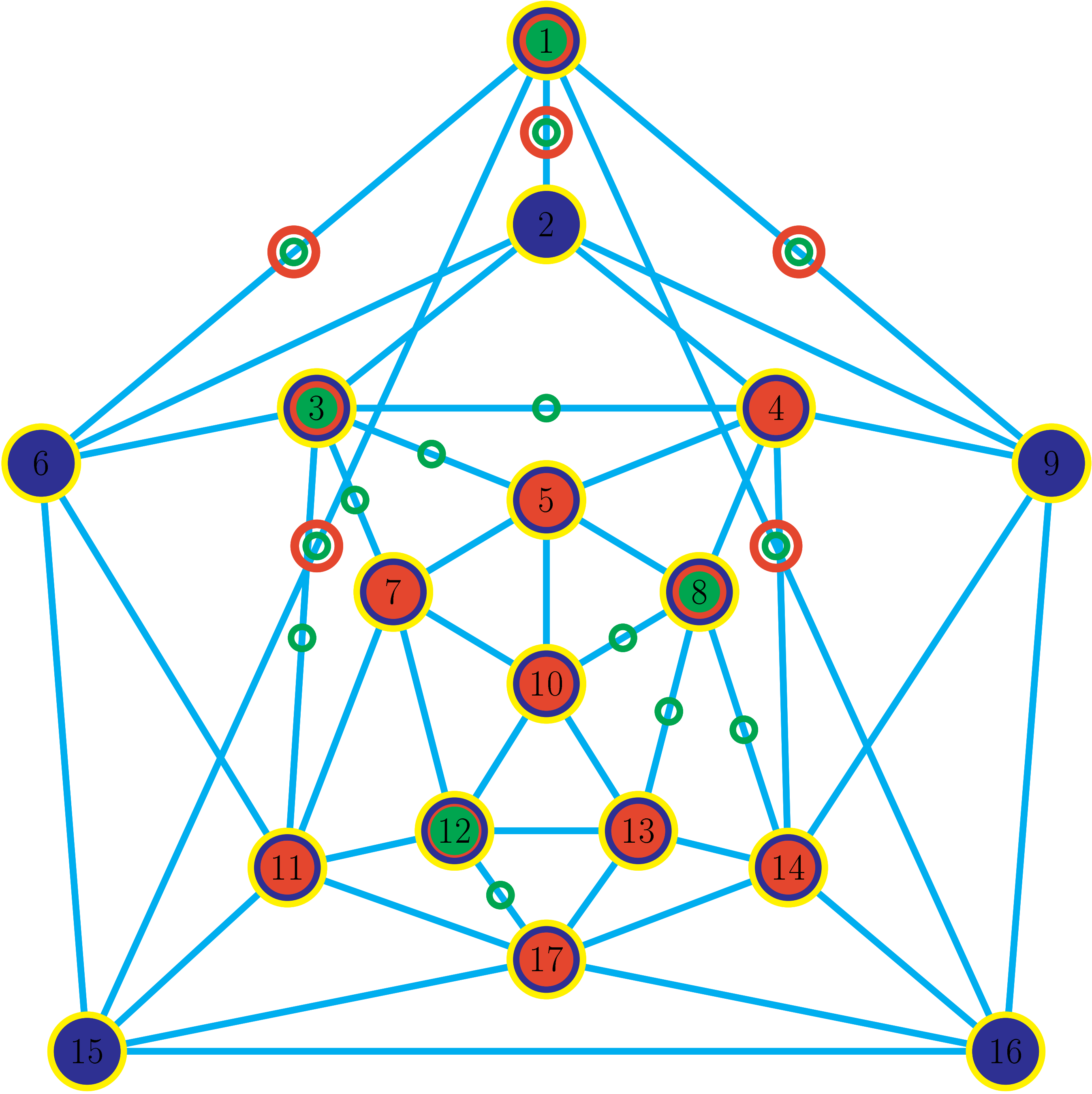


Figure 38.

instruction 70: unplace edge 1->16 Red DeletionArrow
instruction 71: place edge 1-16 Red Checker
instruction 72: unplace vertex 16 Red Checker;

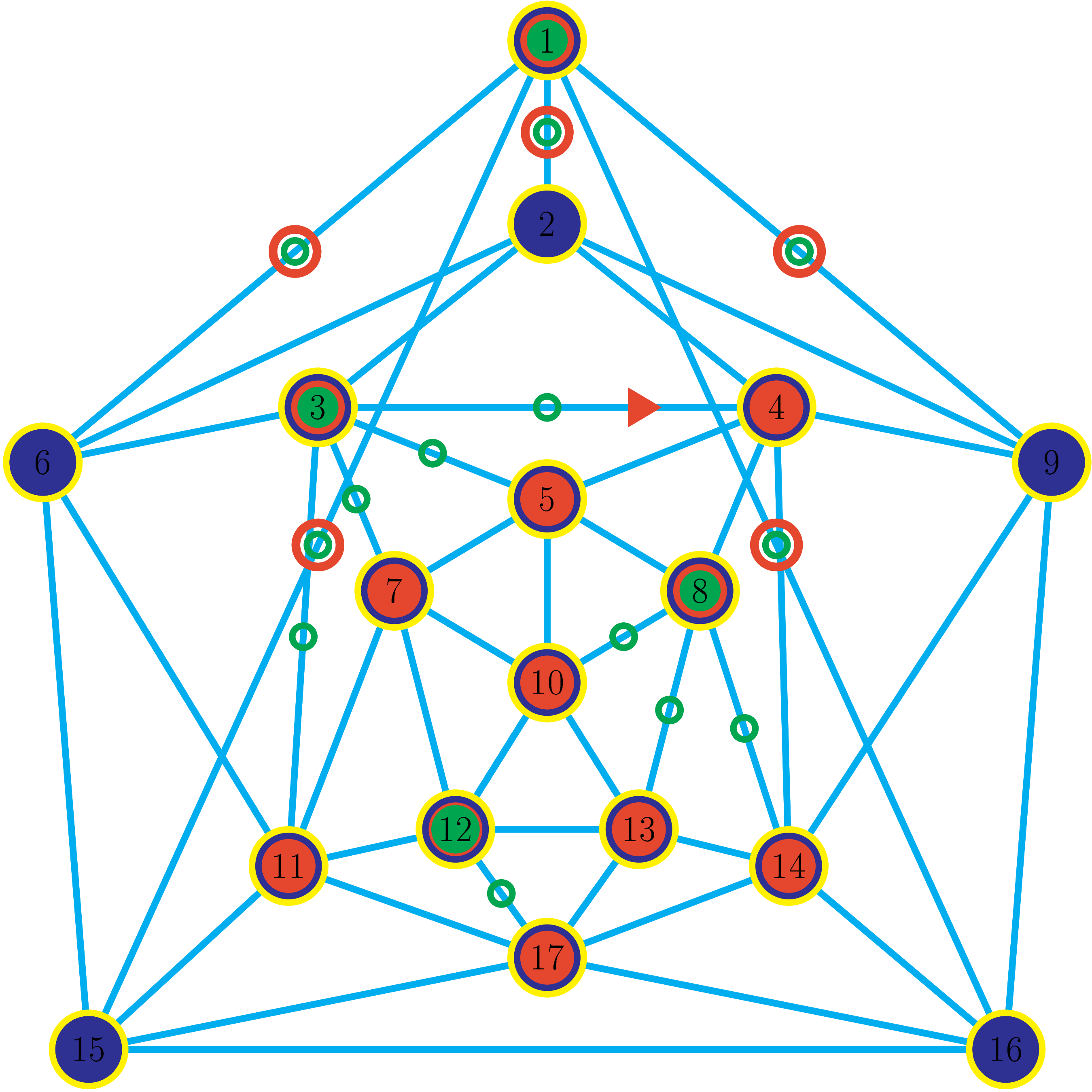


Figure 39.

instruction 73: place edge 3->4 Red DeletionArrow

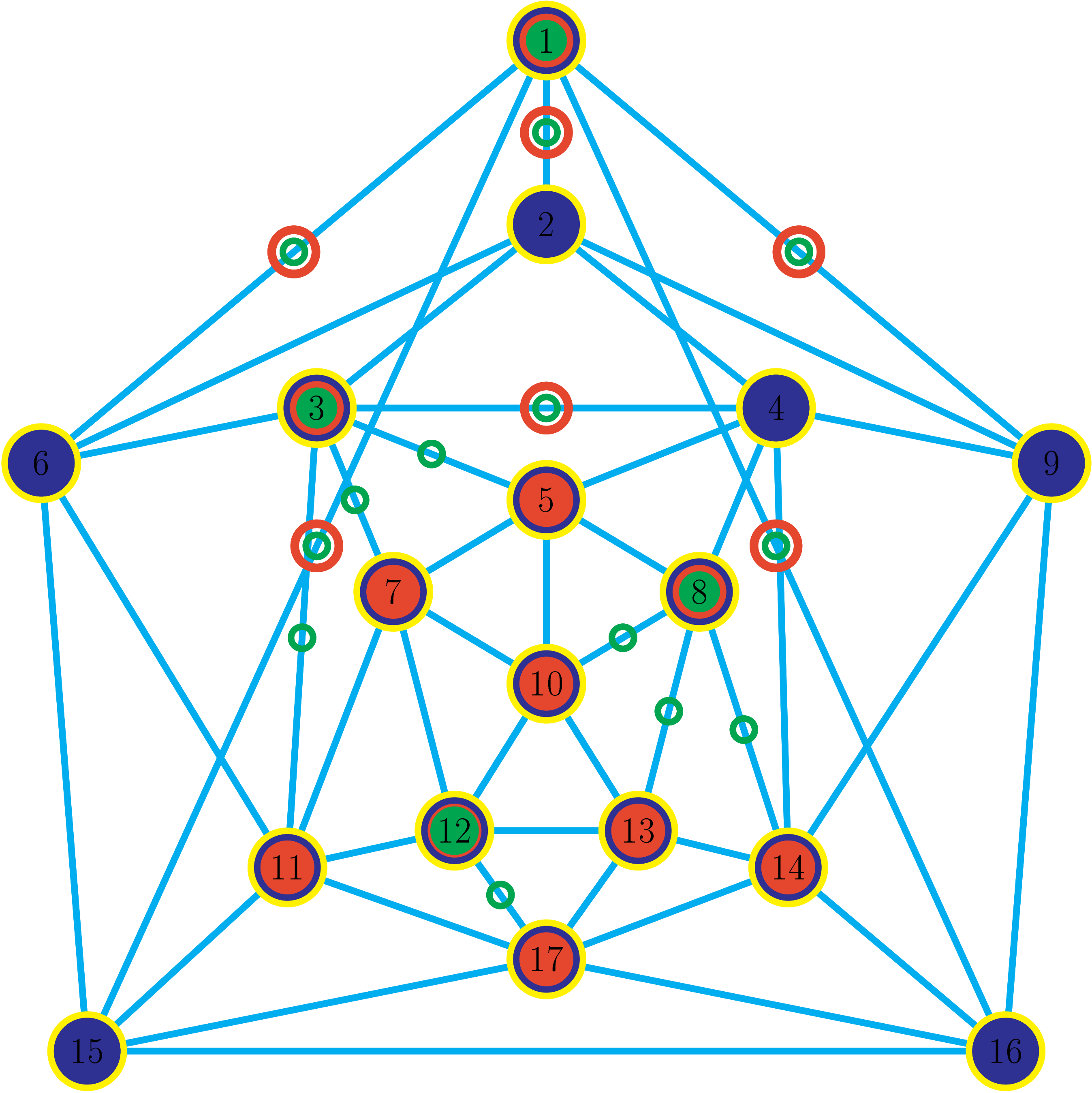


Figure 40.

instruction 74: unplace edge 3->4 Red DeletionArrow
instruction 75: place edge 3-4 Red Checker
instruction 76: unplace vertex 4 Red Checker;

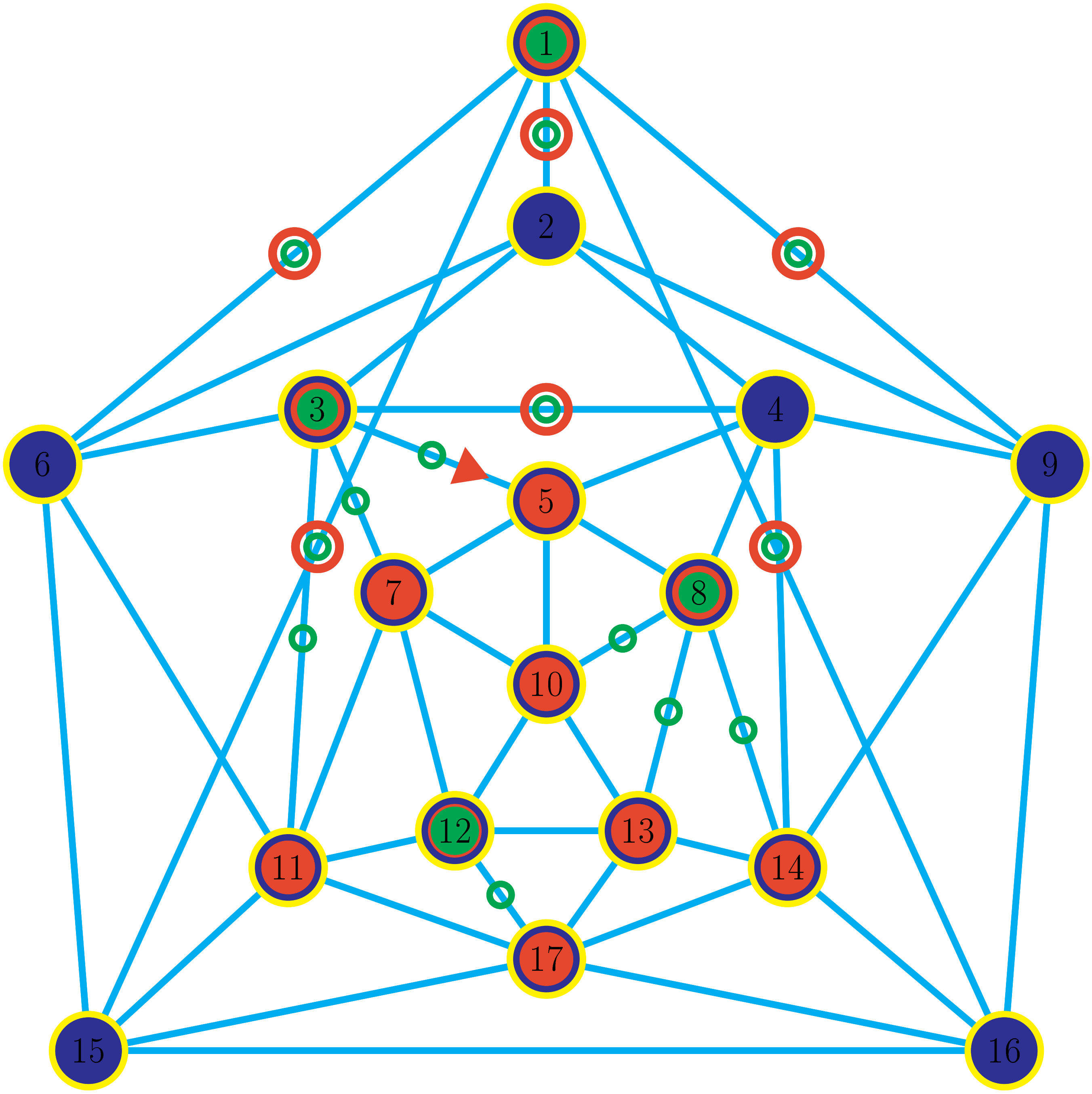


Figure 41.

instruction 77: place edge 3->5 Red DeletionArrow

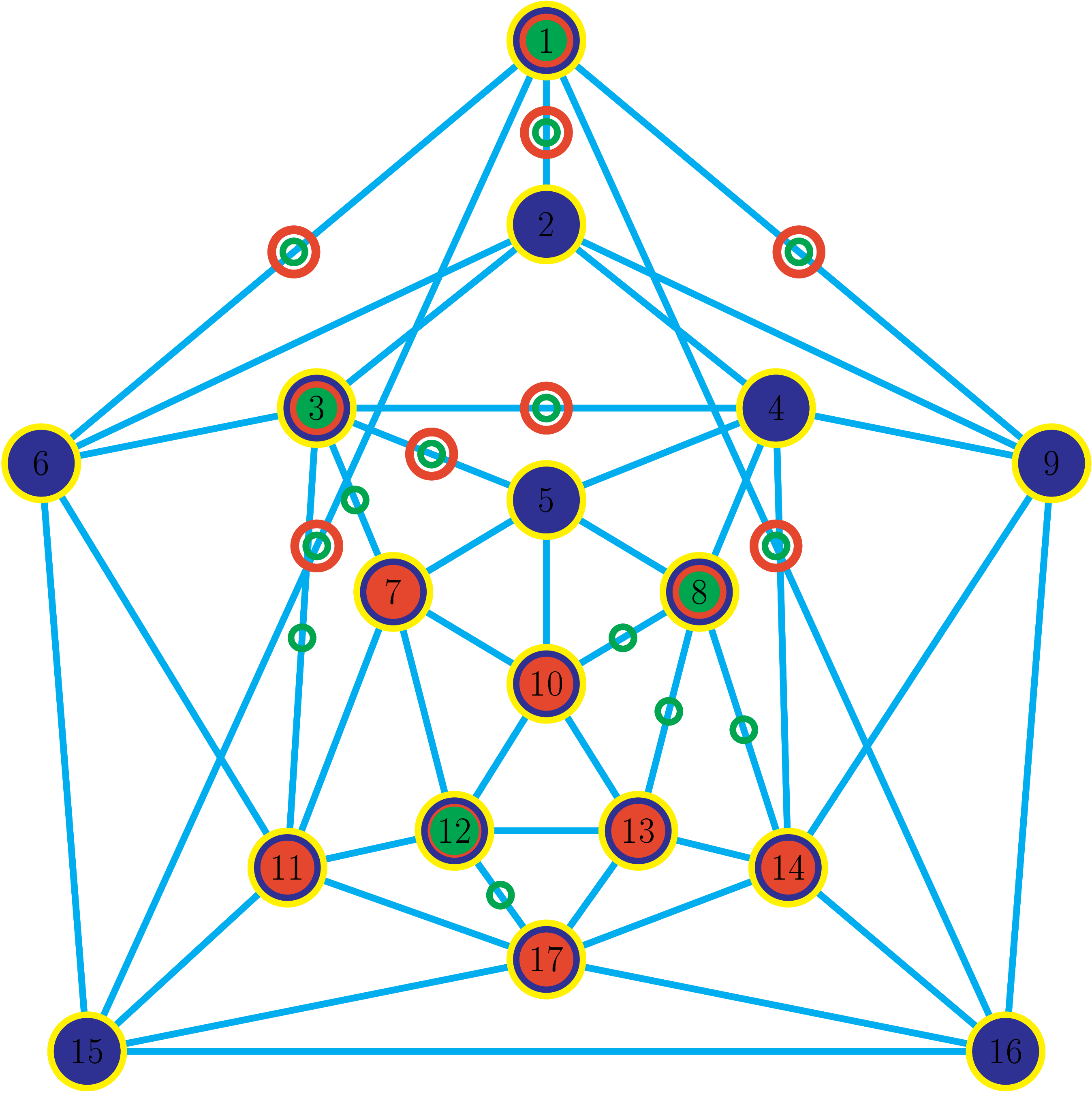


Figure 42.

instruction 78: unplace edge 3->5 Red DeletionArrow
instruction 79: place edge 3-5 Red Checker
instruction 80: unplace vertex 5 Red Checker;

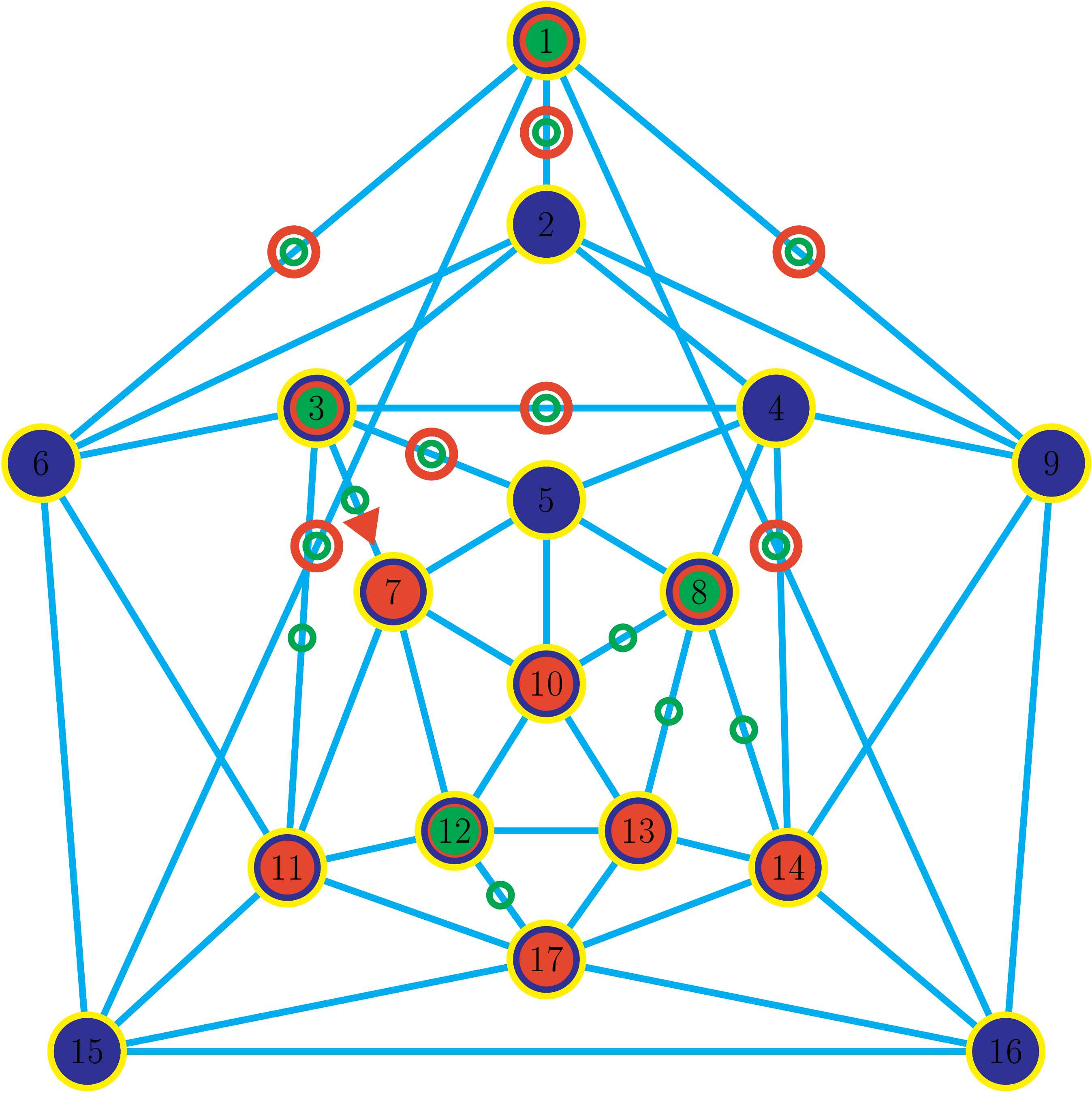


Figure 43.

instruction 81: place edge 3->7 Red DeletionArrow

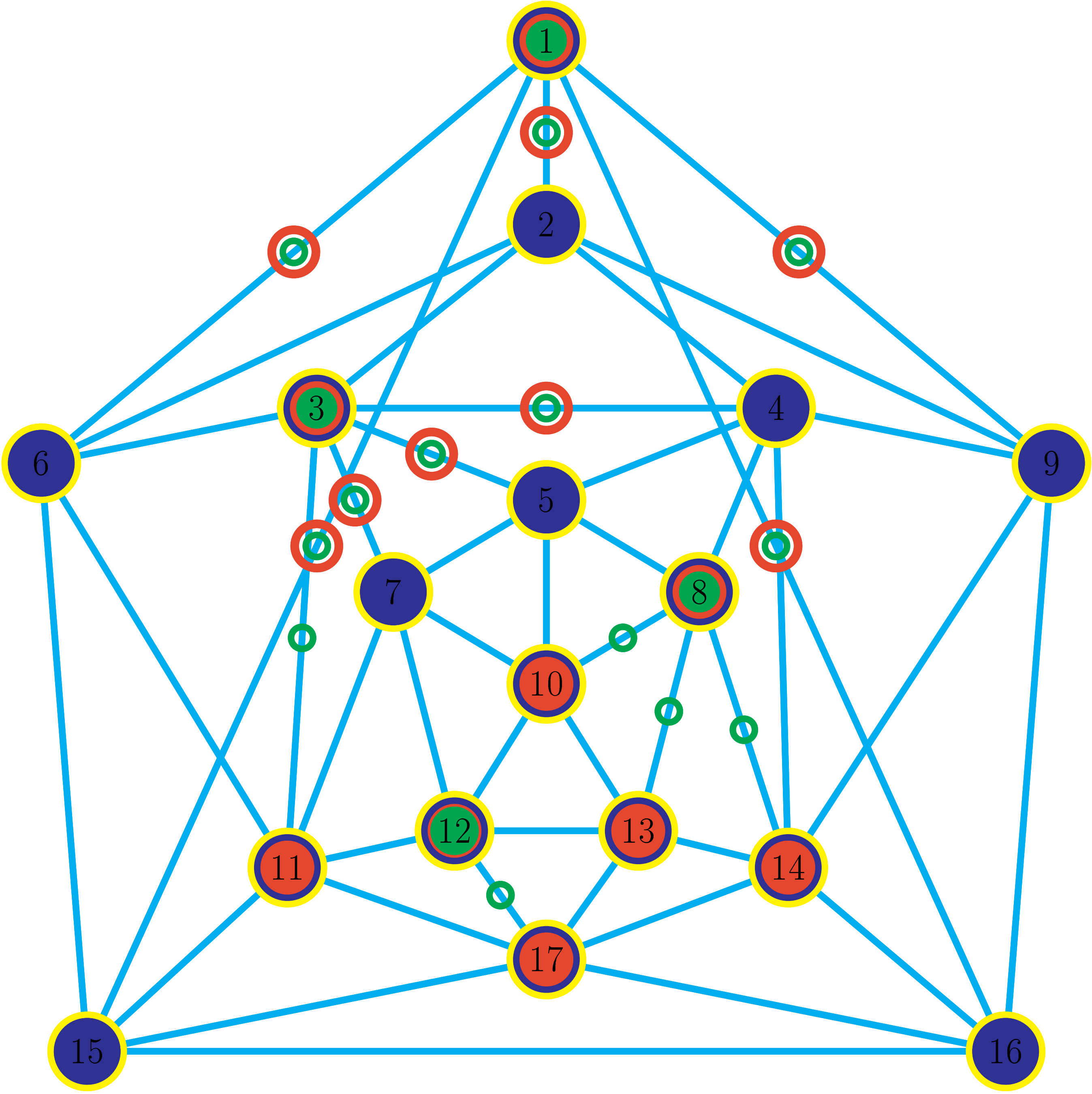


Figure 44.

instruction 82: unplace edge 3->7 Red DeletionArrow
instruction 83: place edge 3-7 Red Checker
instruction 84: unplace vertex 7 Red Checker;

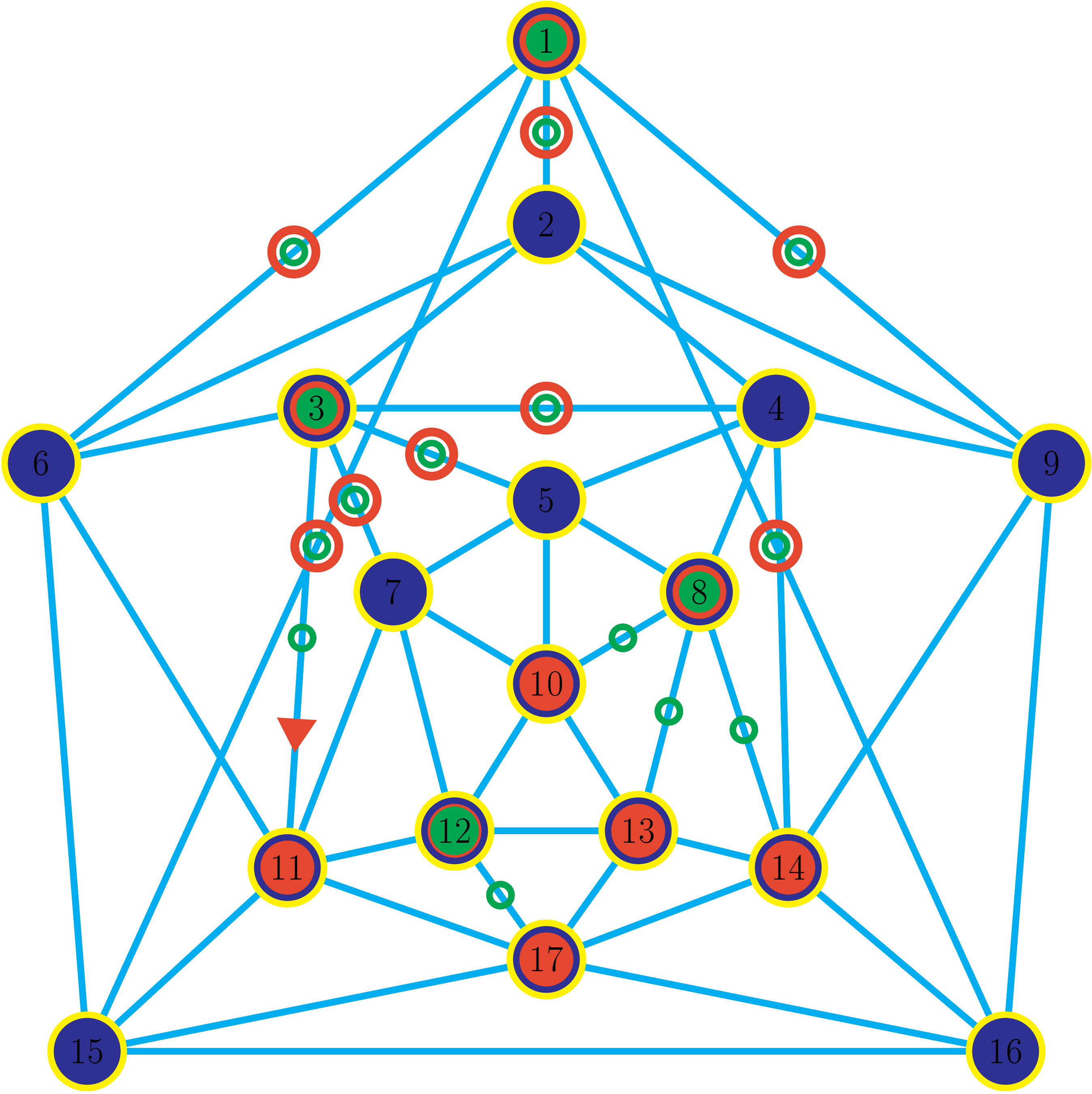


Figure 45.
instruction 85: place edge 3->11 Red DeletionArrow

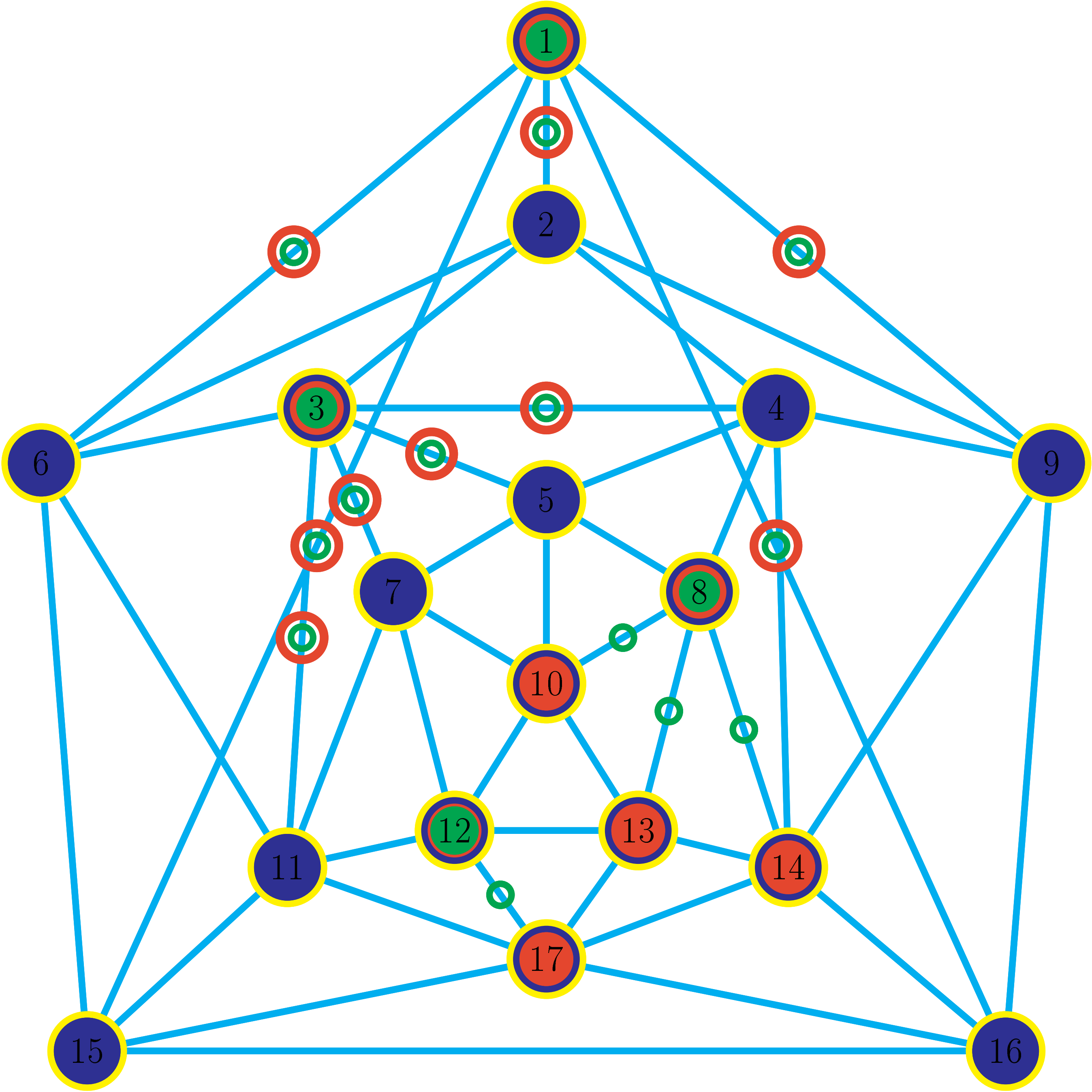


Figure 46.

instruction 86: unplace edge 3->11 Red DeletionArrow
instruction 87: place edge 3-11 Red Checker
instruction 88: unplace vertex 11 Red Checker;

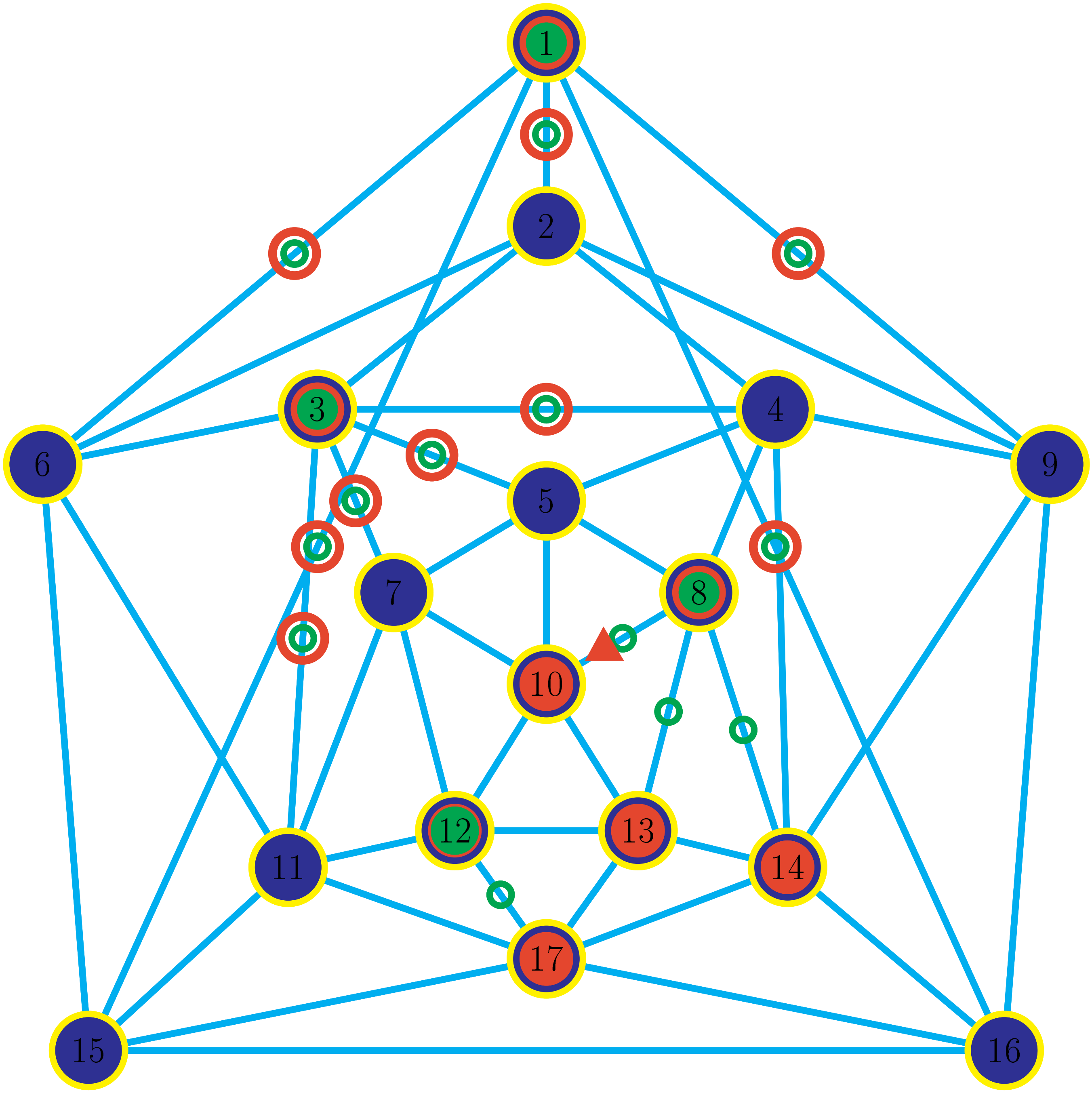


Figure 47.

instruction 89: place edge 8->10 Red DeletionArrow

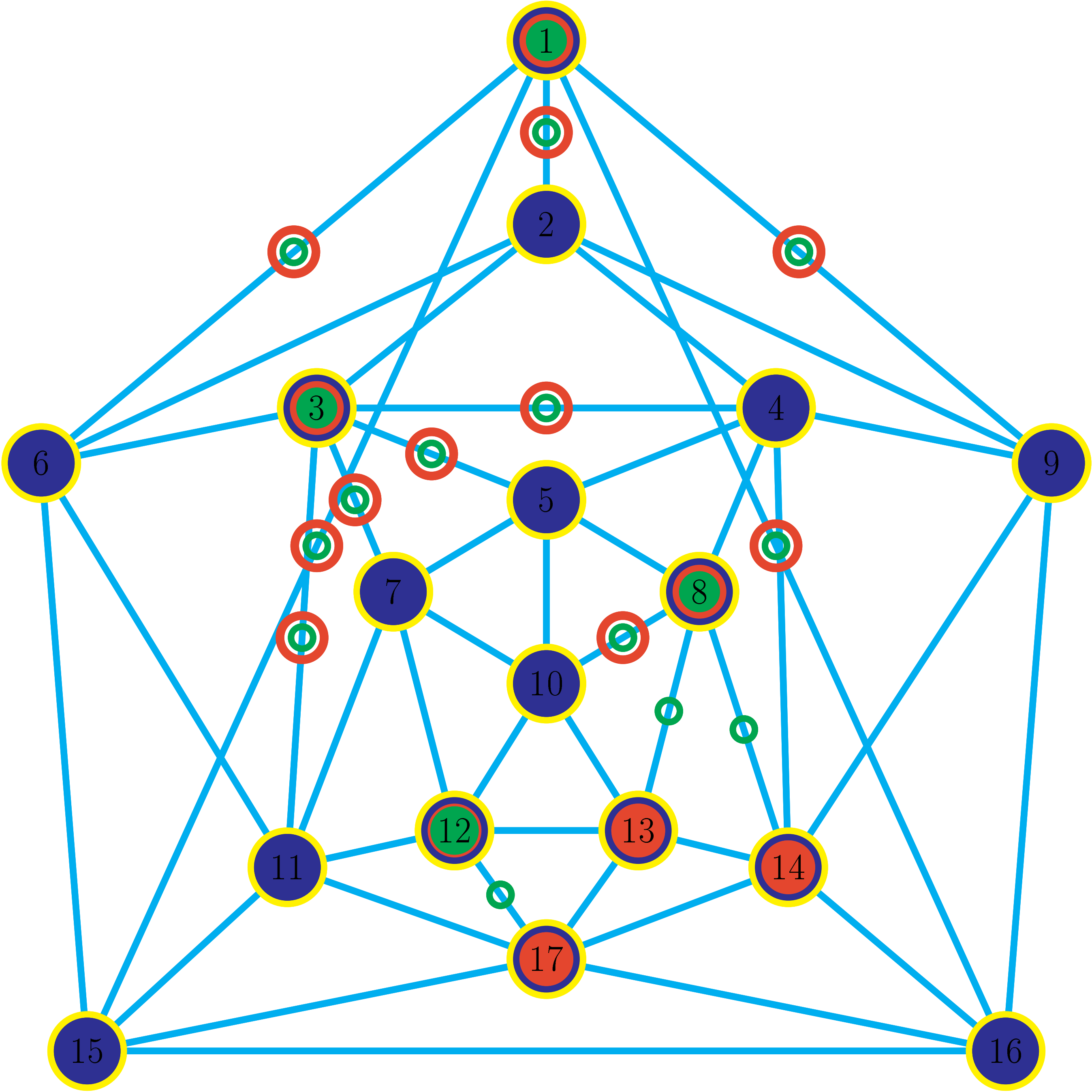


Figure 48.

instruction 90: unplace edge 8->10 Red DeletionArrow
instruction 91: place edge 8-10 Red Checker
instruction 92: unplace vertex 10 Red Checker;

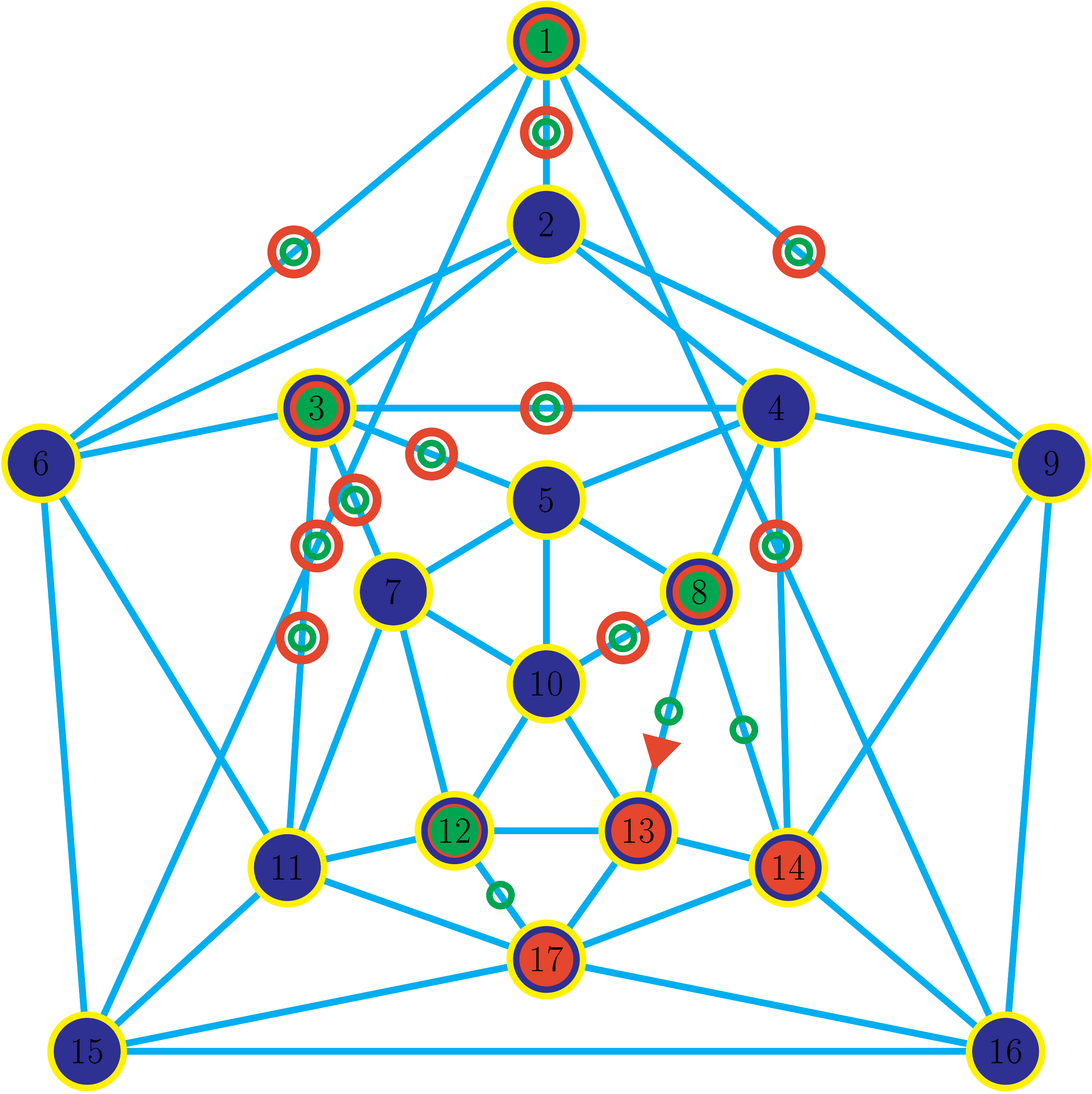


Figure 49.

instruction 93: place edge 8->13 Red DeletionArrow

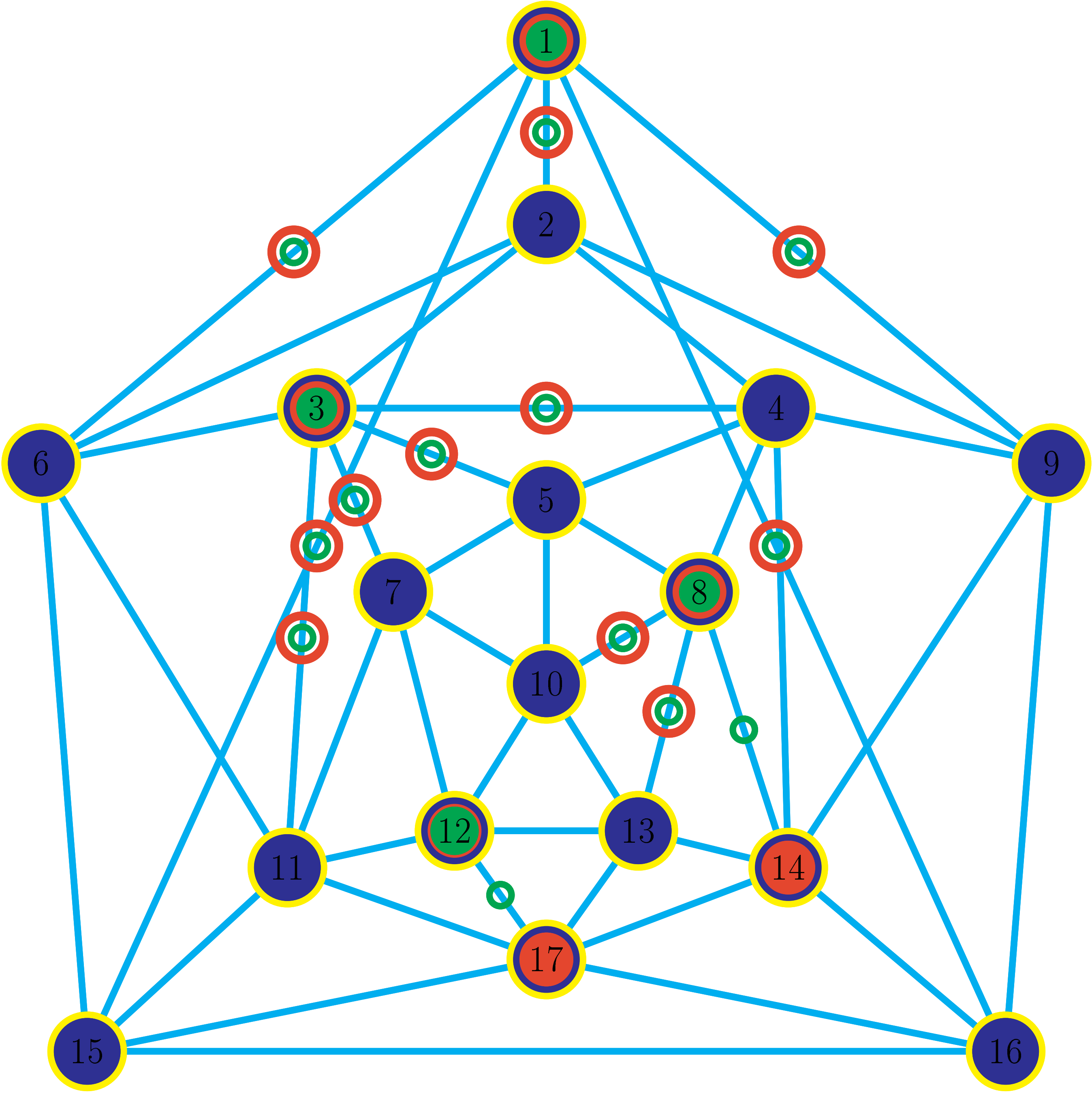


Figure 50.

instruction 94: unplace edge 8->13 Red DeletionArrow
instruction 95: place edge 8-13 Red Checker
instruction 96: unplace vertex 13 Red Checker;

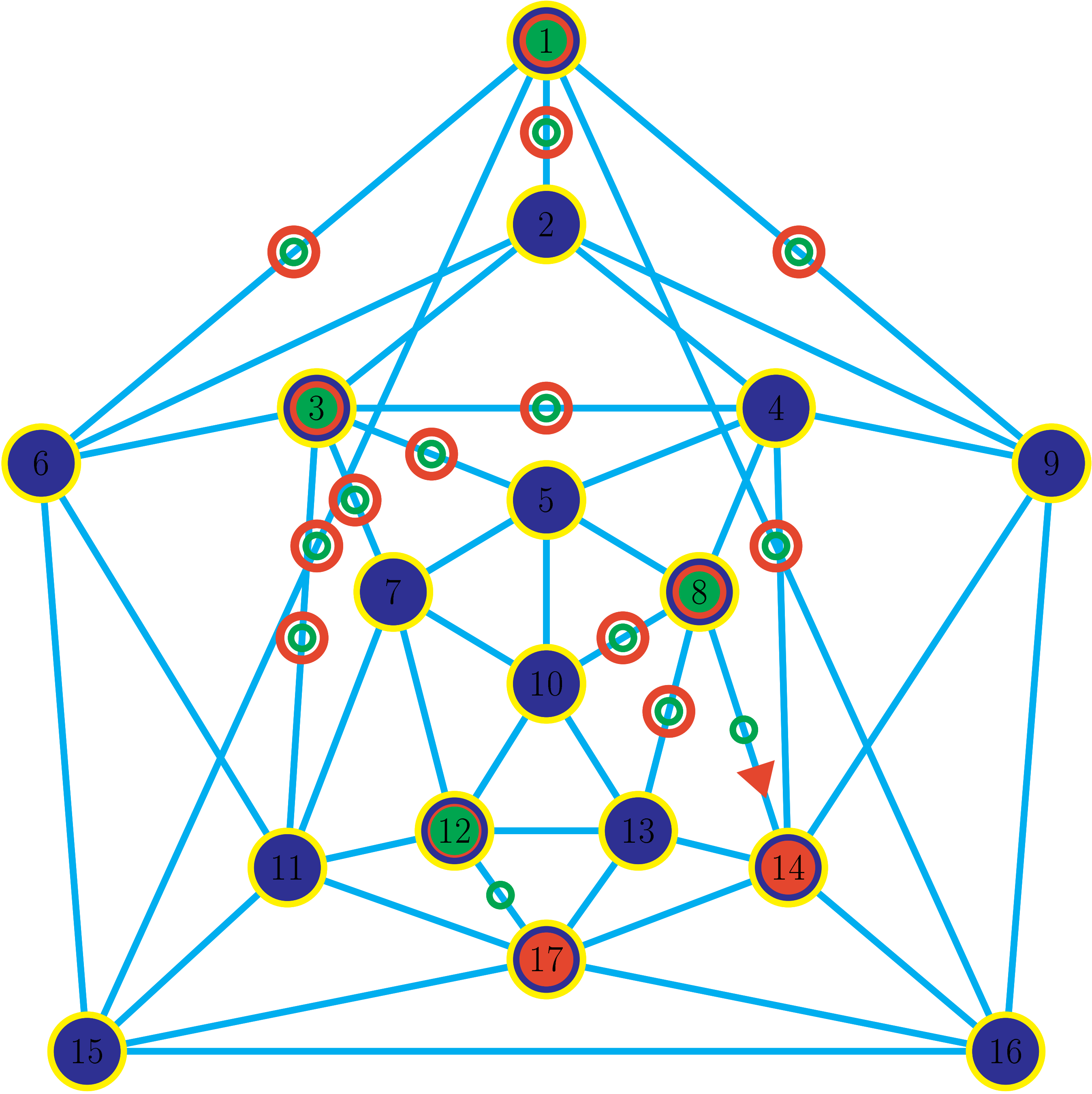


Figure 51.

instruction 97: place edge 8->14 Red DeletionArrow

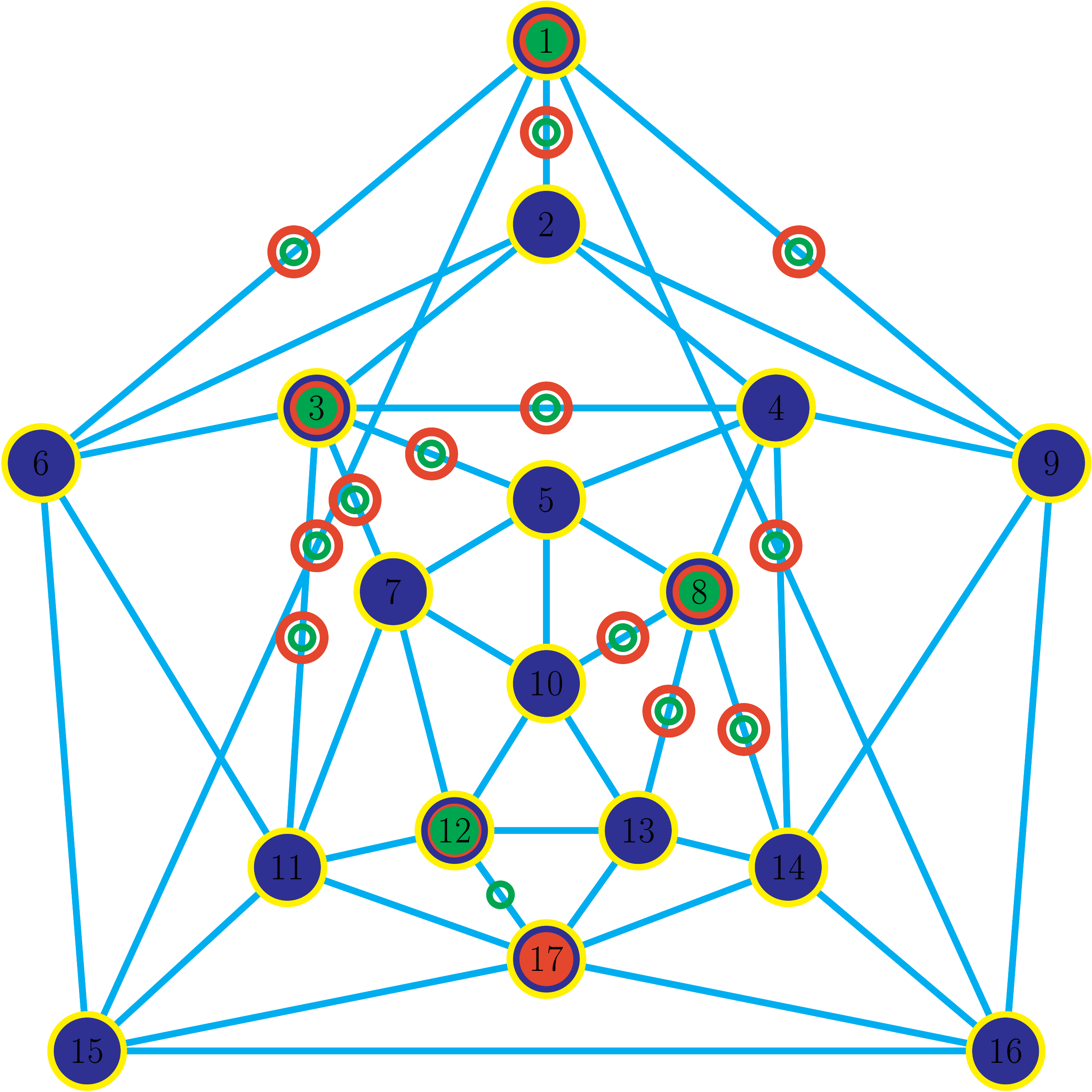


Figure 52.

instruction 98: unplace edge 8->14 Red DeletionArrow
instruction 99: place edge 8-14 Red Checker
instruction 100: unplace vertex 14 Red Checker;

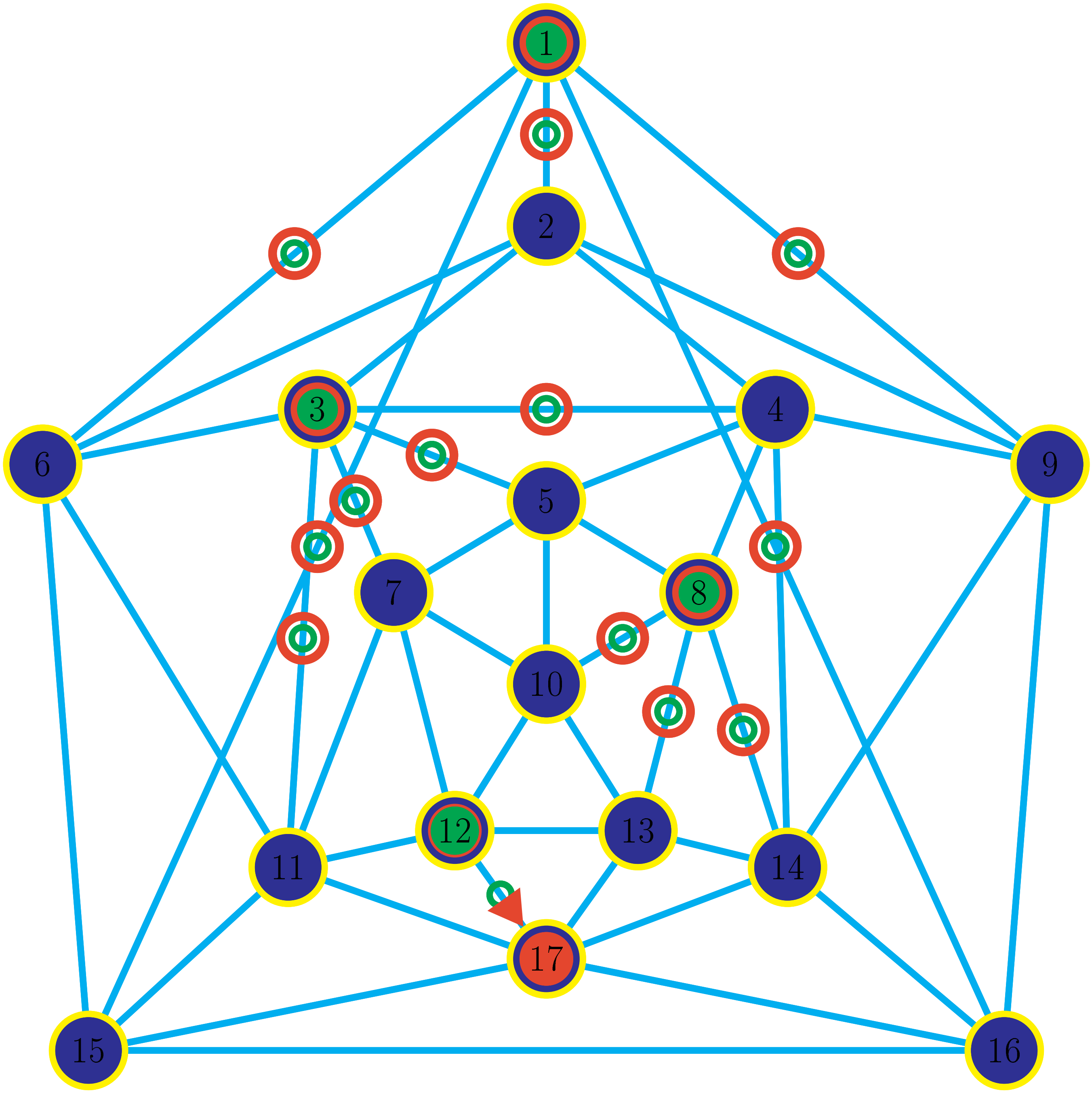


Figure 53.

instruction 101: place edge 12->17 Red DeletionArrow

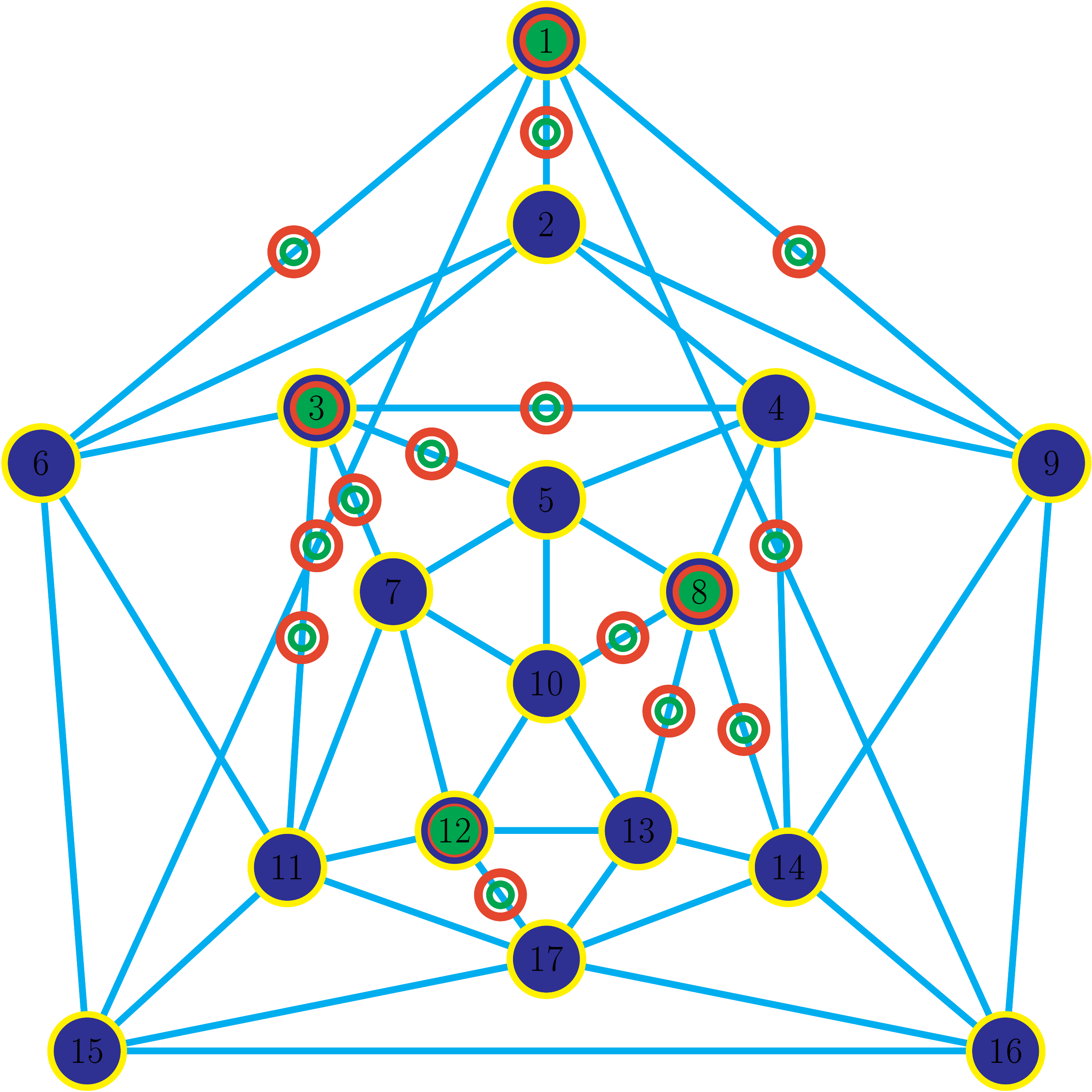


Figure 54.

instruction 102: unplace edge 12->17 Red DeletionArrow
instruction 103: place edge 12-17 Red Checker
instruction 104: unplace vertex 17 Red Checker;

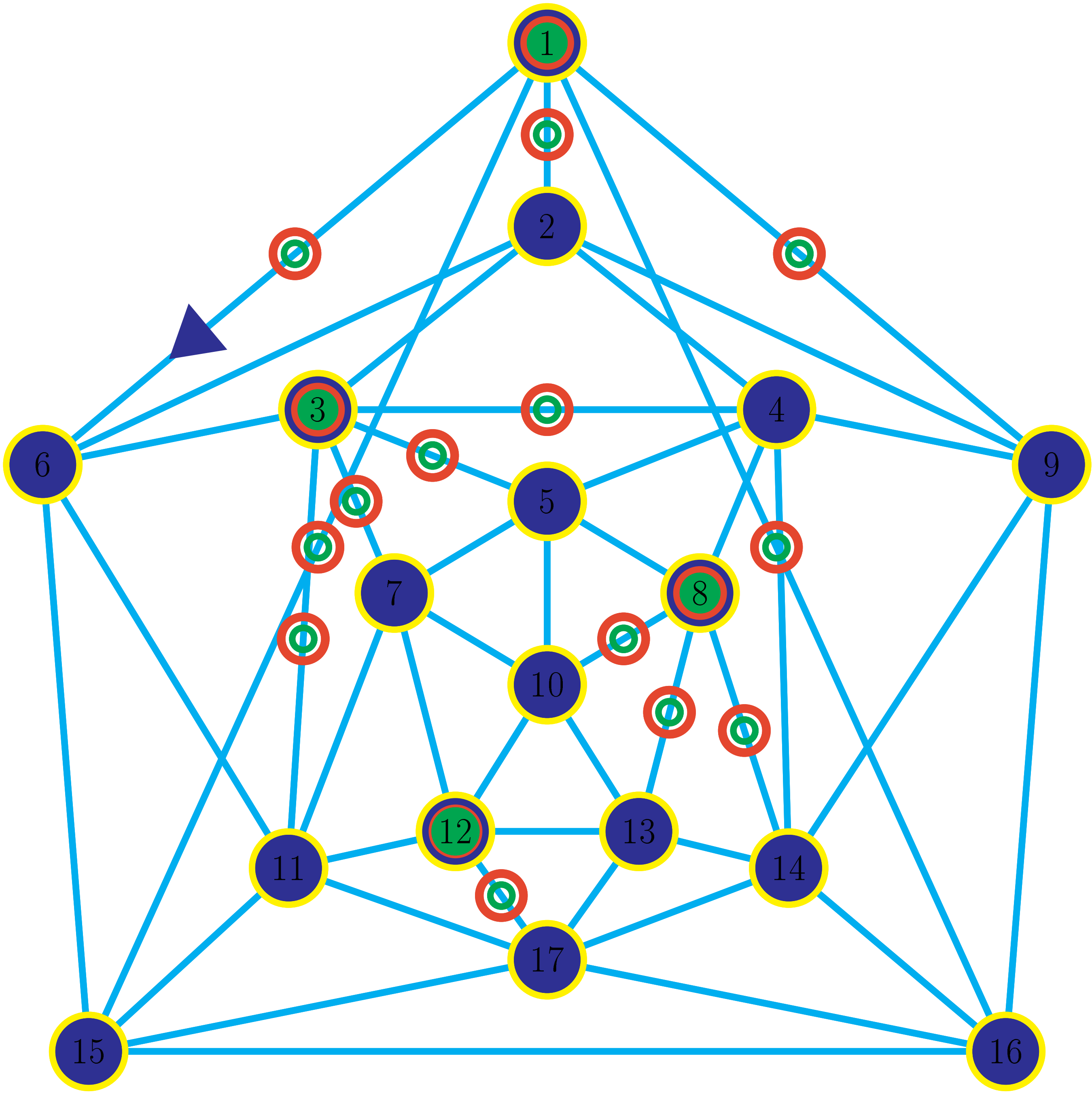


Figure 55.

instruction 105: place edge 1->6 Blue DeletionArrow

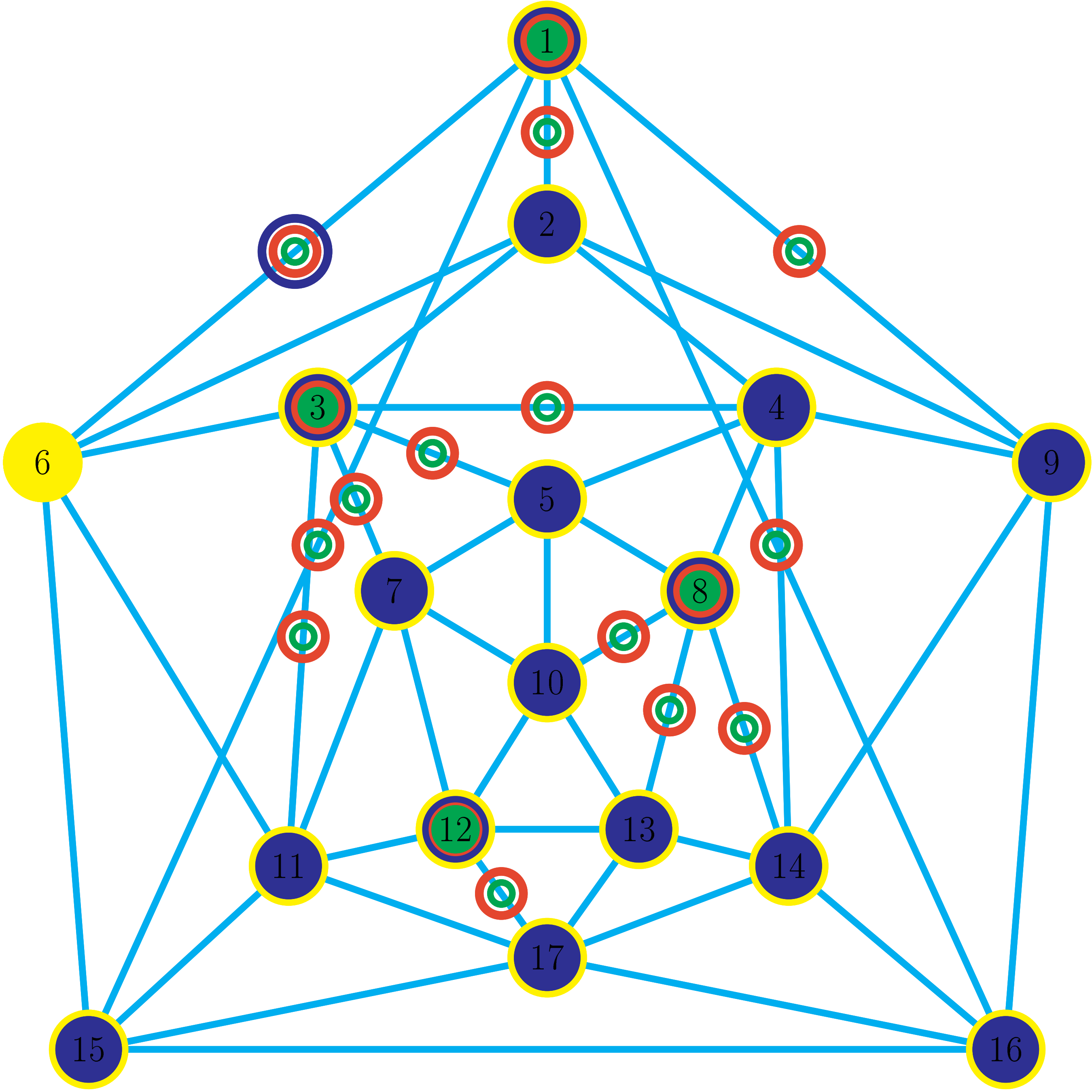


Figure 56.

instruction 106: unplace edge 1->6 Blue DeletionArrow
instruction 107: place edge 1-6 Blue Checker
instruction 108: unplace vertex 6 Blue Checker;

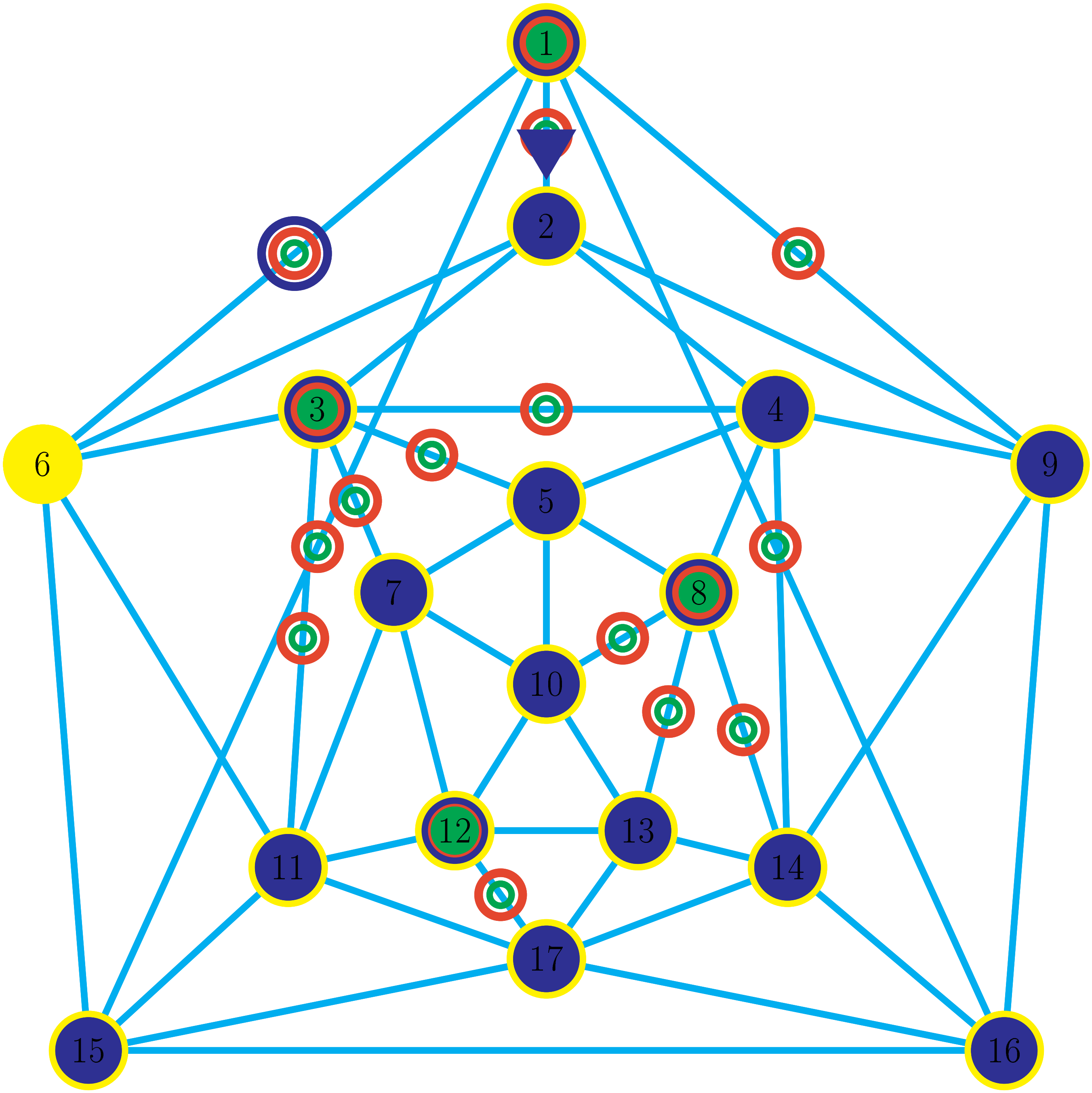


Figure 57.

instruction 109: place edge 1->2 Blue DeletionArrow

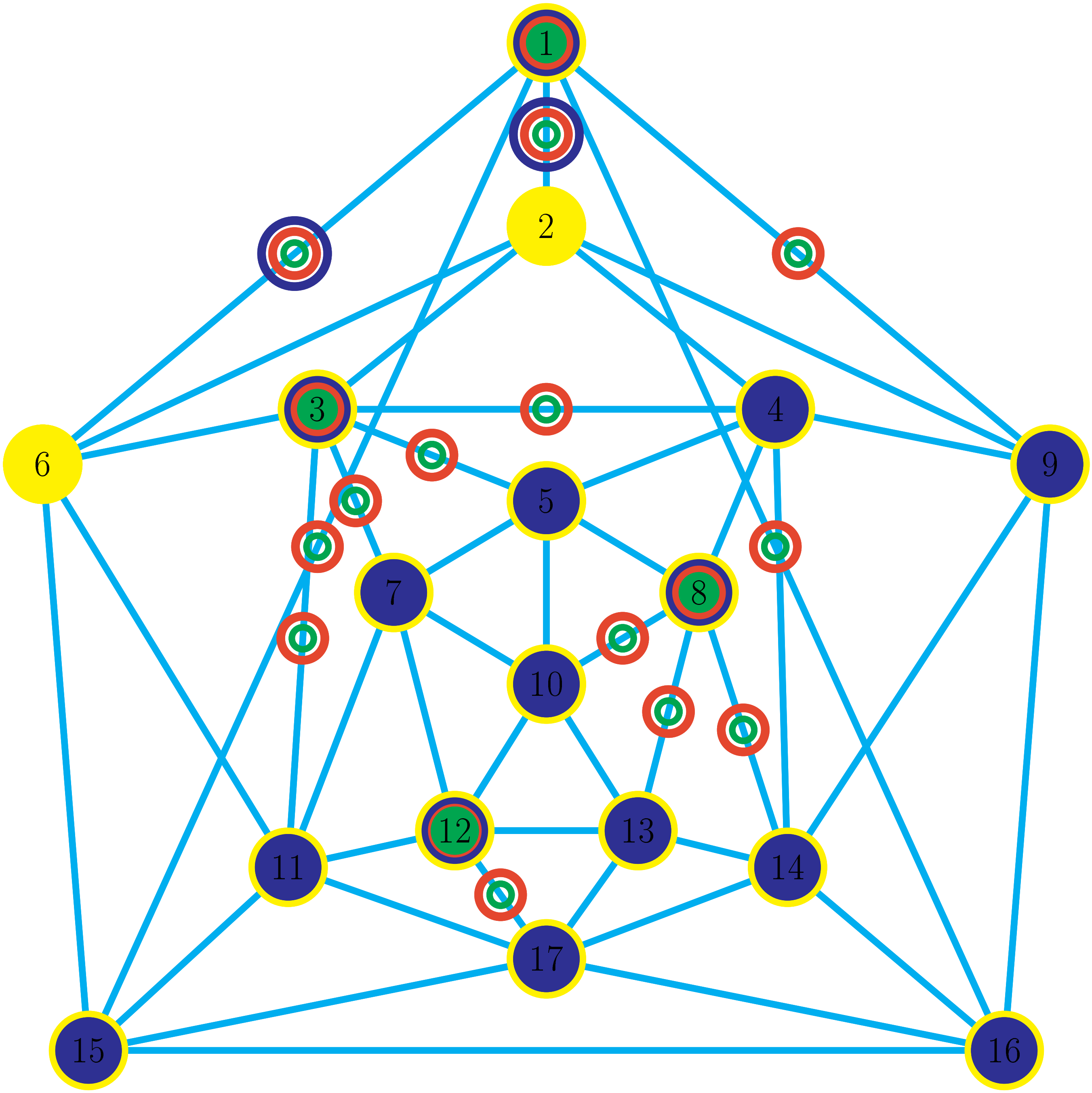


Figure 58.

instruction 110: unplace edge 1->2 Blue DeletionArrow
instruction 111: place edge 1-2 Blue Checker
instruction 112: unplace vertex 2 Blue Checker;

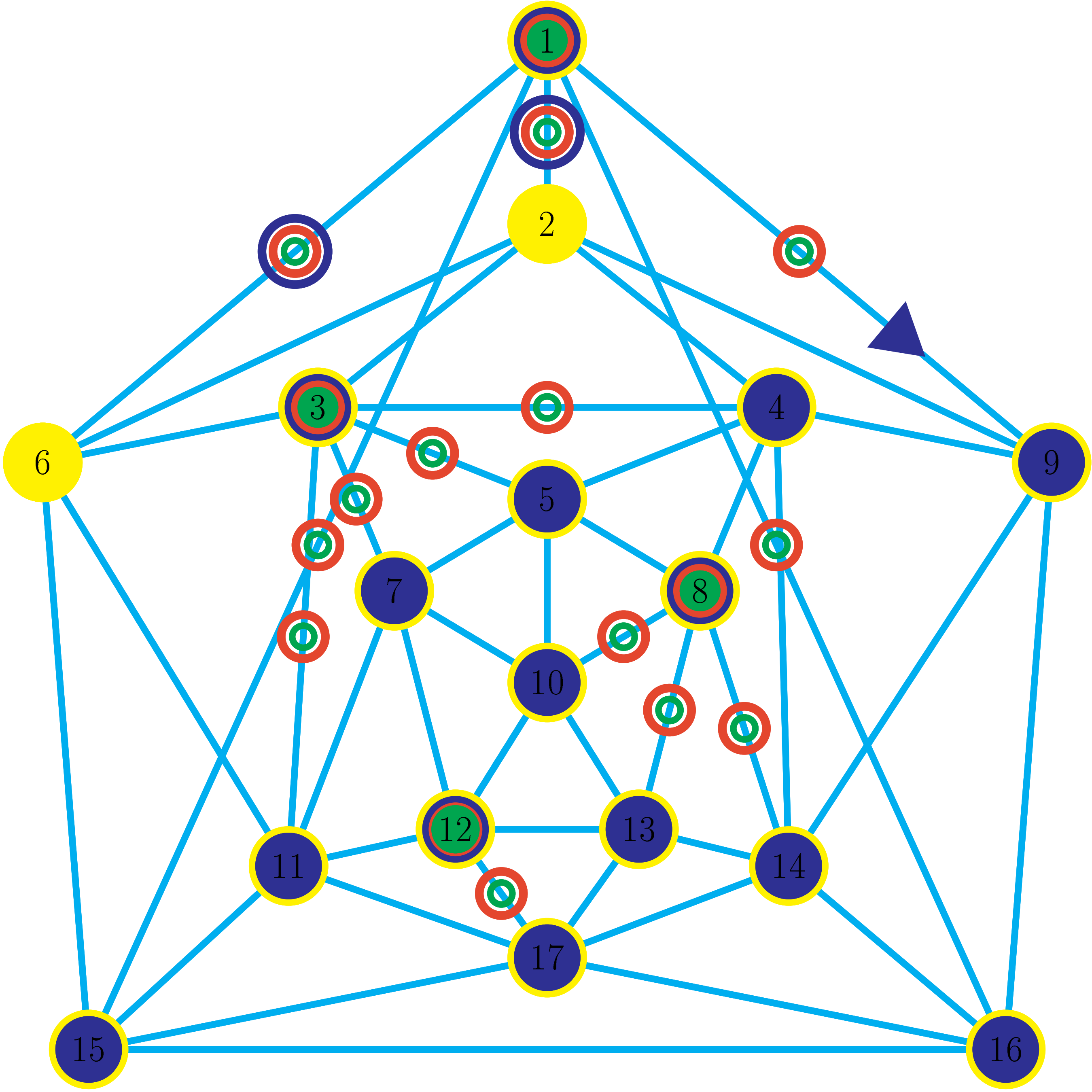


Figure 59.

instruction 113: place edge 1->9 Blue DeletionArrow

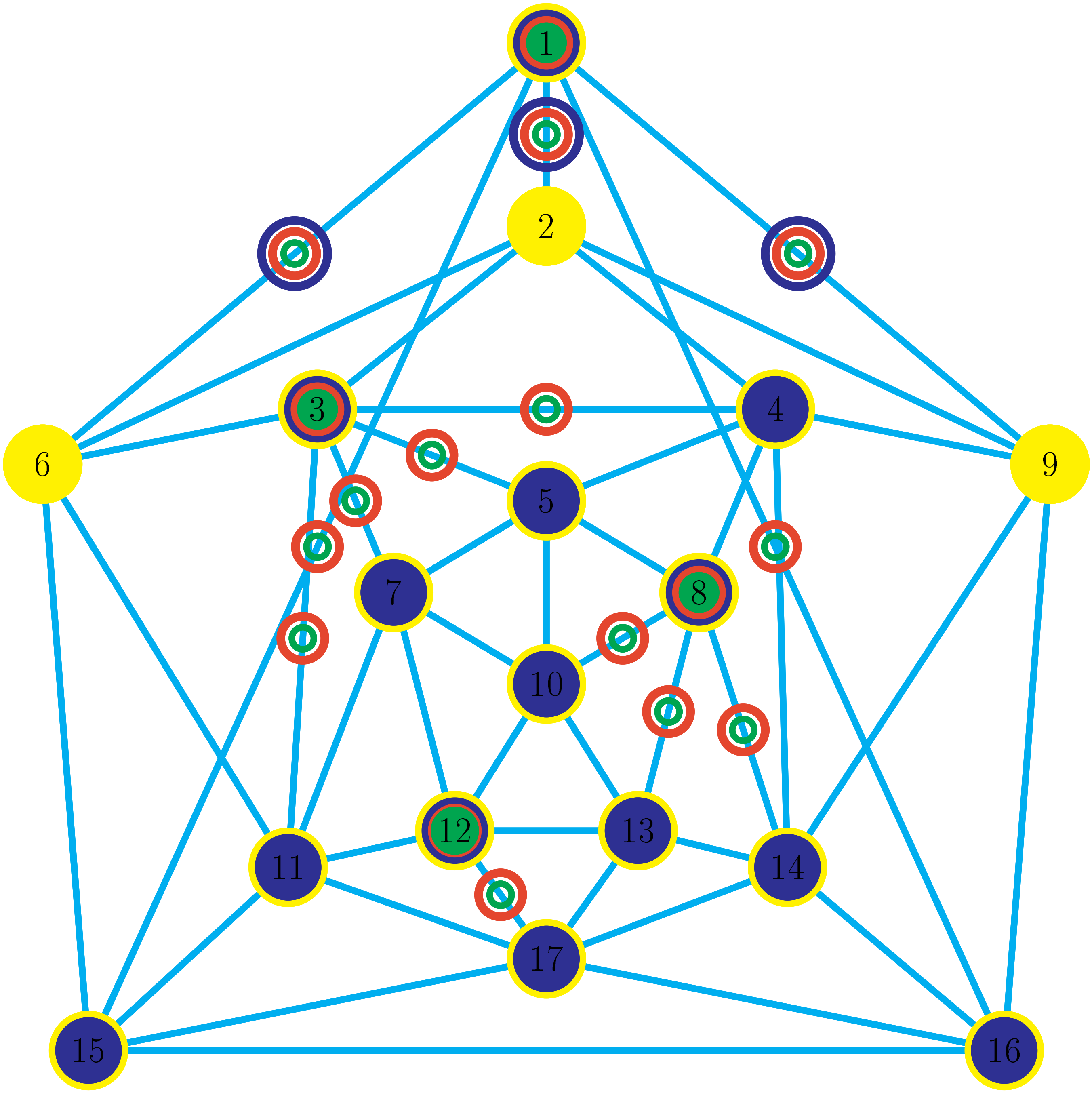


Figure 60.

instruction 114: unplace edge 1->9 Blue DeletionArrow
instruction 115: place edge 1-9 Blue Checker
instruction 116: unplace vertex 9 Blue Checker;

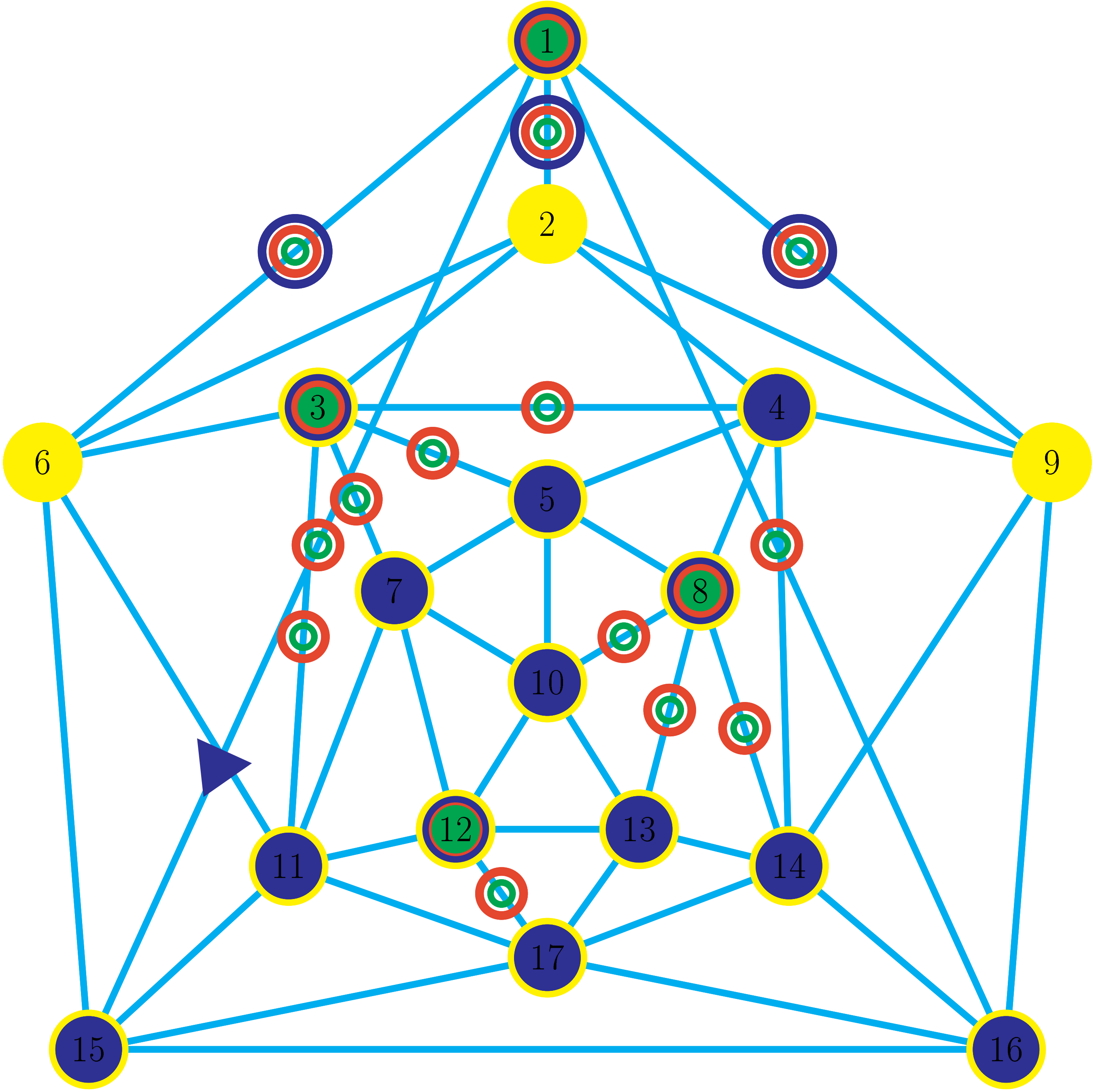


Figure 61.

instruction 117: place edge 1->15 Blue DeletionArrow

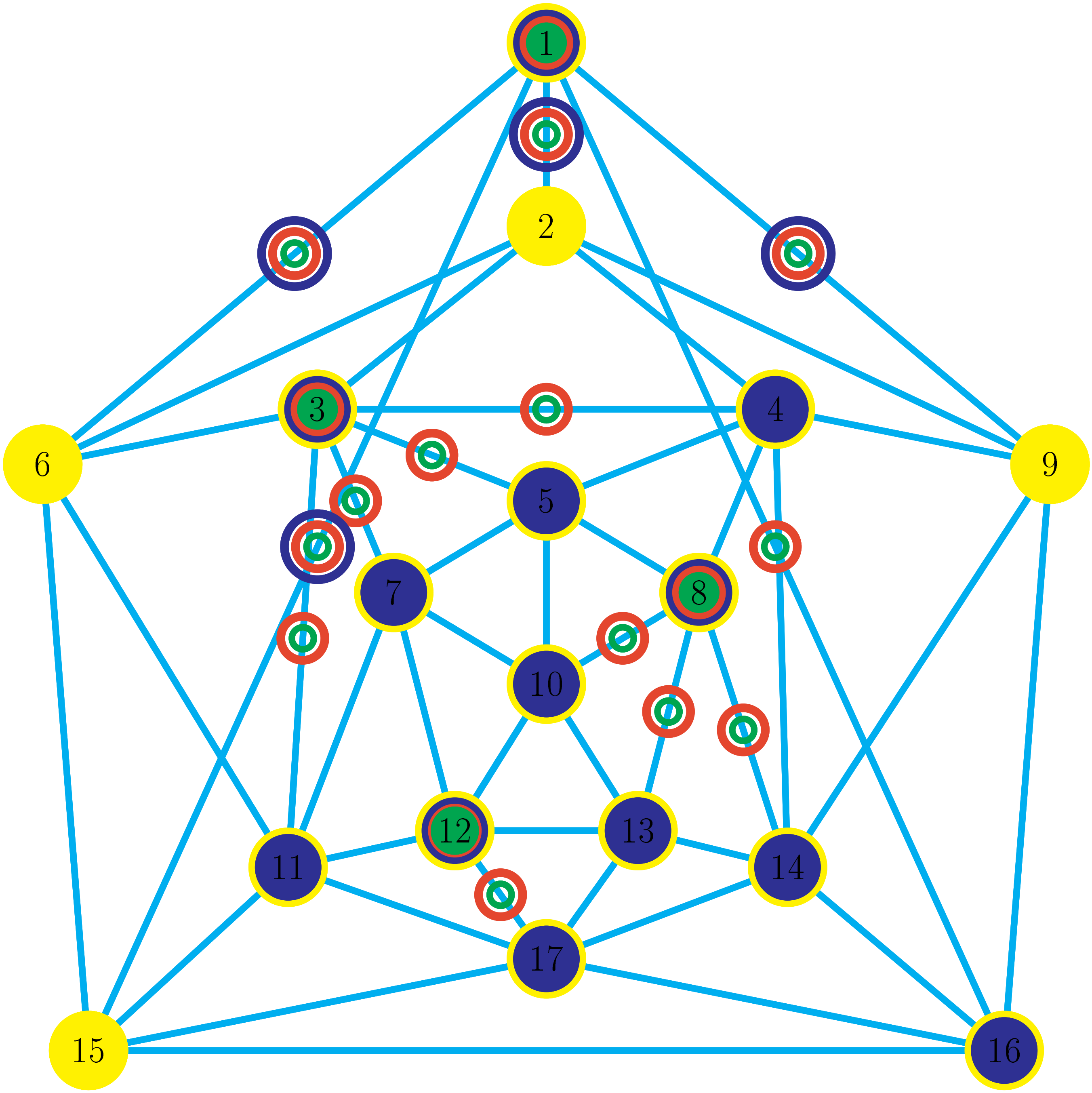


Figure 62.

instruction 118: unplace edge 1->15 Blue DeletionArrow
instruction 119: place edge 1-15 Blue Checker
instruction 120: unplace vertex 15 Blue Checker;

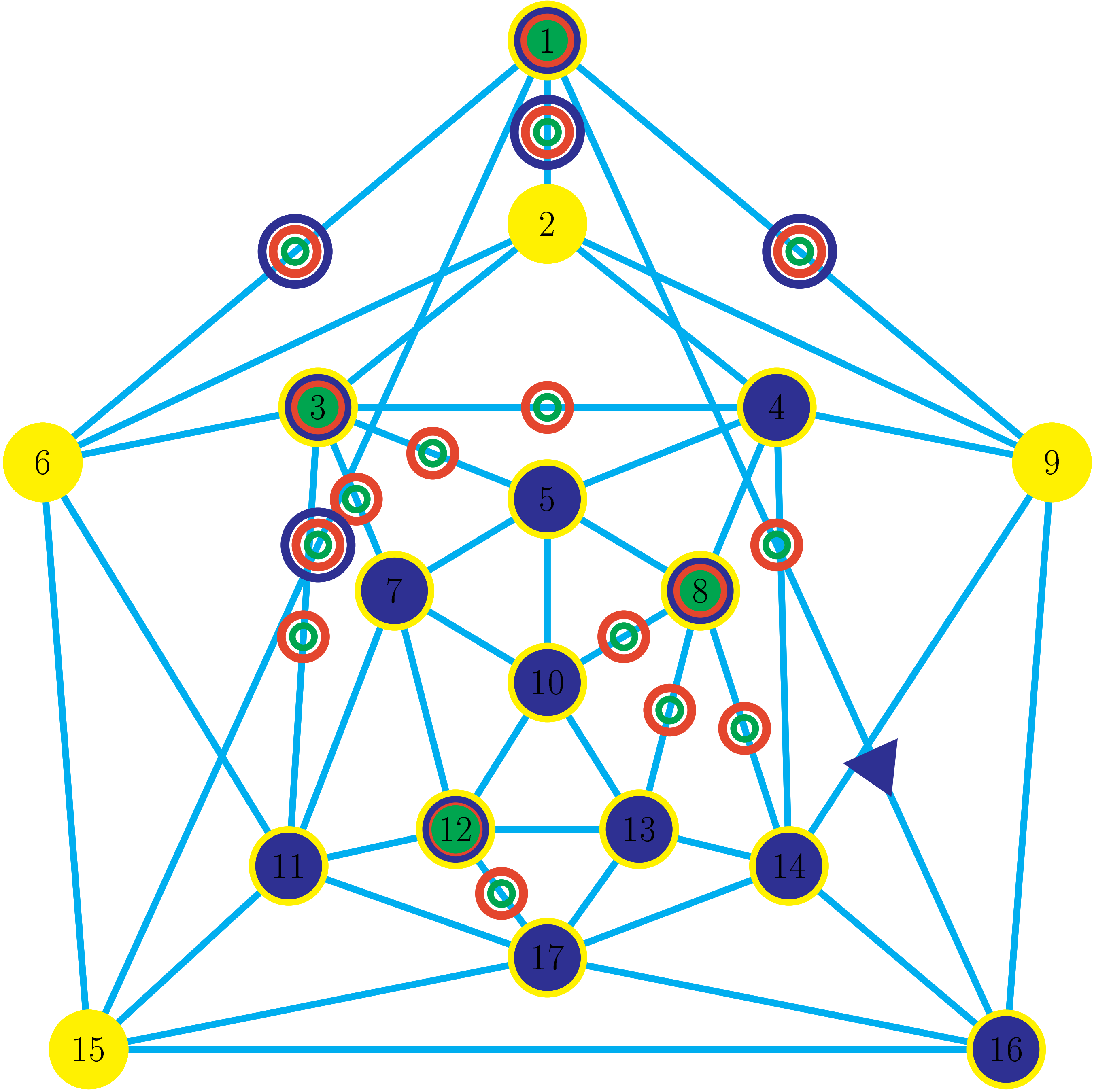


Figure 63.

instruction 121: place edge 1->16 Blue DeletionArrow

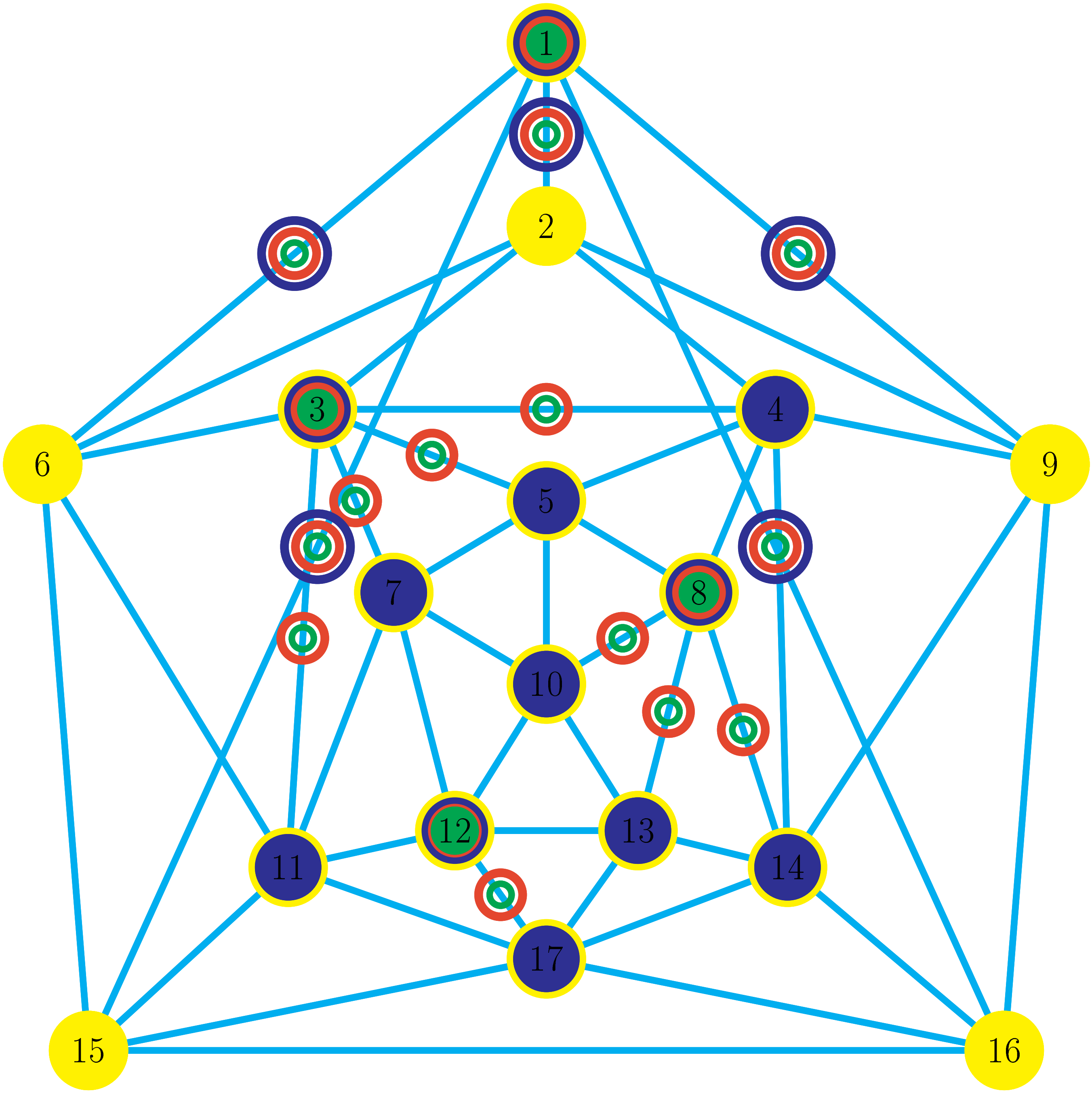


Figure 64.

instruction 122: unplace edge 1->16 Blue DeletionArrow
instruction 123: place edge 1-16 Blue Checker
instruction 124: unplace vertex 16 Blue Checker;

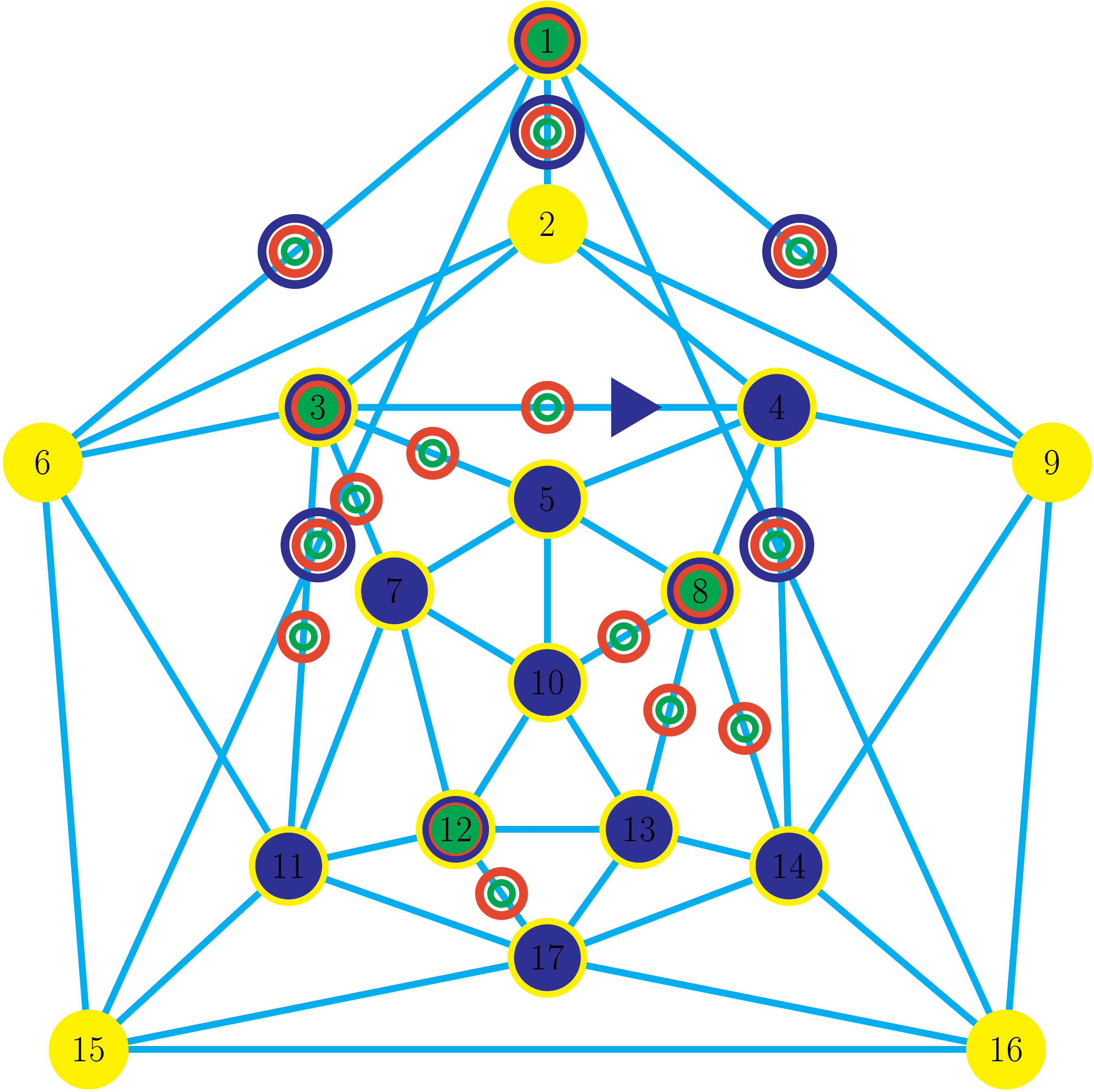


Figure 65.

instruction 125: place edge 3->4 Blue DeletionArrow

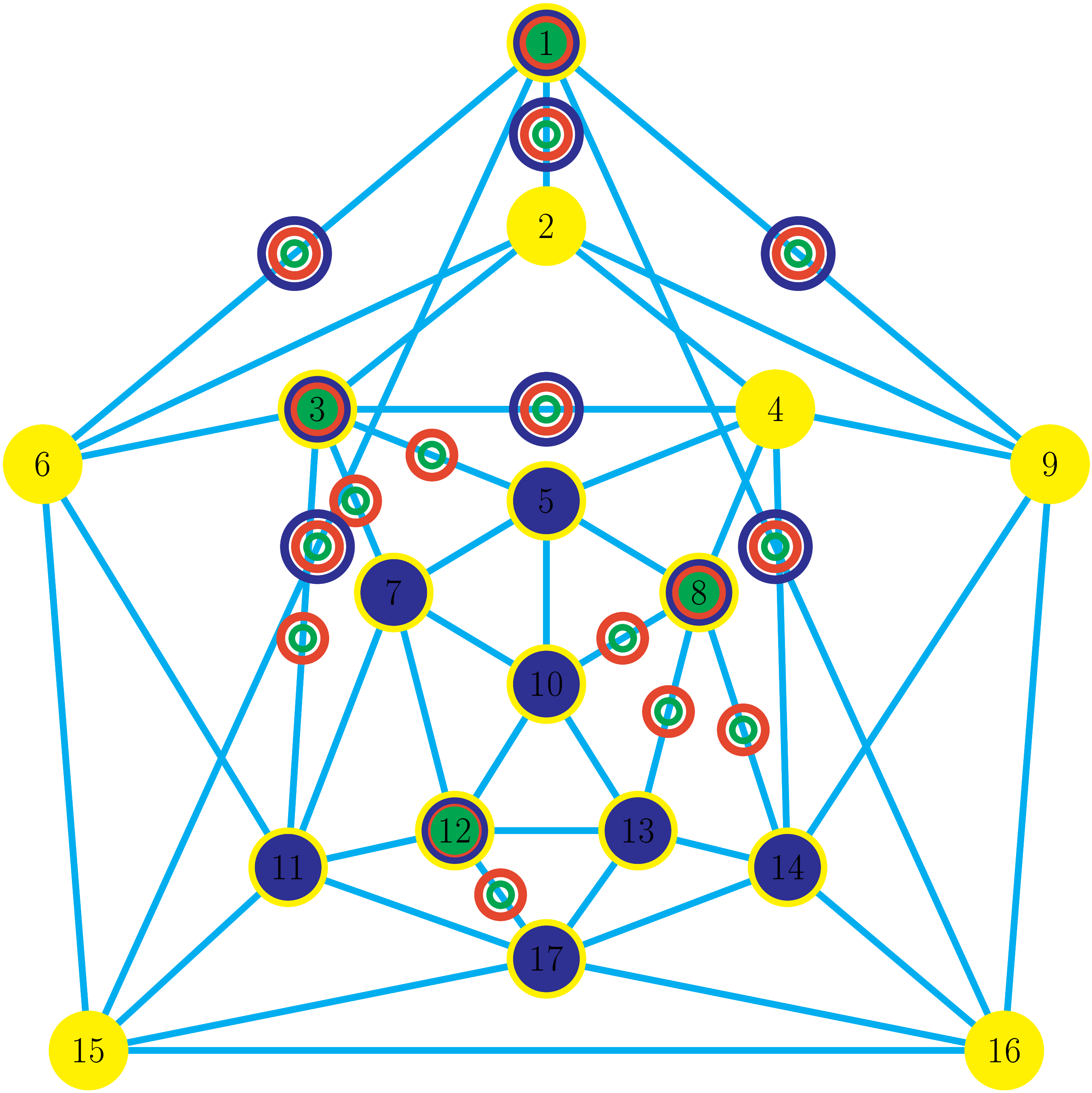


Figure 66.

instruction 126: unplace edge 3->4 Blue DeletionArrow
instruction 127: place edge 3-4 Blue Checker
instruction 128: unplace vertex 4 Blue Checker;

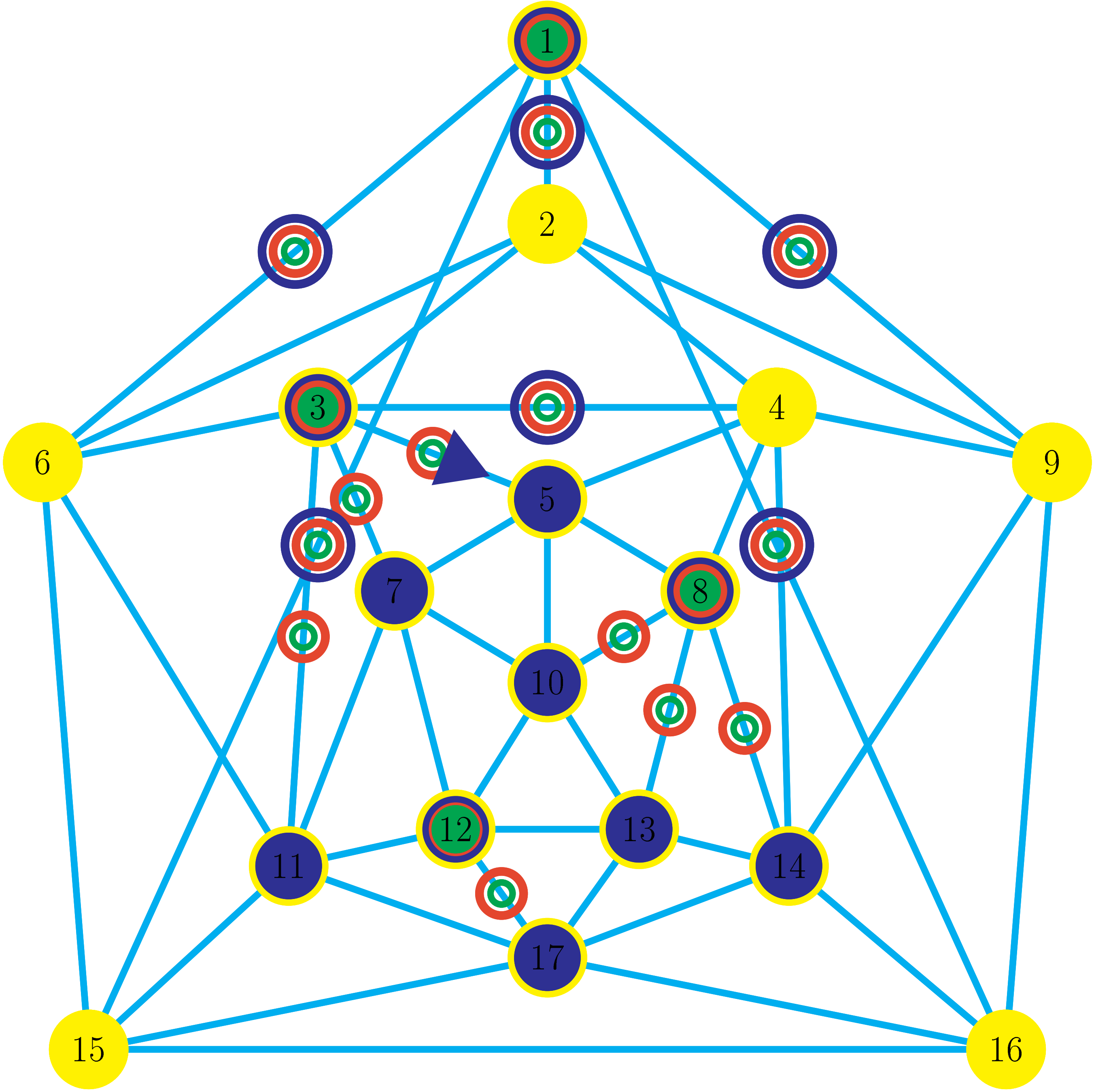


Figure 67.

instruction 129: place edge 3->5 Blue DeletionArrow

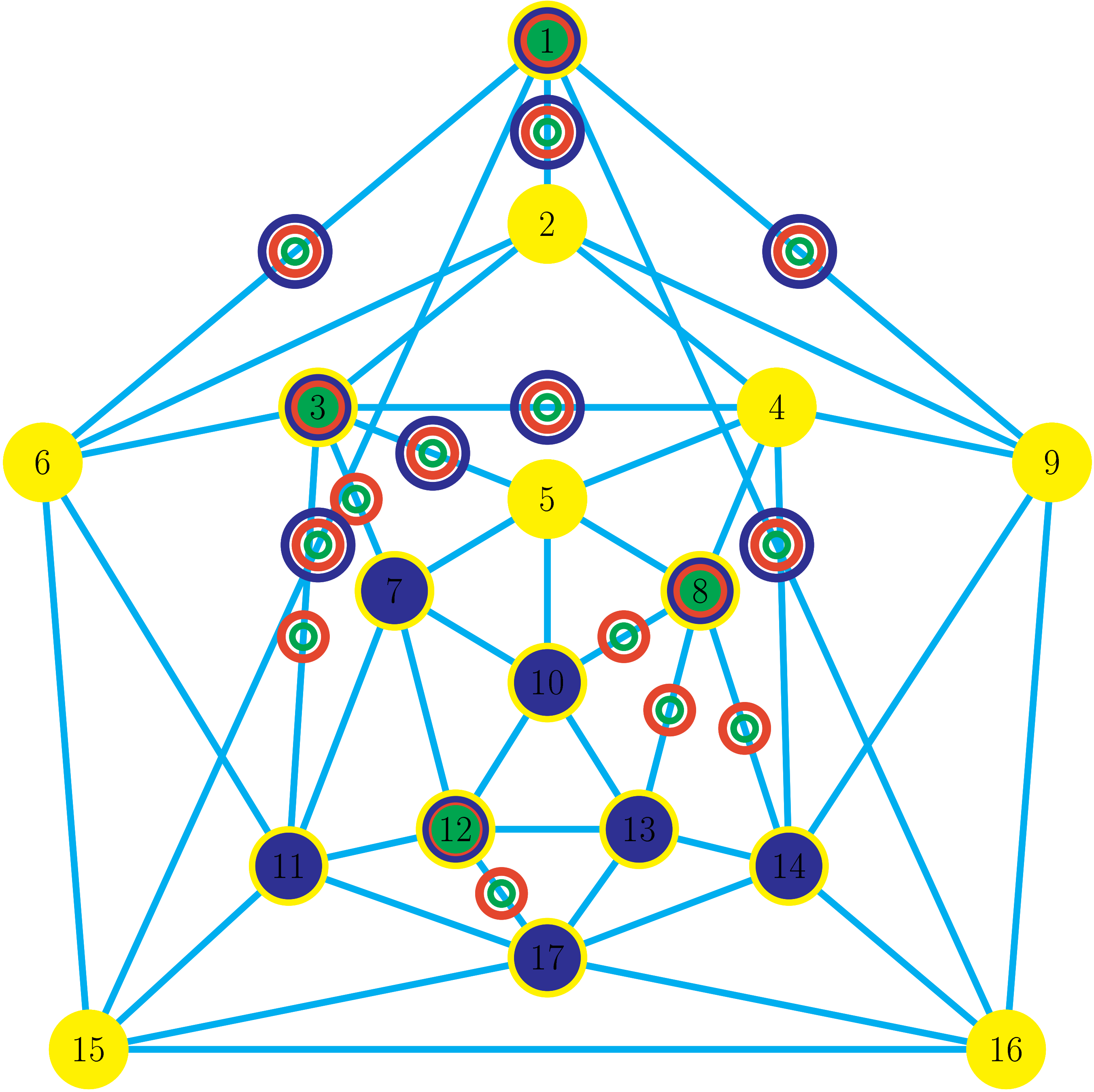


Figure 68.

instruction 130: unplace edge 3->5 Blue DeletionArrow
instruction 131: place edge 3-5 Blue Checker
instruction 132: unplace vertex 5 Blue Checker;

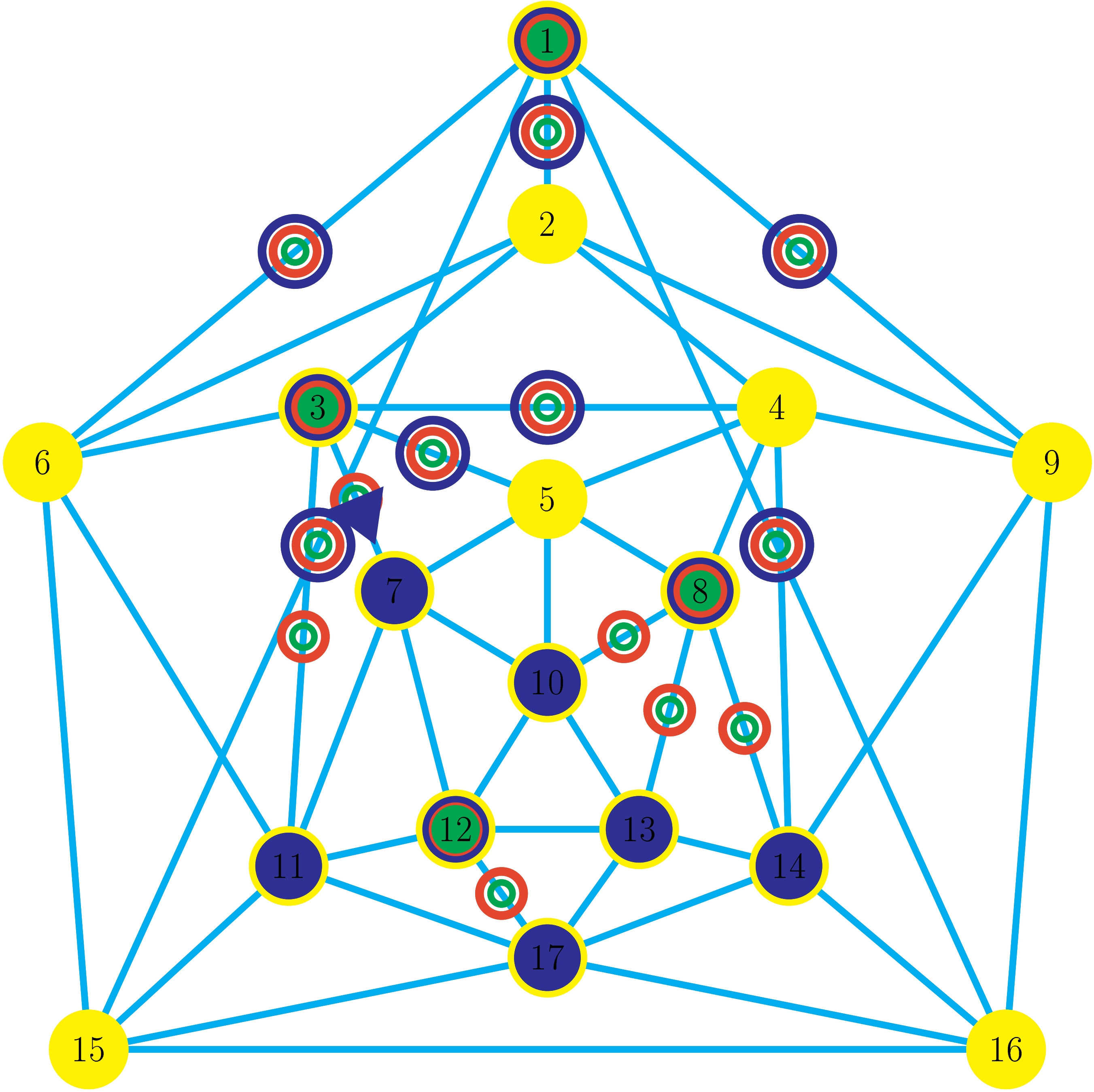


Figure 69.

instruction 133: place edge 3->7 Blue DeletionArrow

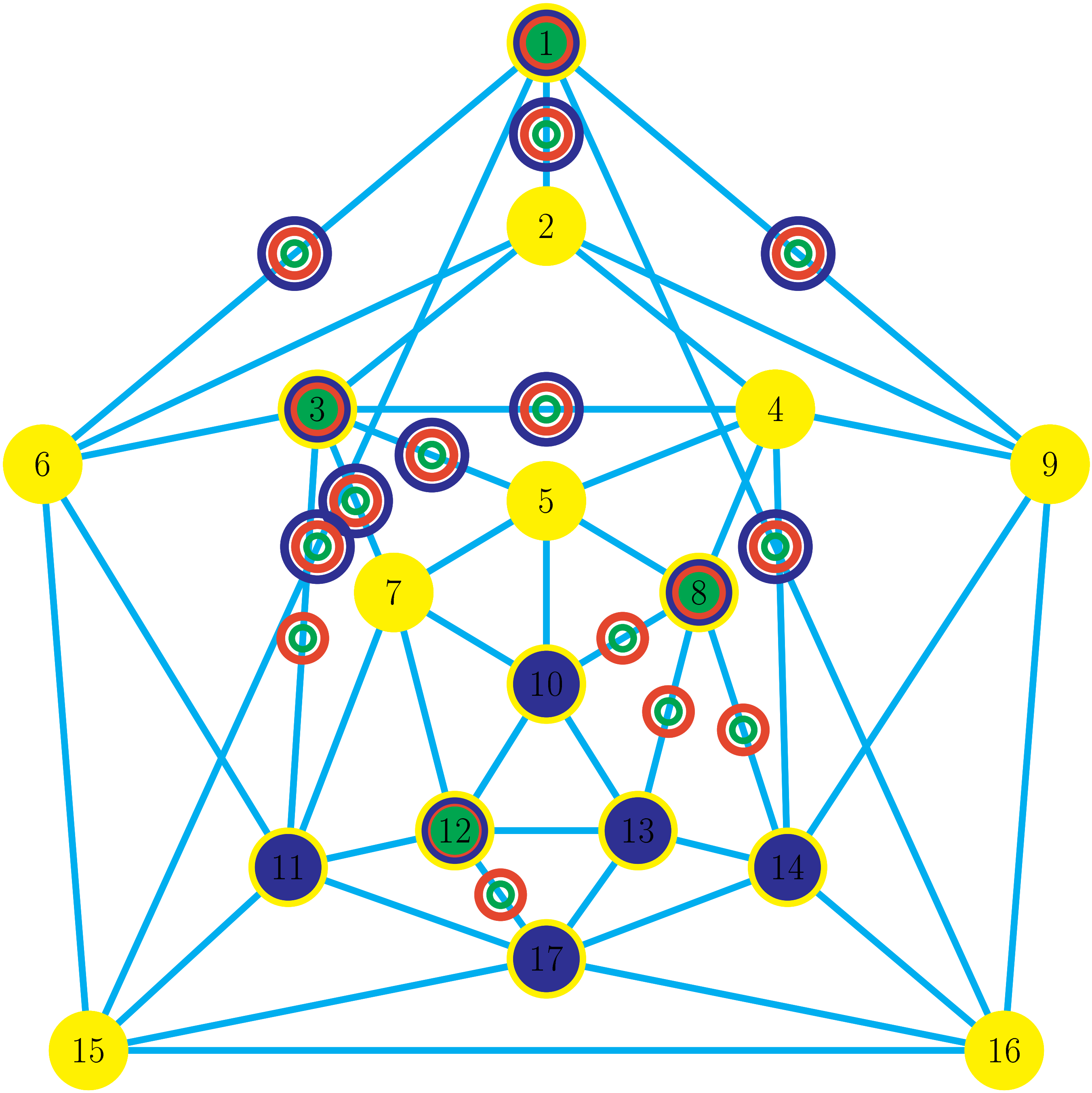


Figure 70.

instruction 134: unplace edge 3->7 Blue DeletionArrow
instruction 135: place edge 3-7 Blue Checker
instruction 136: unplace vertex 7 Blue Checker;

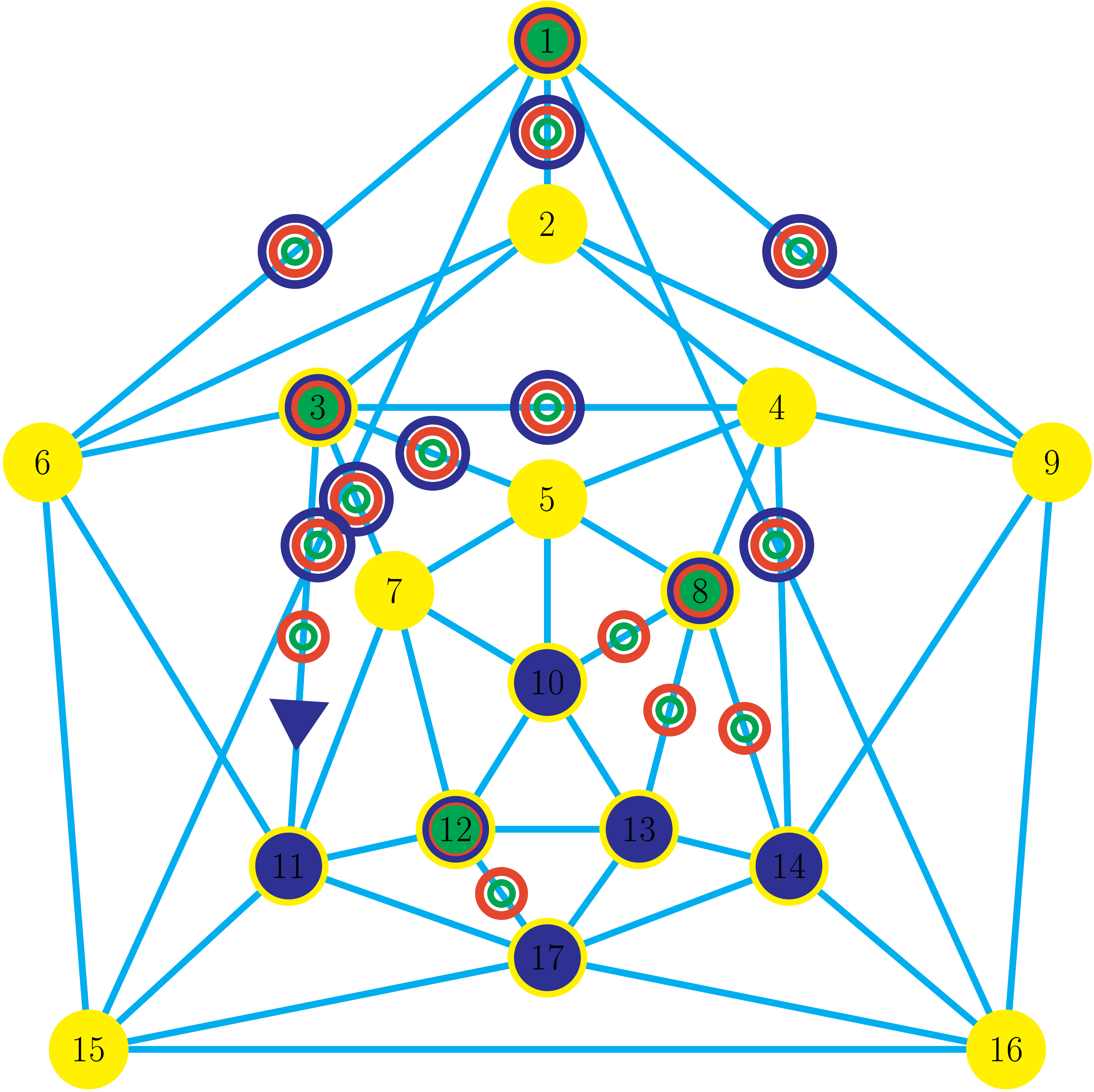


Figure 71.

instruction 137: place edge 3->11 Blue DeletionArrow

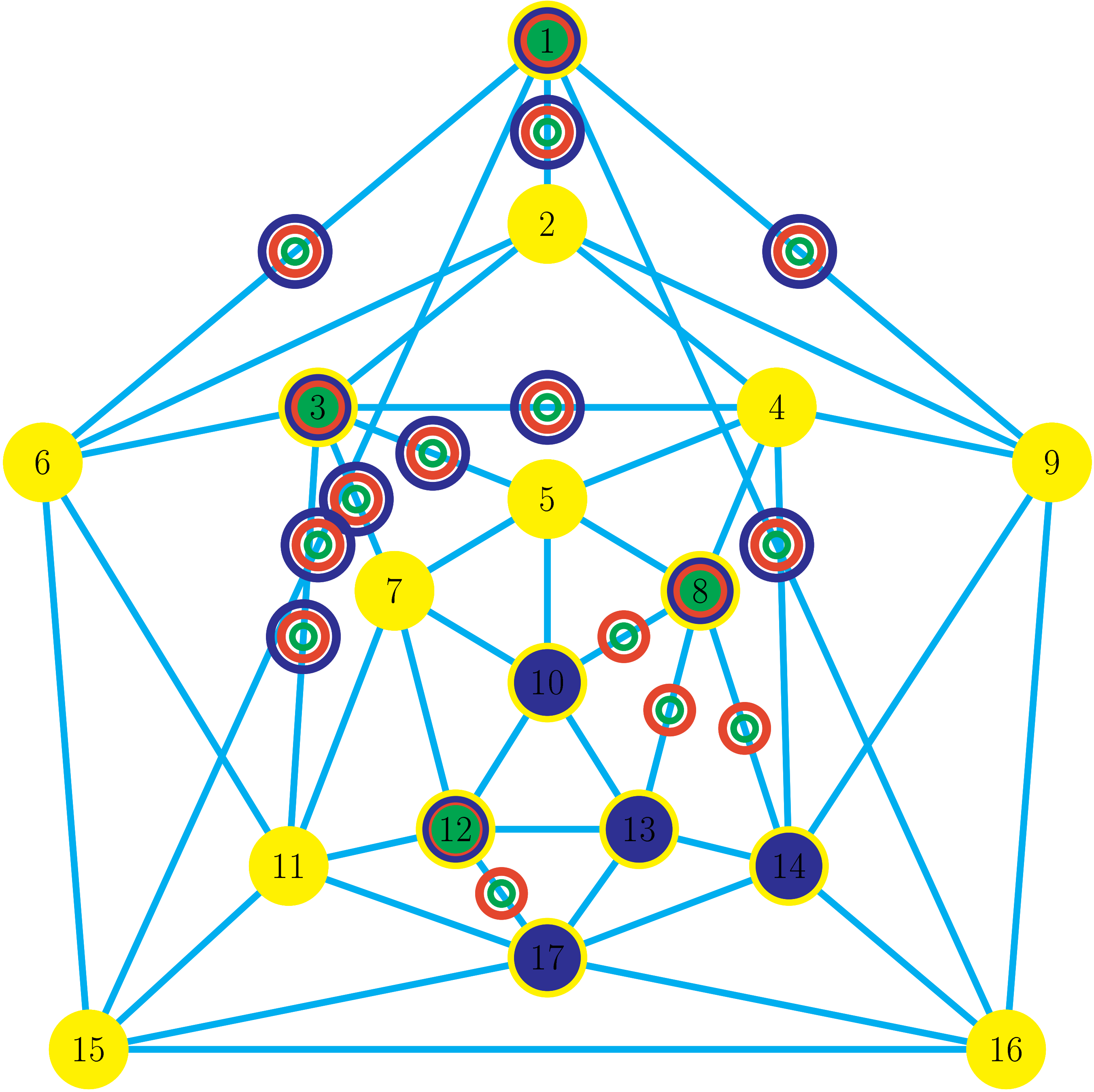


Figure 72.

instruction 138: unplace edge 3->11 Blue DeletionArrow
instruction 139: place edge 3-11 Blue Checker
instruction 140: unplace vertex 11 Blue Checker;

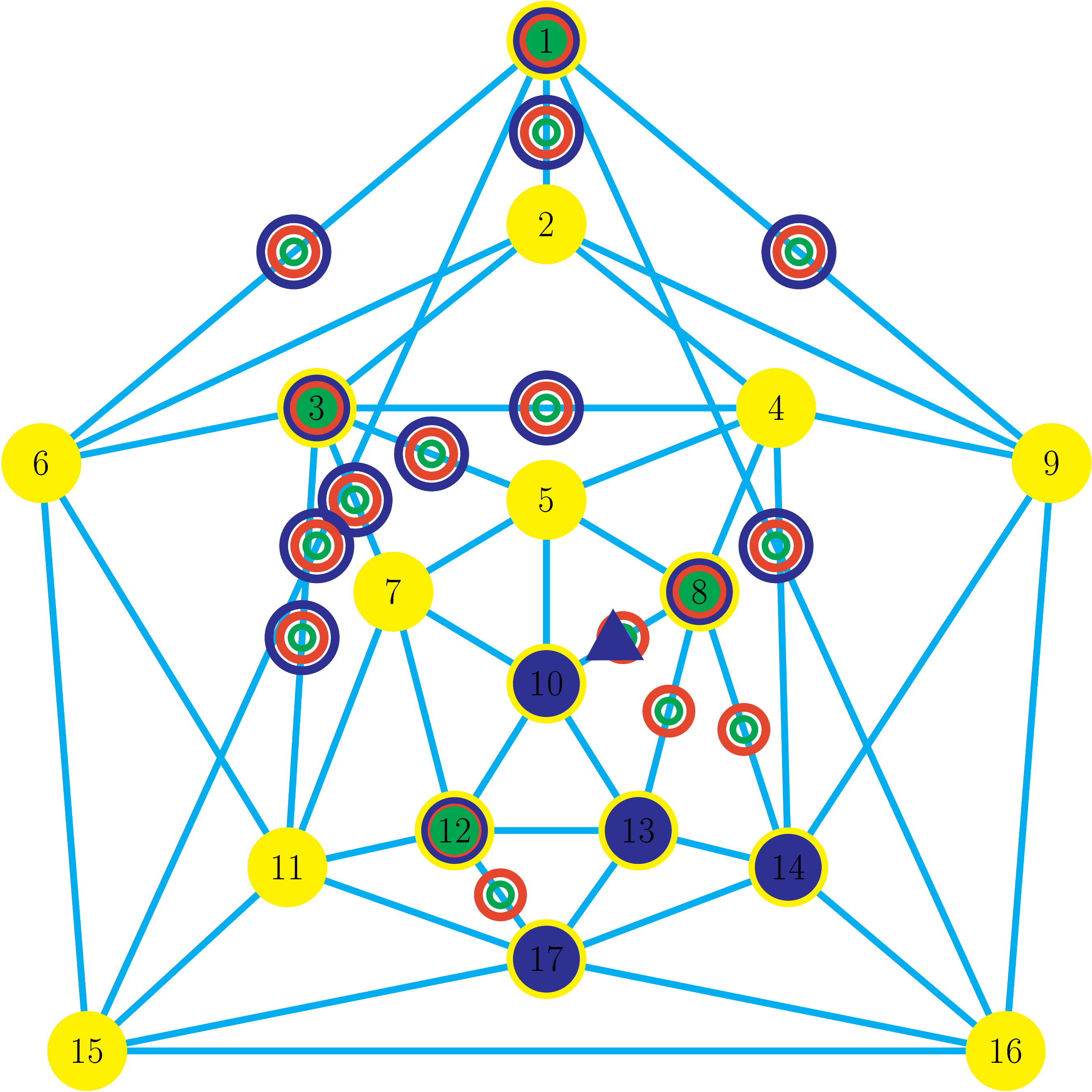


Figure 73.

instruction 141: place edge 8->10 Blue DeletionArrow

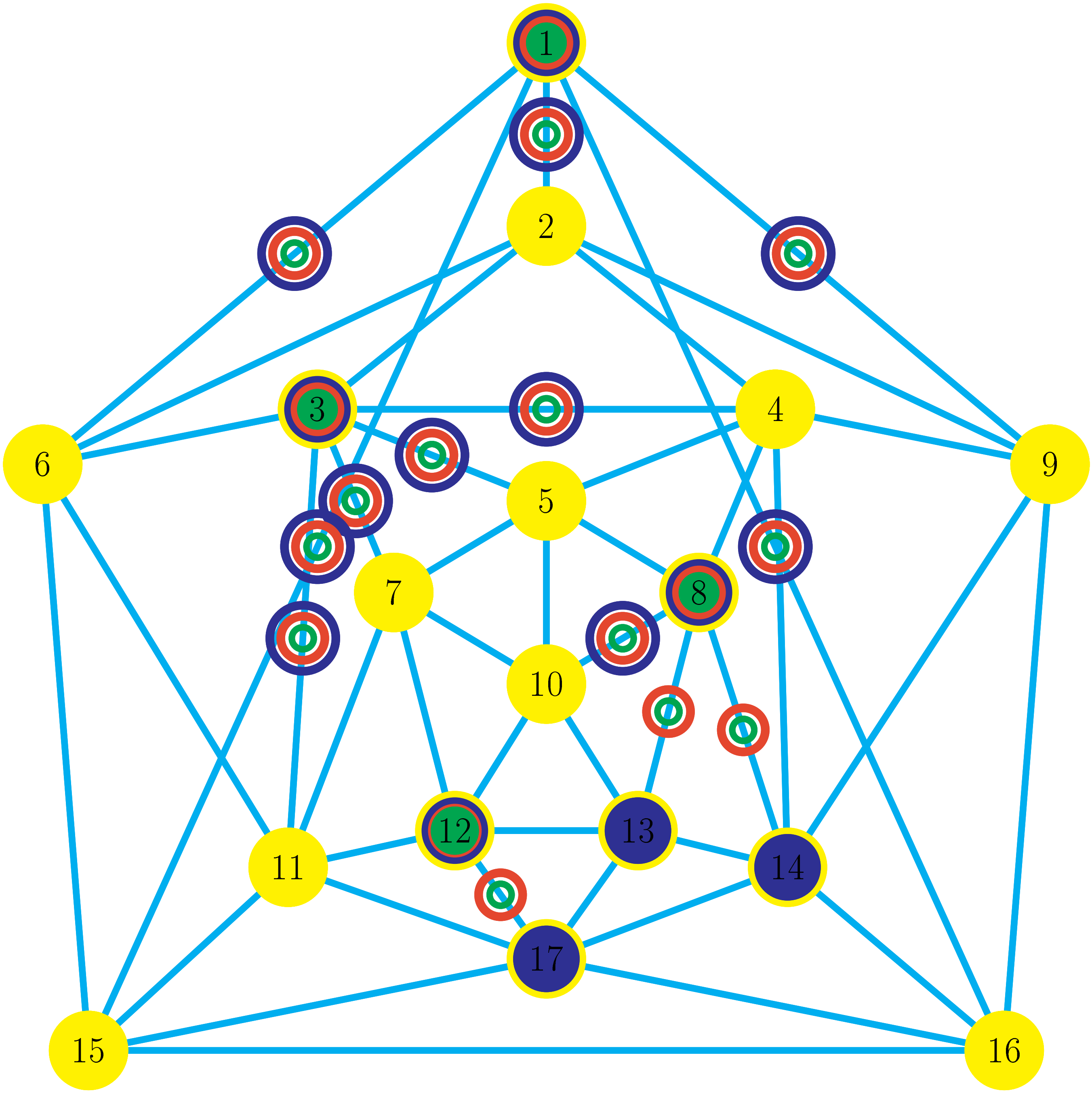


Figure 74.

instruction 142: unplace edge 8->10 Blue DeletionArrow
instruction 143: place edge 8-10 Blue Checker
instruction 144: unplace vertex 10 Blue Checker;

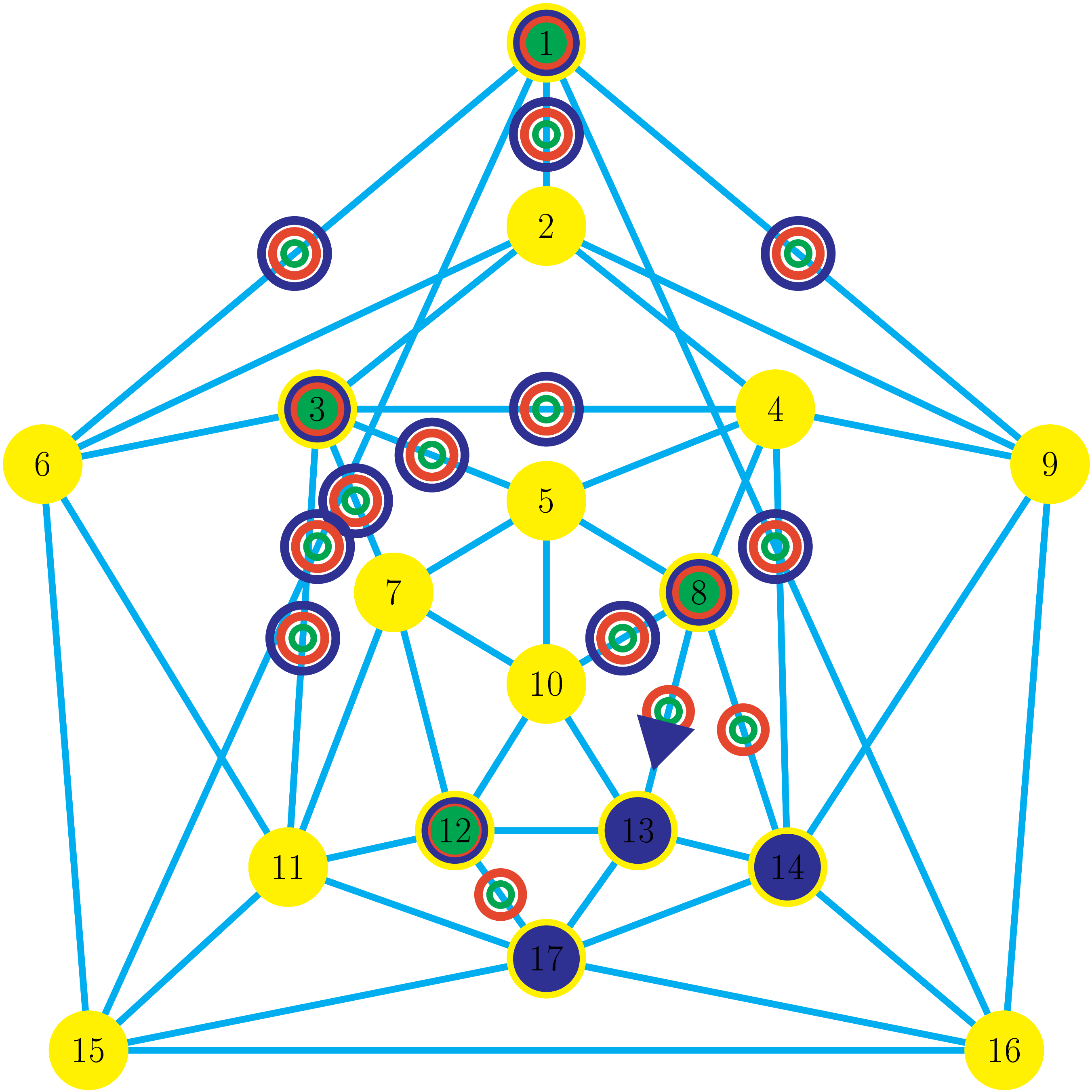


Figure 75.

instruction 145: place edge 8->13 Blue DeletionArrow

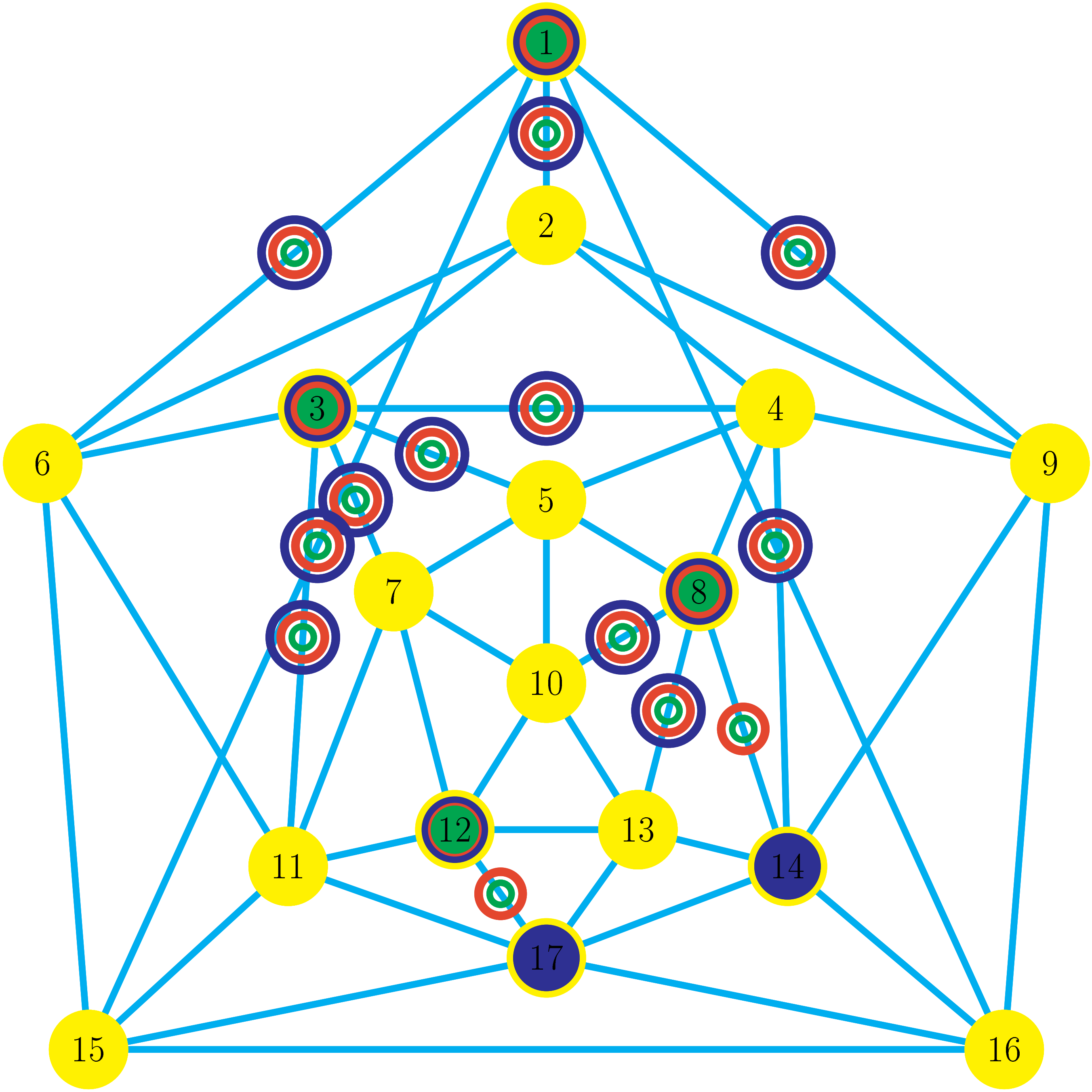


FIGURE 76.

instruction 146: unplace edge 8->13 Blue DeletionArrow
instruction 147: place edge 8-13 Blue Checker
instruction 148: unplace vertex 13 Blue Checker;

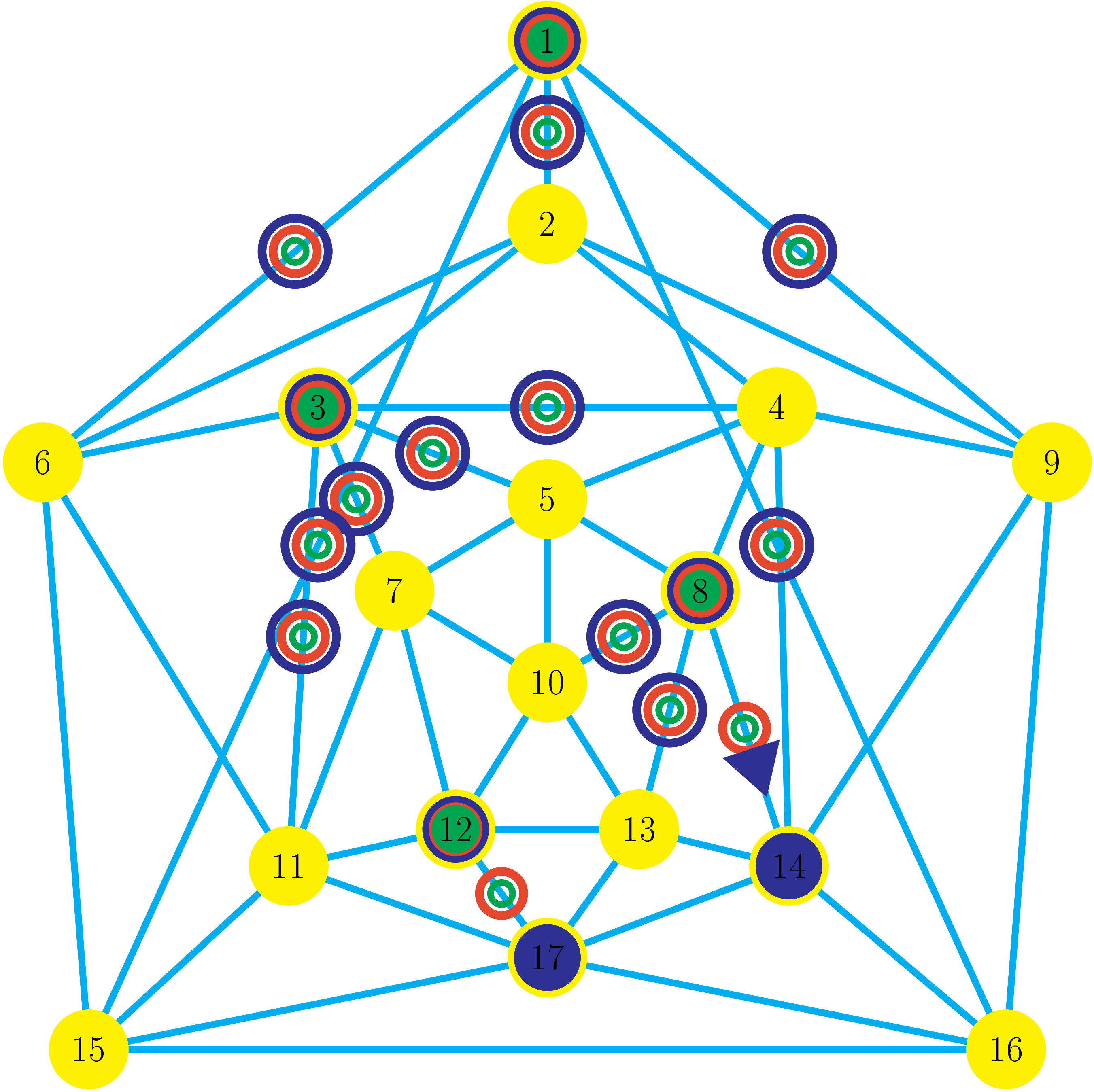


Figure 77.

instruction 149: place edge 8->14 Blue DeletionArrow

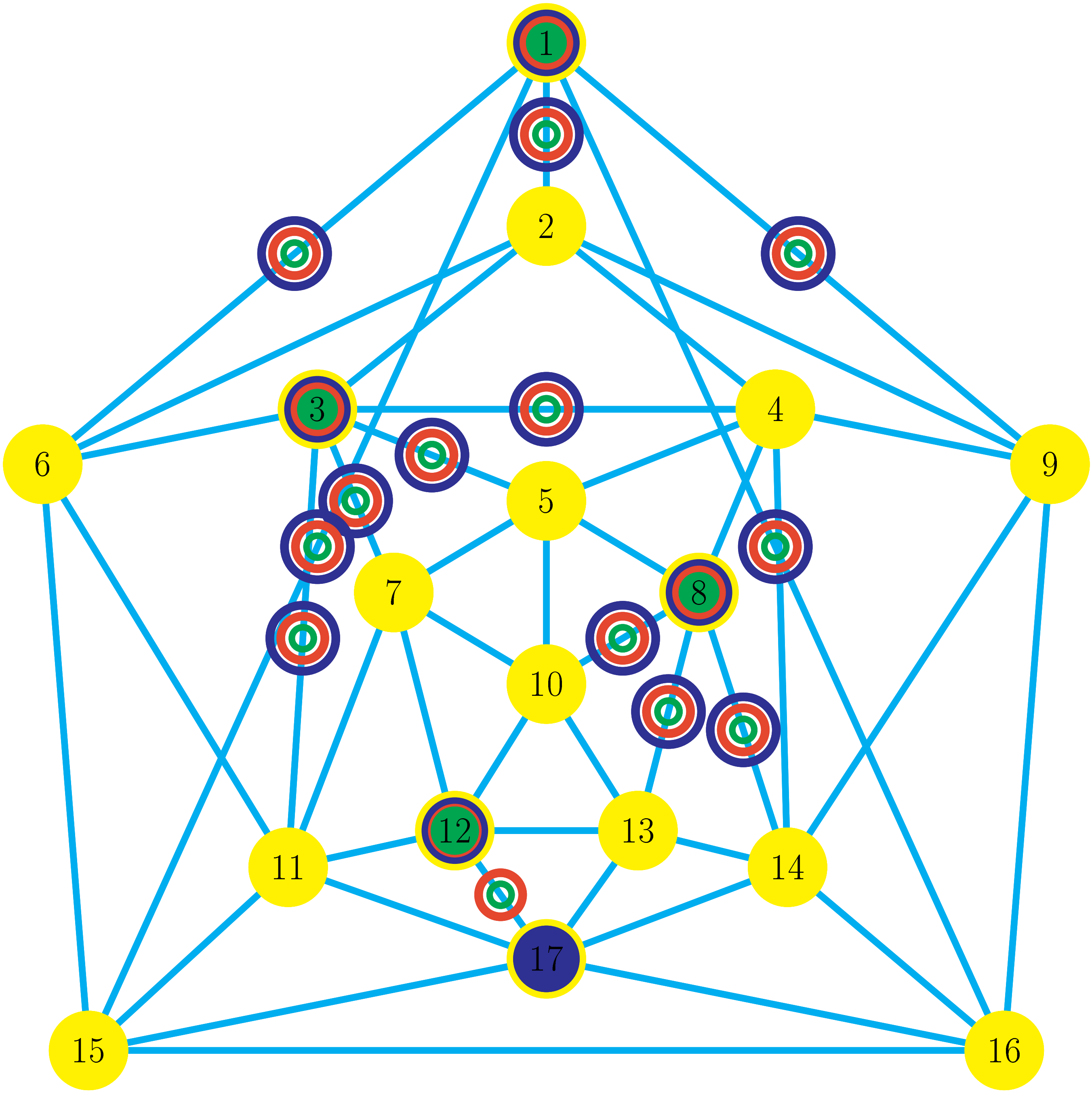


FIGURE 78.

instruction 150: unplace edge 8->14 Blue DeletionArrow
instruction 151: place edge 8-14 Blue Checker
instruction 152: unplace vertex 14 Blue Checker;

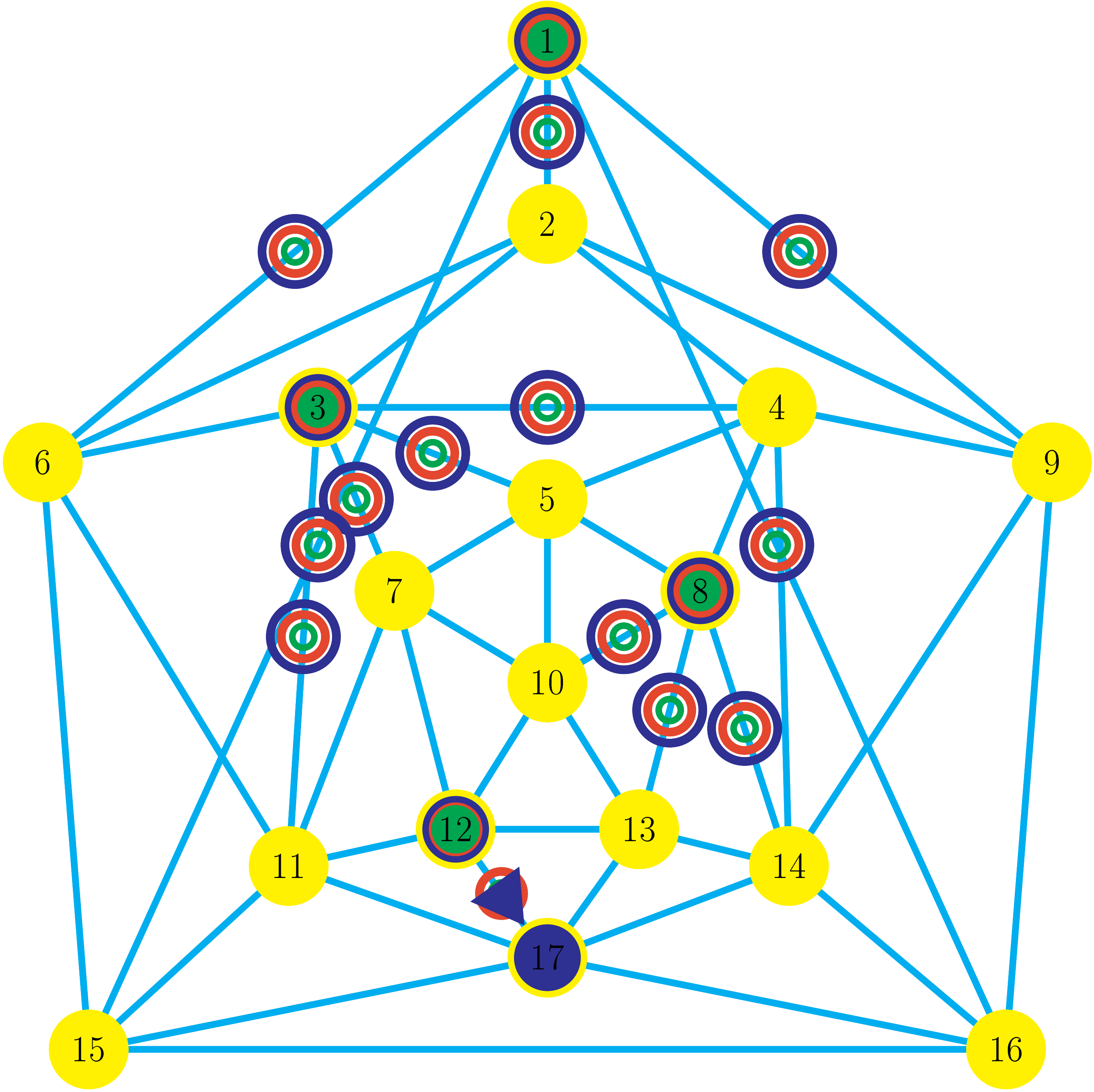


Figure 79.

instruction 153: place edge 12->17 Blue DeletionArrow

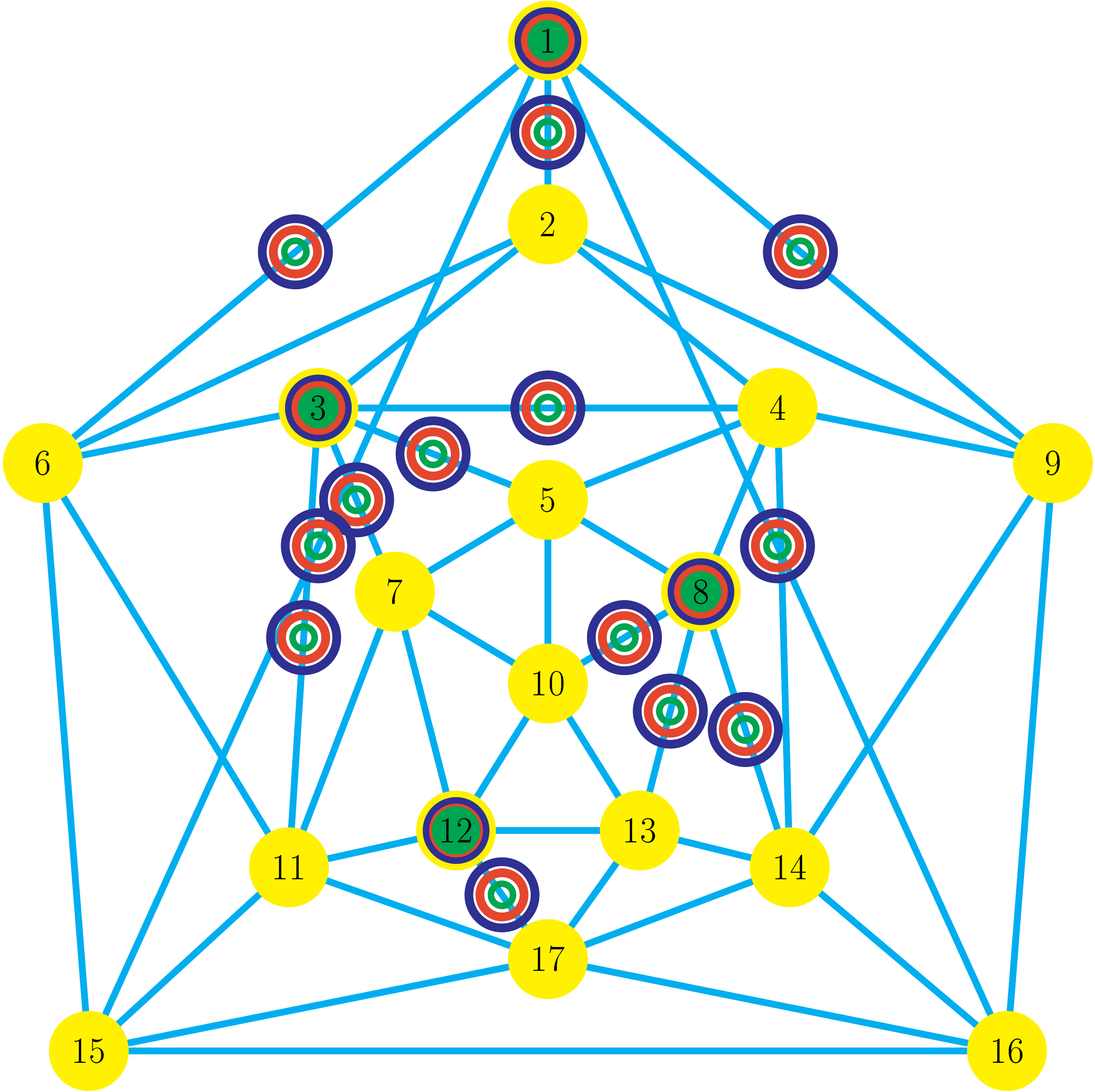


Figure 80.

instruction 154: unplace edge 12->17 Blue DeletionArrow
instruction 155: place edge 12-17 Blue Checker
instruction 156: unplace vertex 17 Blue Checker;

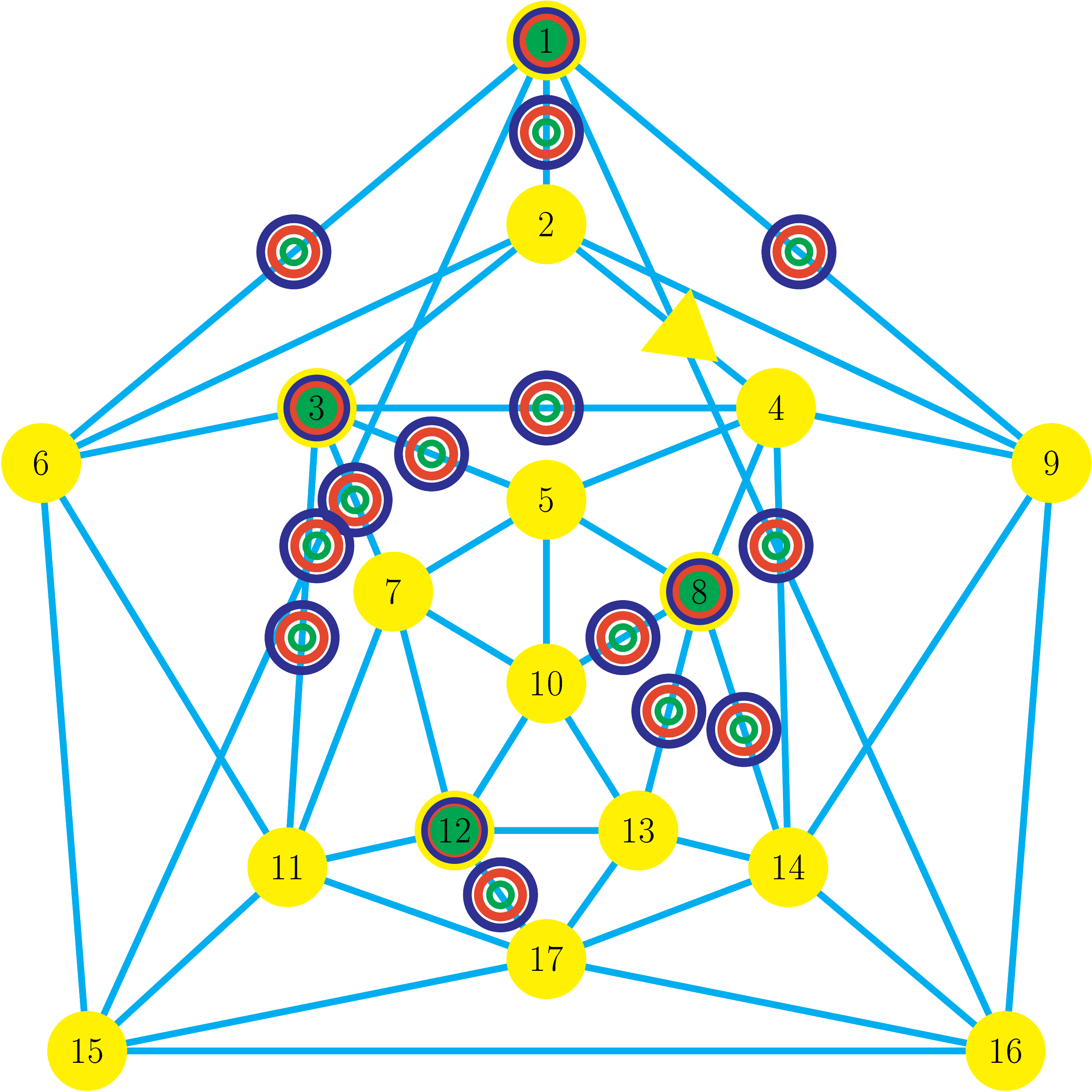


Figure 81.

instruction 157: place edge 2->4 Yellow DeletionArrow

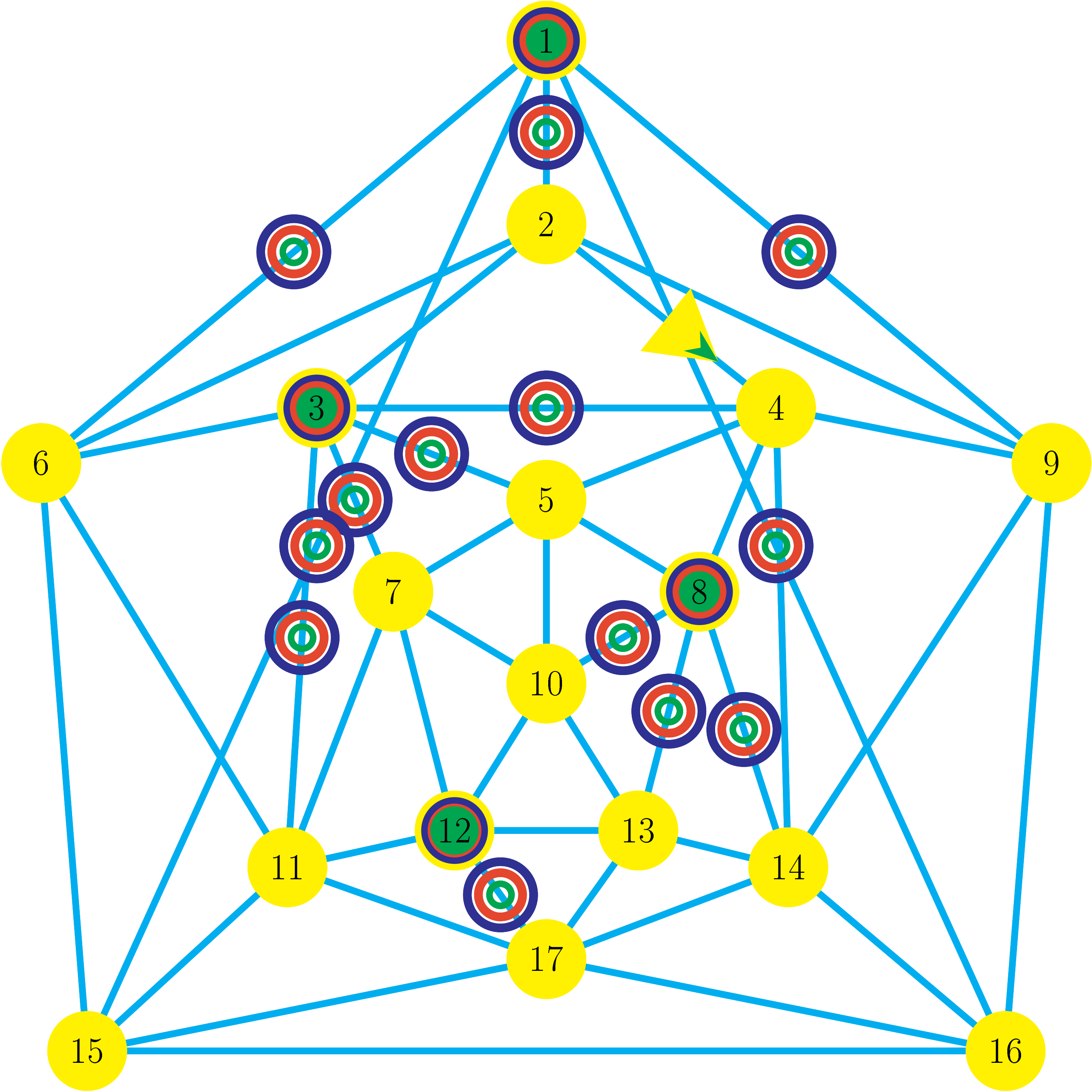


Figure 82.

instruction 158: place edge 2->4 Green InsertionArrow

Here we encounter the first insertion arrow, namely the Green arrow on edge 2→4. Insertion arrows are indicated by a pointy tail, whilst Deletion arrows have a flat tail. Insertion arrows indicate the insertion (placement) of a checker on a vertex, in this case a Green Checker on vertex 4

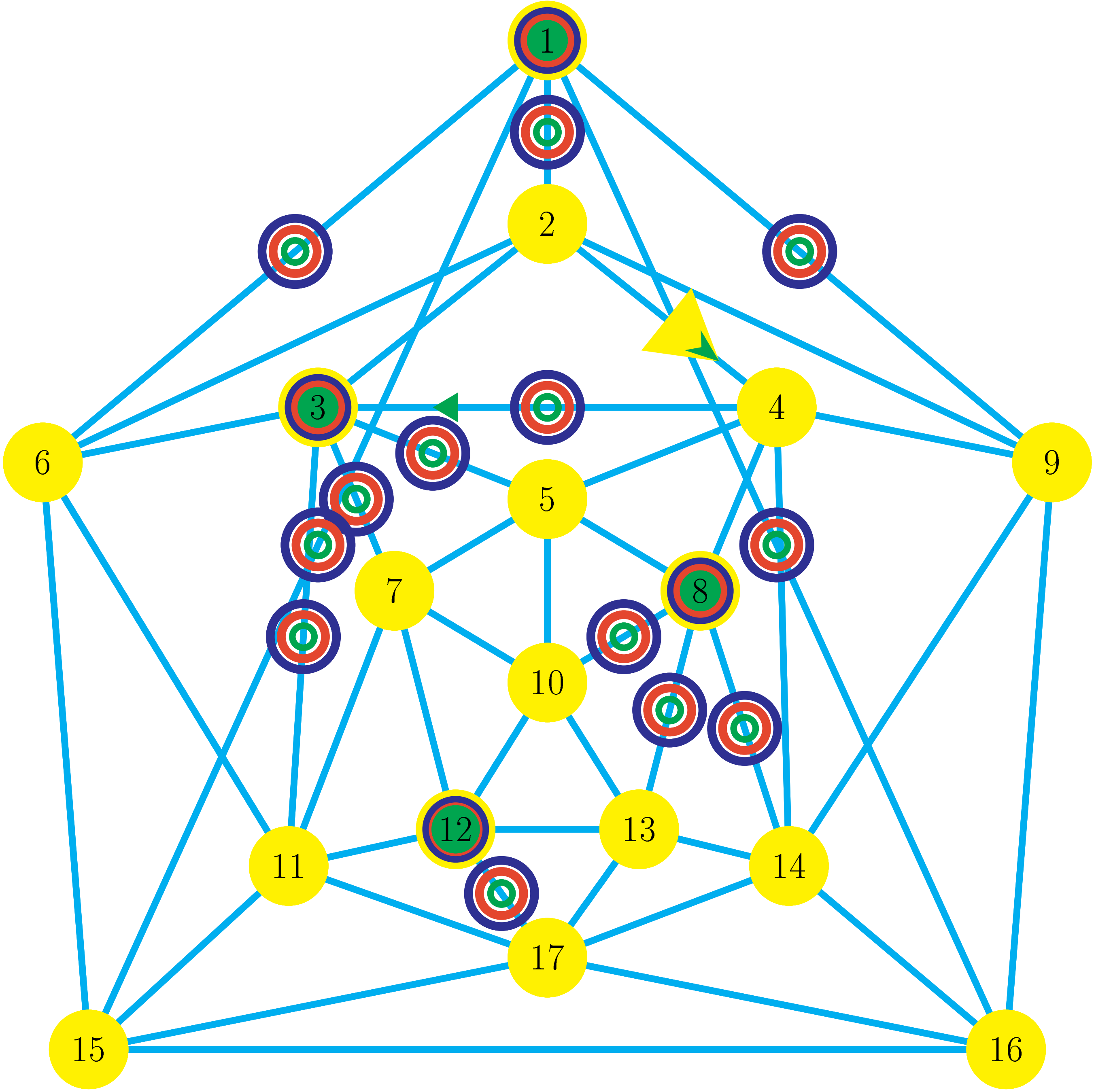


Figure 83.

instruction 159: place edge 4->3 Green DeletionArrow

Here we encounter the first alternating chain of more than one edge.

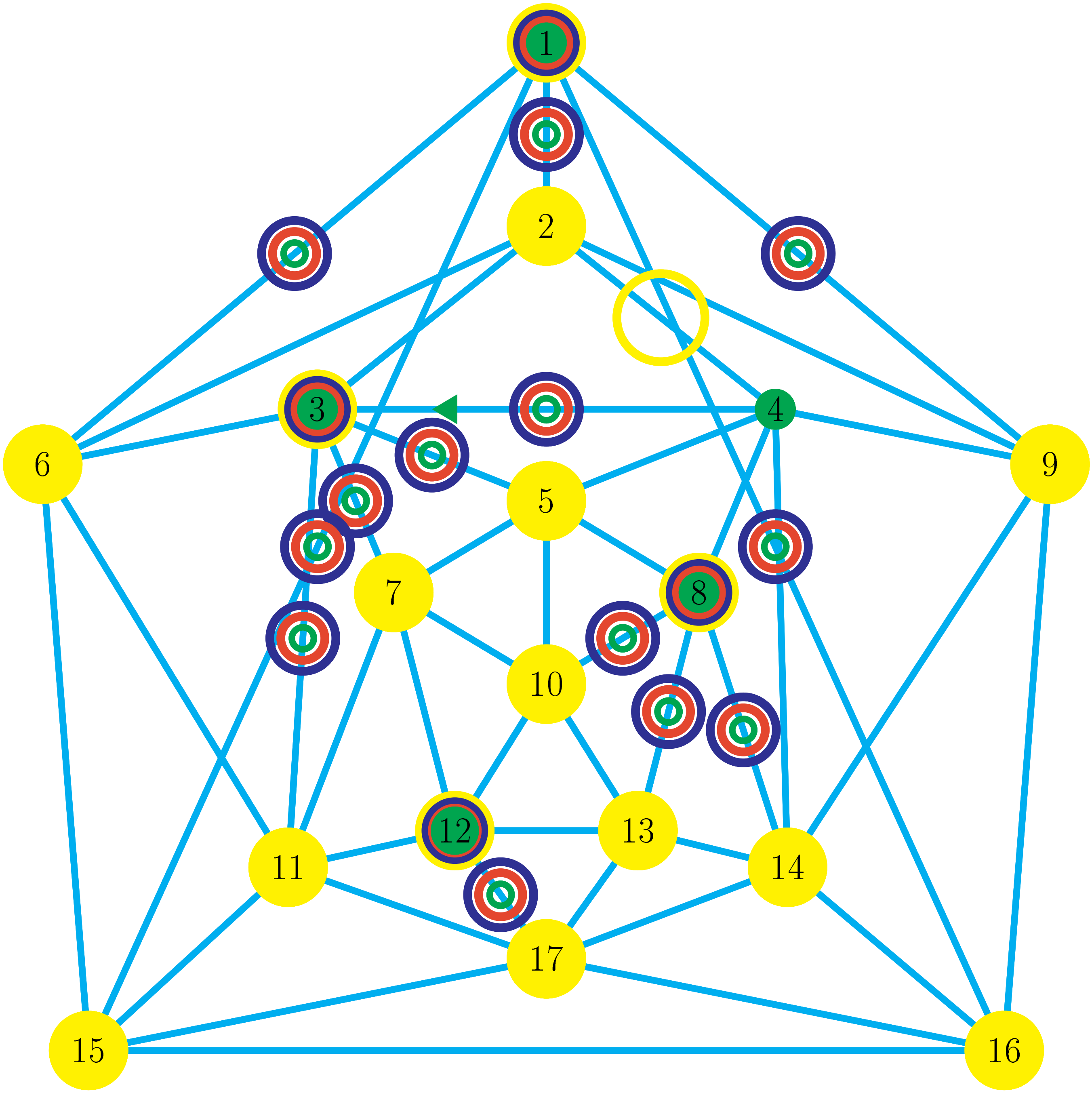


Figure 84.

instruction 160: unplace vertex 4 Yellow Checker;
instruction 161: place vertex 4 Green Checker;
instruction 162: unplace edge 2->4 Yellow DeletionArrow
instruction 163: unplace edge 2->4 Green InsertionArrow
instruction 164: place edge 2-4 Yellow Checker

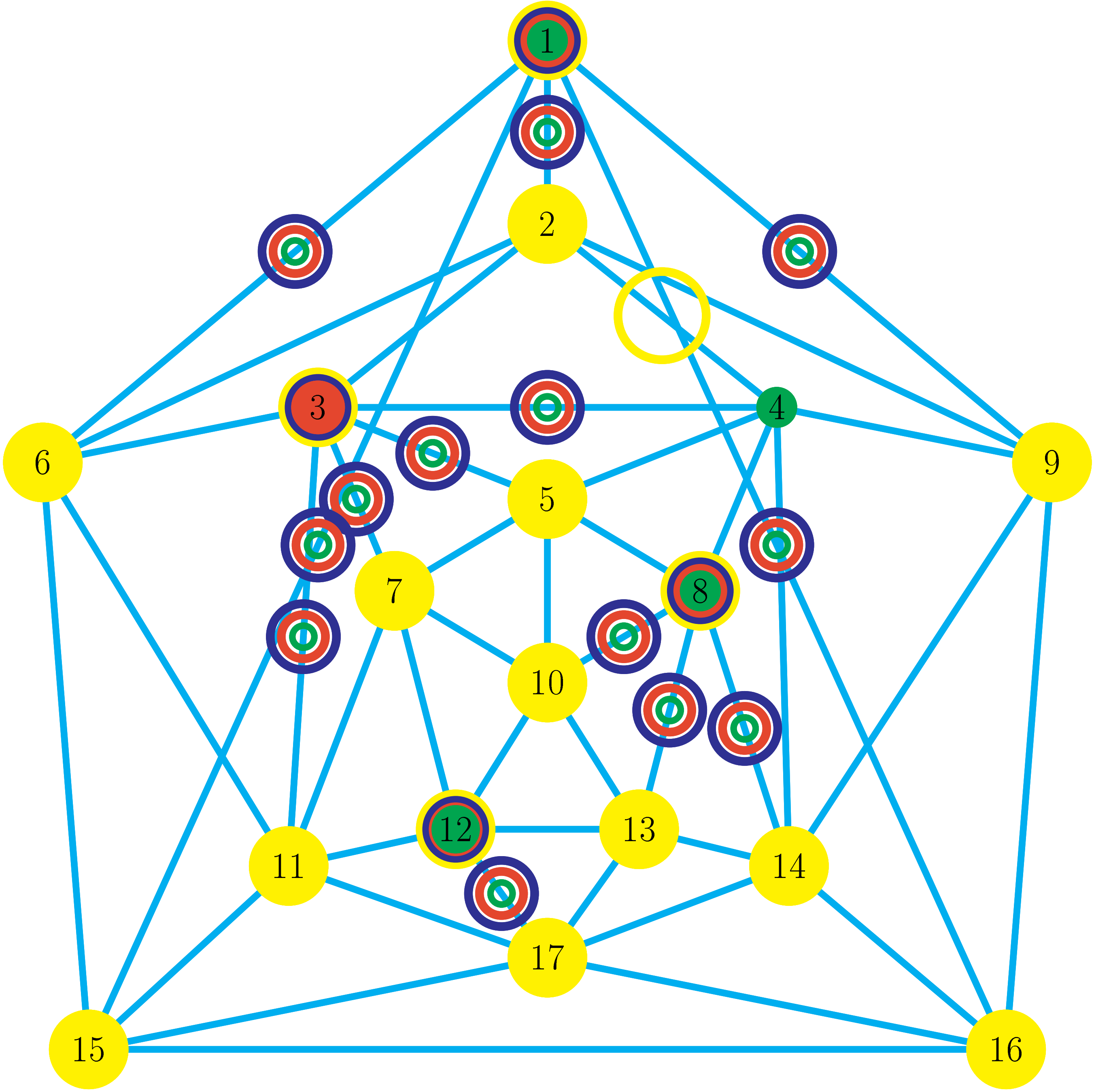


Figure 85.

instruction 165: unplace vertex 3 Green Checker;
instruction 166: unplace edge 4->3 Green DeletionArrow

Notice that propert A holds but propety B does not, It fails on edge 3-5, because the is a Green Checker on that edge but no Green Checker on iether of its vertices. Likewise for edges 3-7 and 3-11. In order to restore property B we transfer the Green Checker on edge 3-5 to edge 5-8. We do similary for edges 3-7 and 3-11.

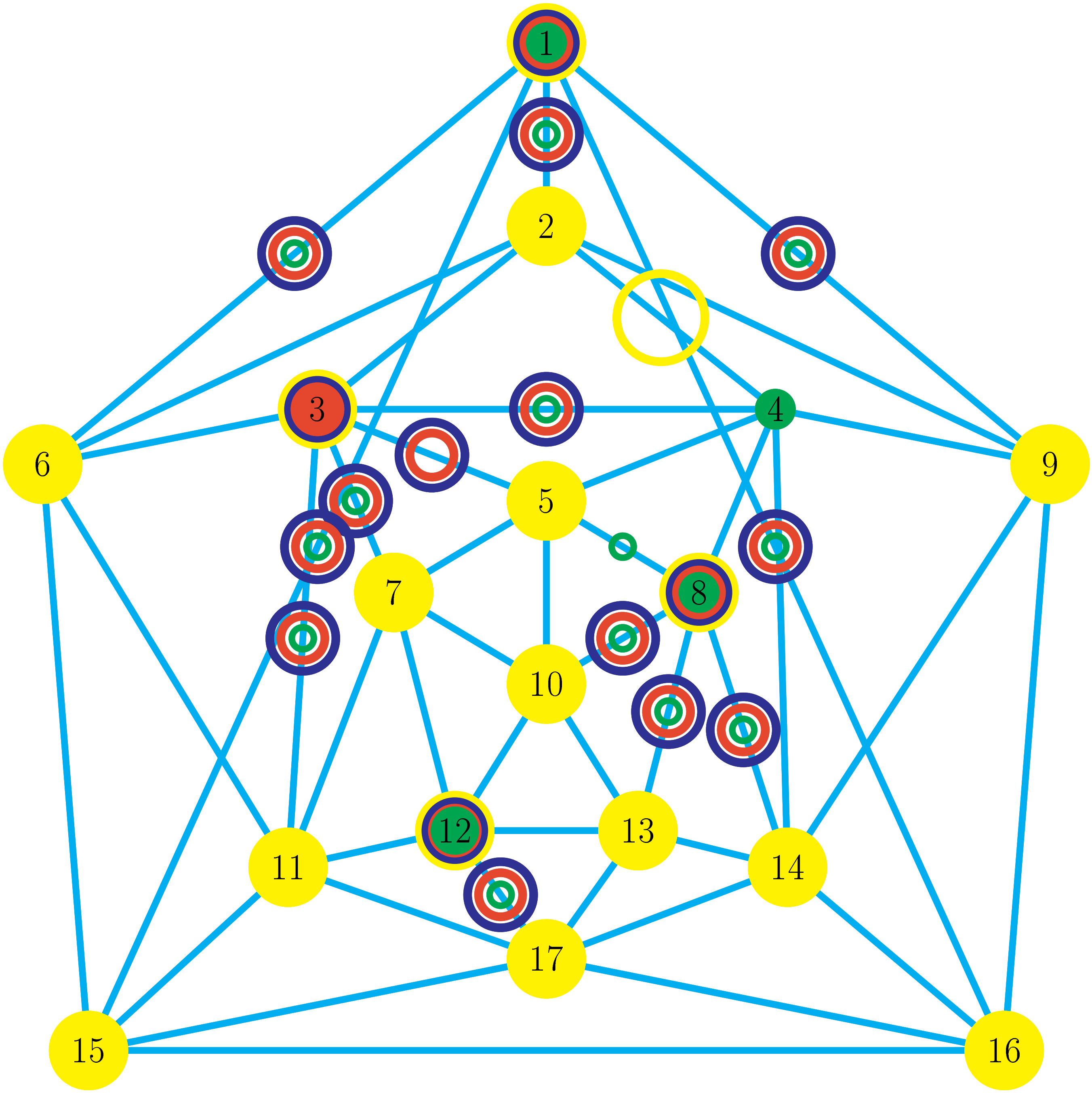


Figure 86.

instruction 167: unplace edge 3-5 Green Checker

instruction 168: place edge 5-8 Green Checker

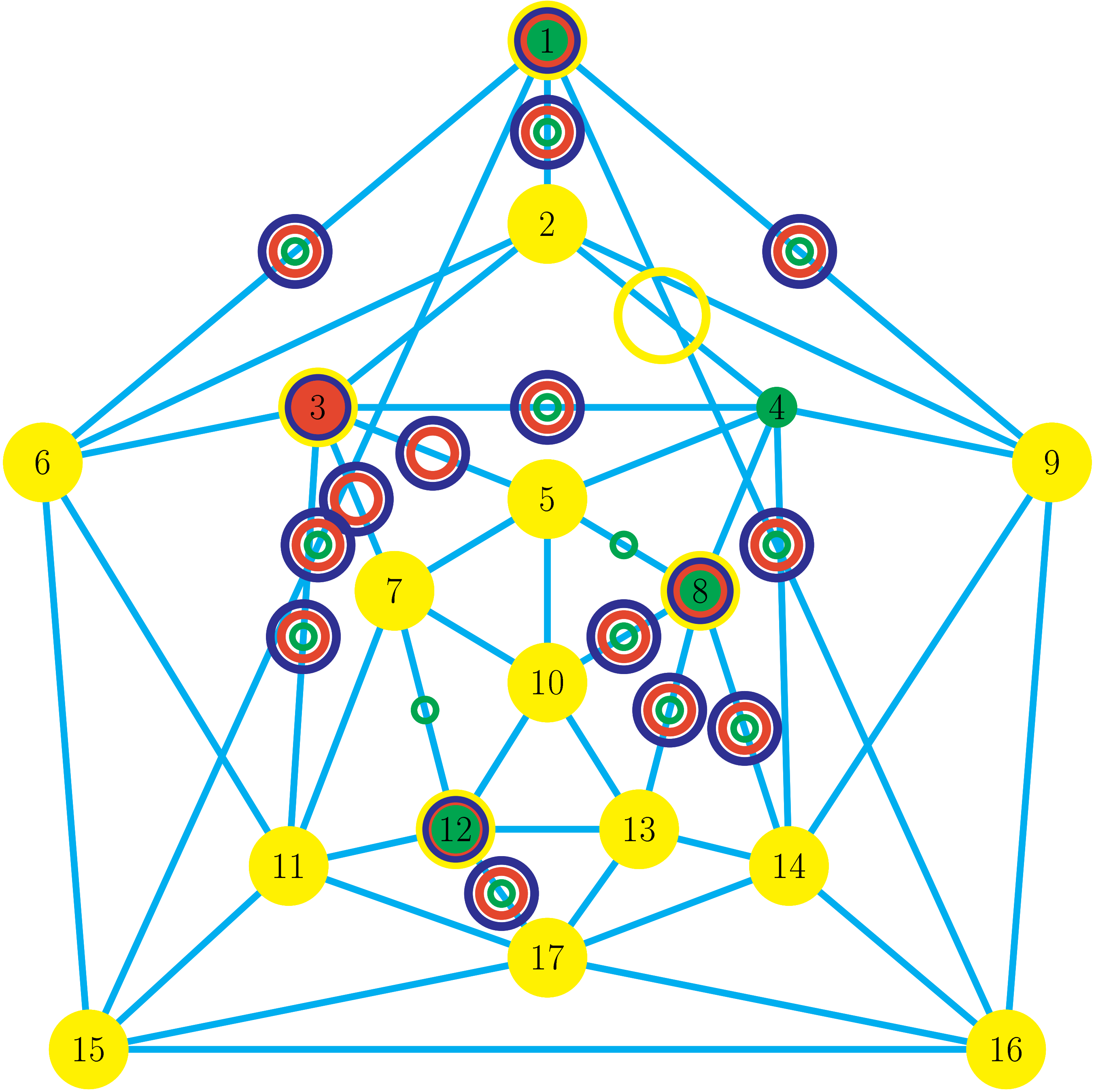


Figure 87.

instruction 169: unplace edge 3-7 Green Checker
instruction 170: place edge 7-12 Green Checker

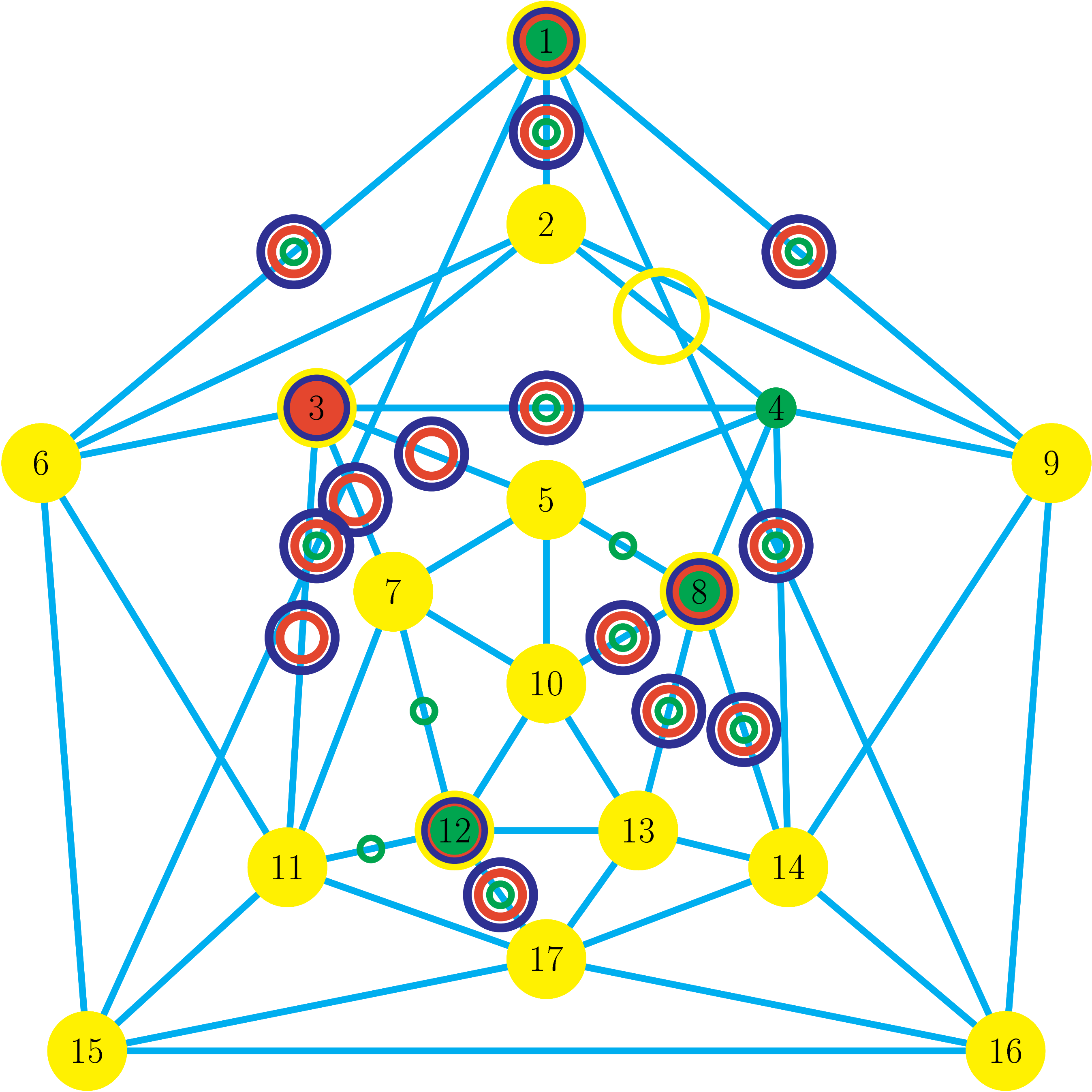


Figure 88.

instruction 171: unplace edge 3-11 Green Checker
instruction 172: place edge 11-12 Green Checker

Property B is restored and we can now find the next alternating chain.

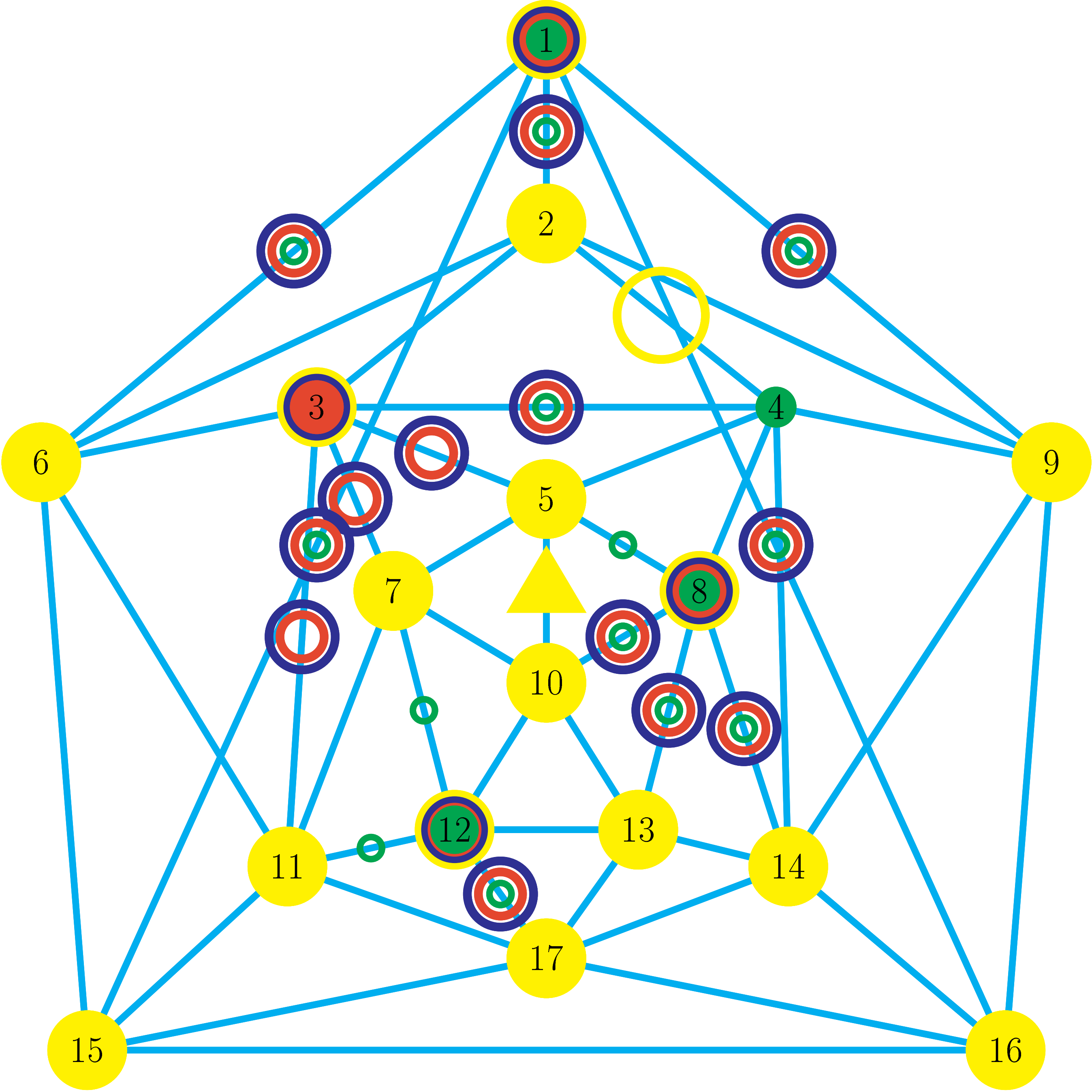


Figure 89.

instruction 173: place edge 10->5 Yellow DeletionArrow

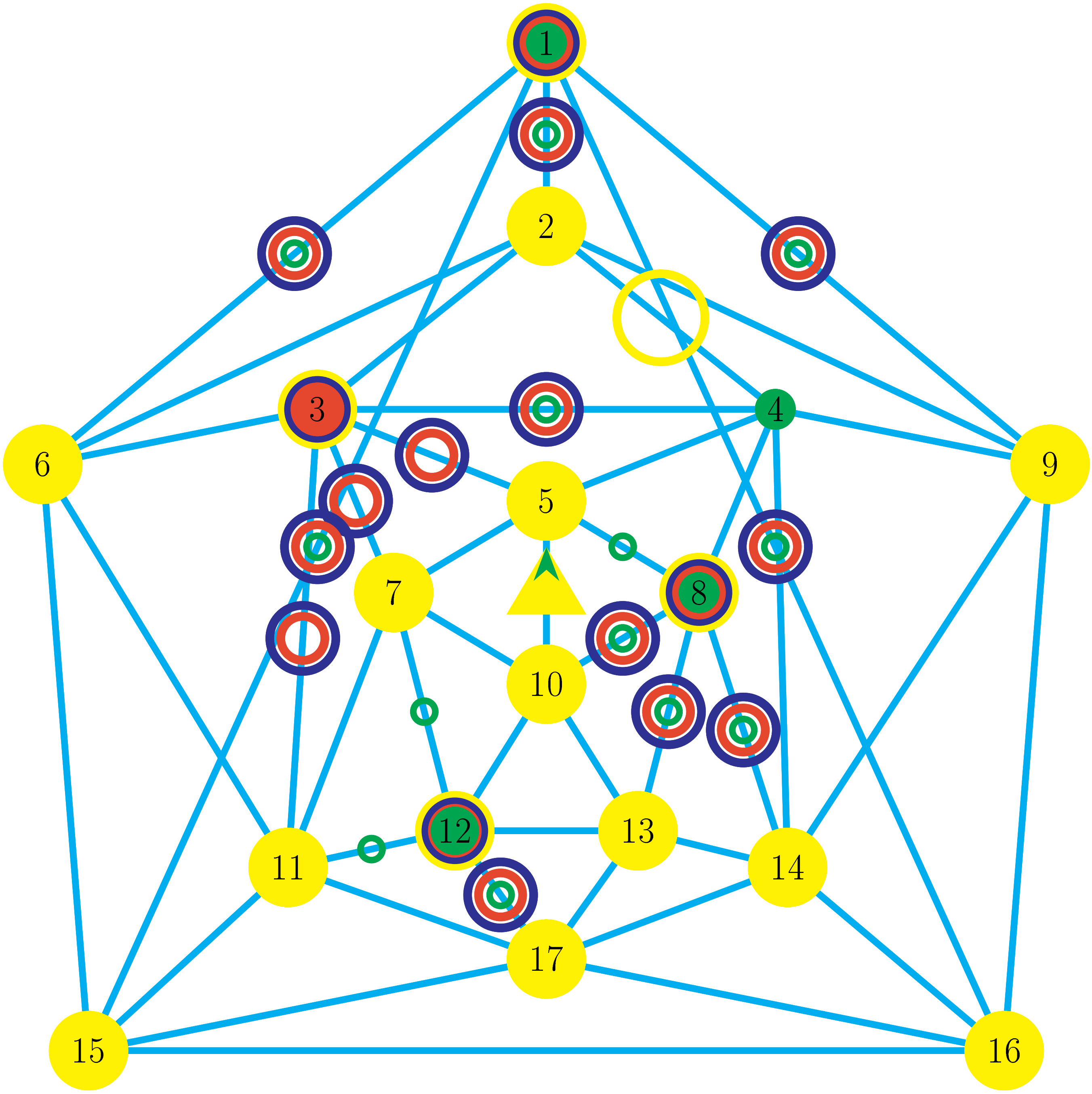


FIGURE 90.

instruction 174: place edge 10->5 Green InsertionArrow

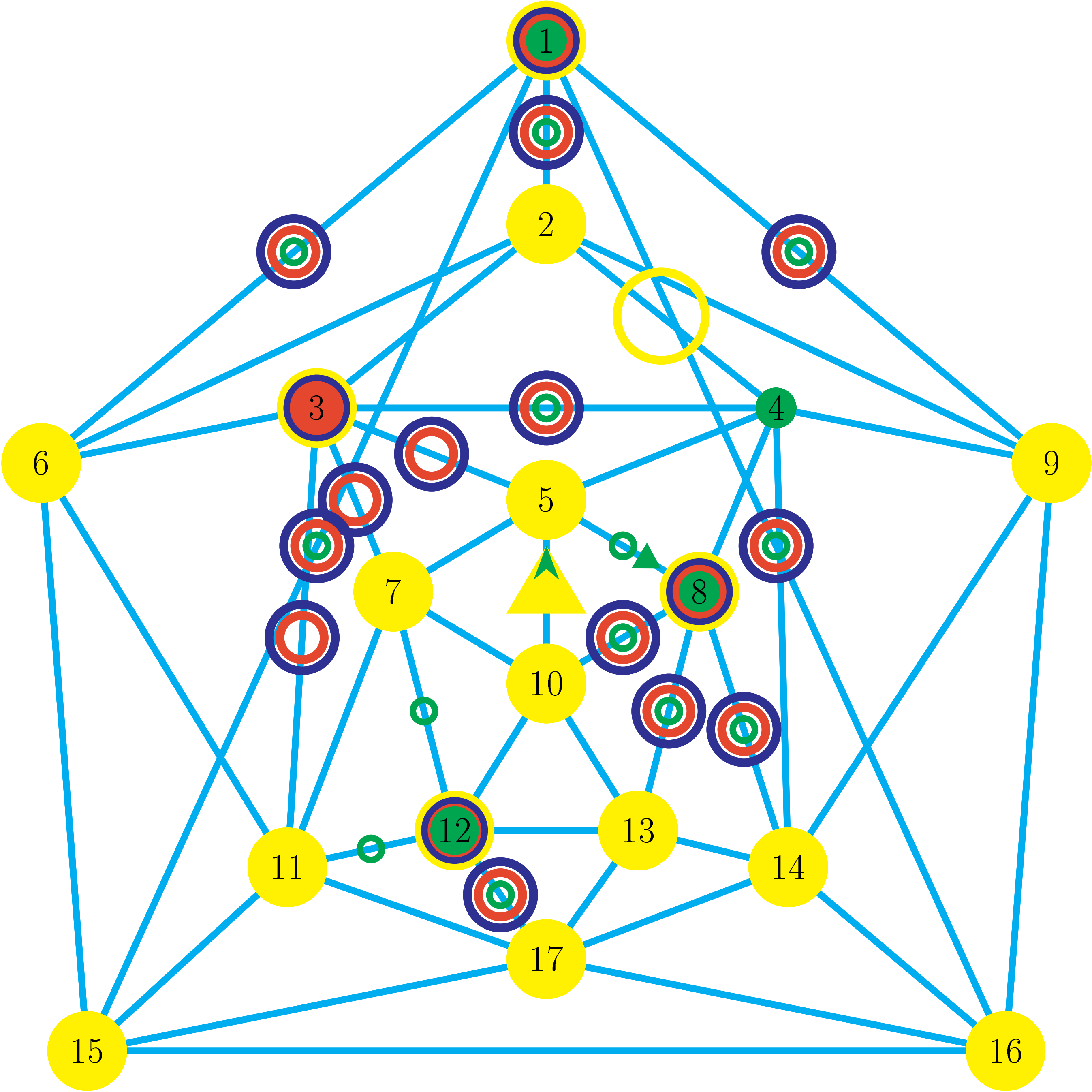


Figure 91.

instruction 175: place edge 5->8 Green DeletionArrow

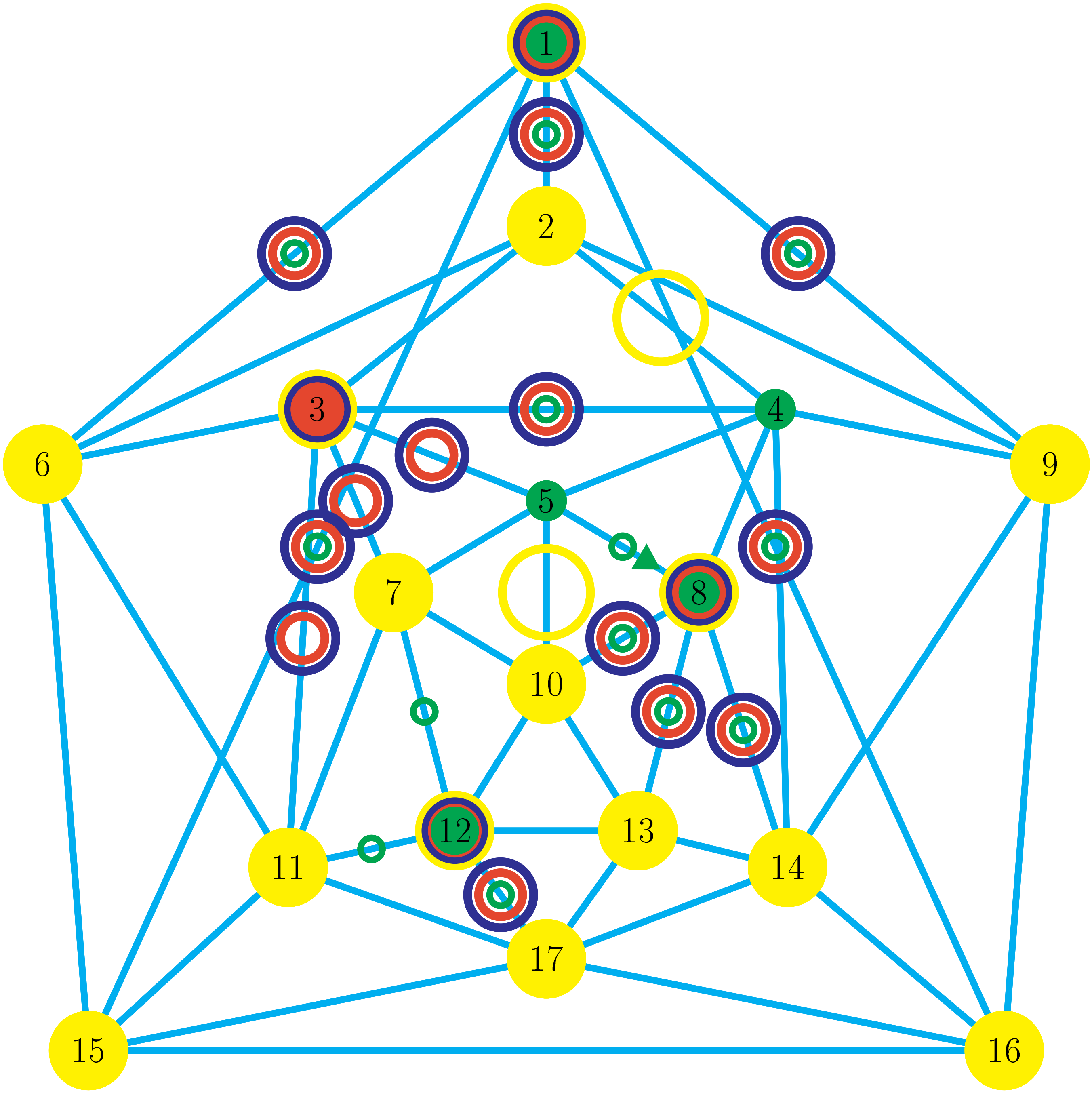


Figure 92.

instruction 176: unplace edge 10->5 Yellow DeletionArrow
instruction 177: unplace edge 10->5 Green InsertionArrow
instruction 178: place edge 10-5 Yellow Checker
instruction 179: unplace vertex 5 Yellow Checker;
instruction 180: place vertex 5 Green Checker;

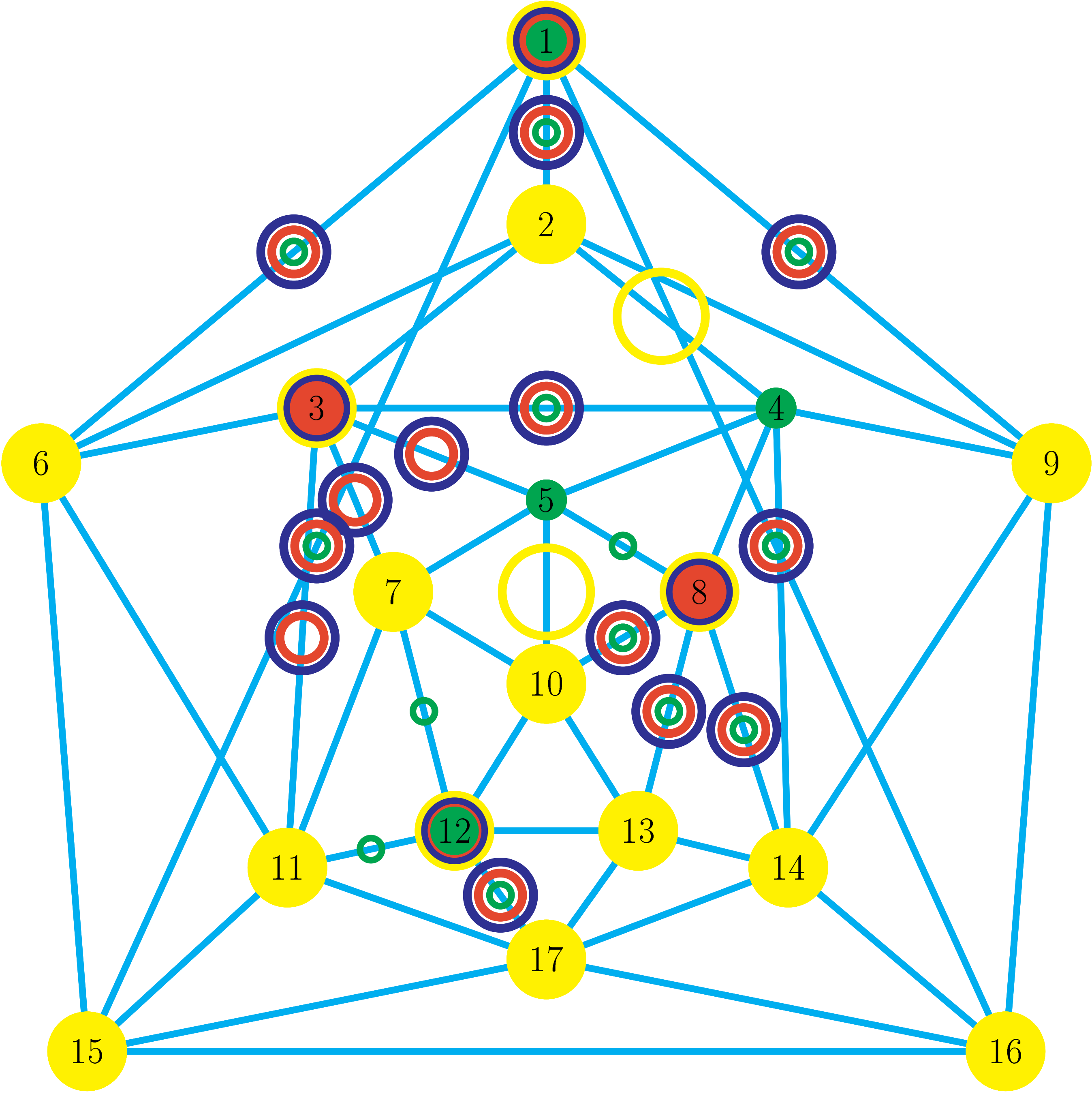


Figure 93.

instruction 181: unplace edge 5->8 Green DeletionArrow
instruction 182: unplace vertex 8 Green Checker;

Notice that propert A holds but propety B does not, It fails on edges 8-14, 8-13 and 8-10. It is restored as follows.

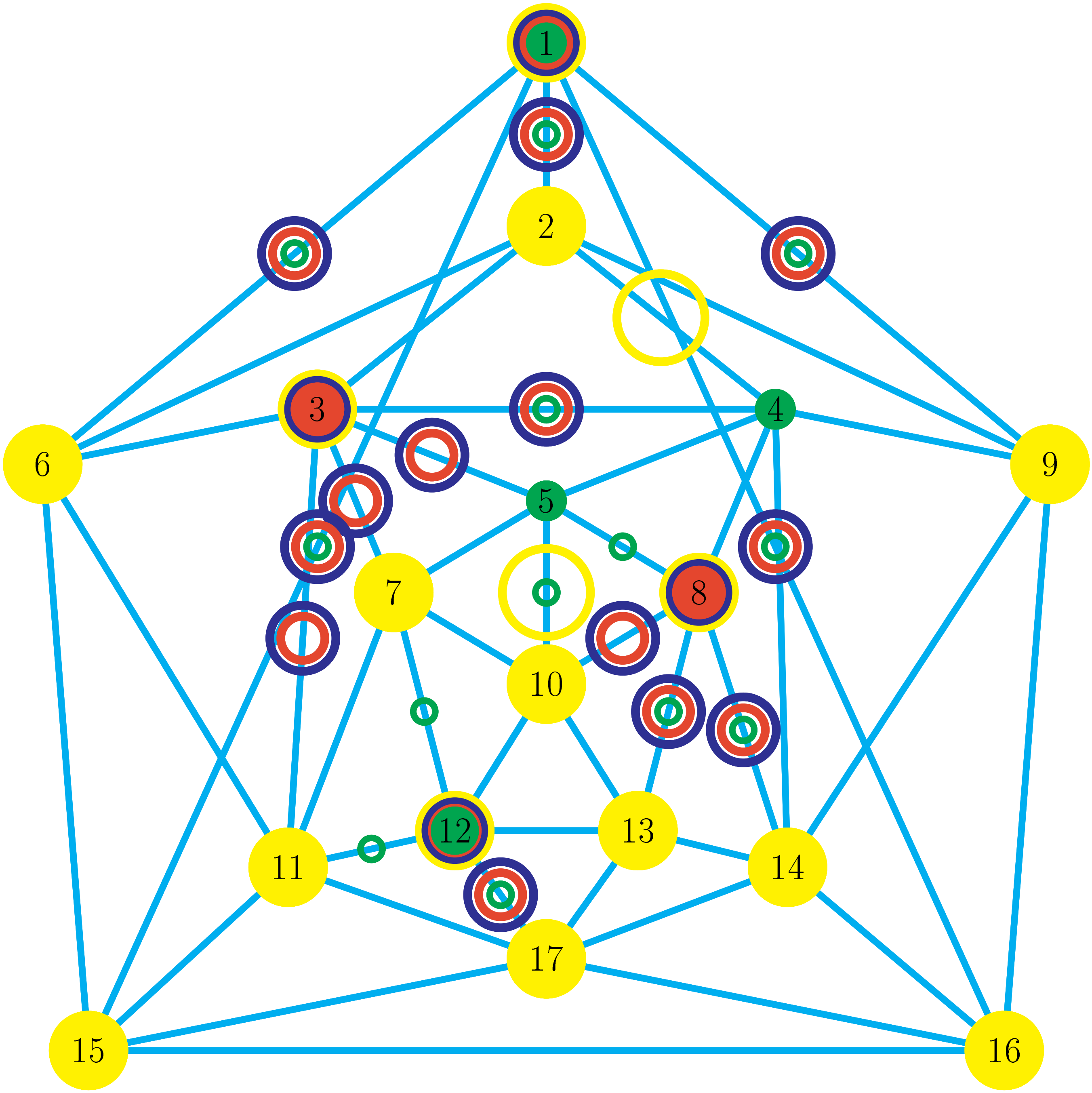


Figure 94.

instruction 183: unplace edge 10-8 Green Checker
instruction 184: place edge 10-5 Green Checker

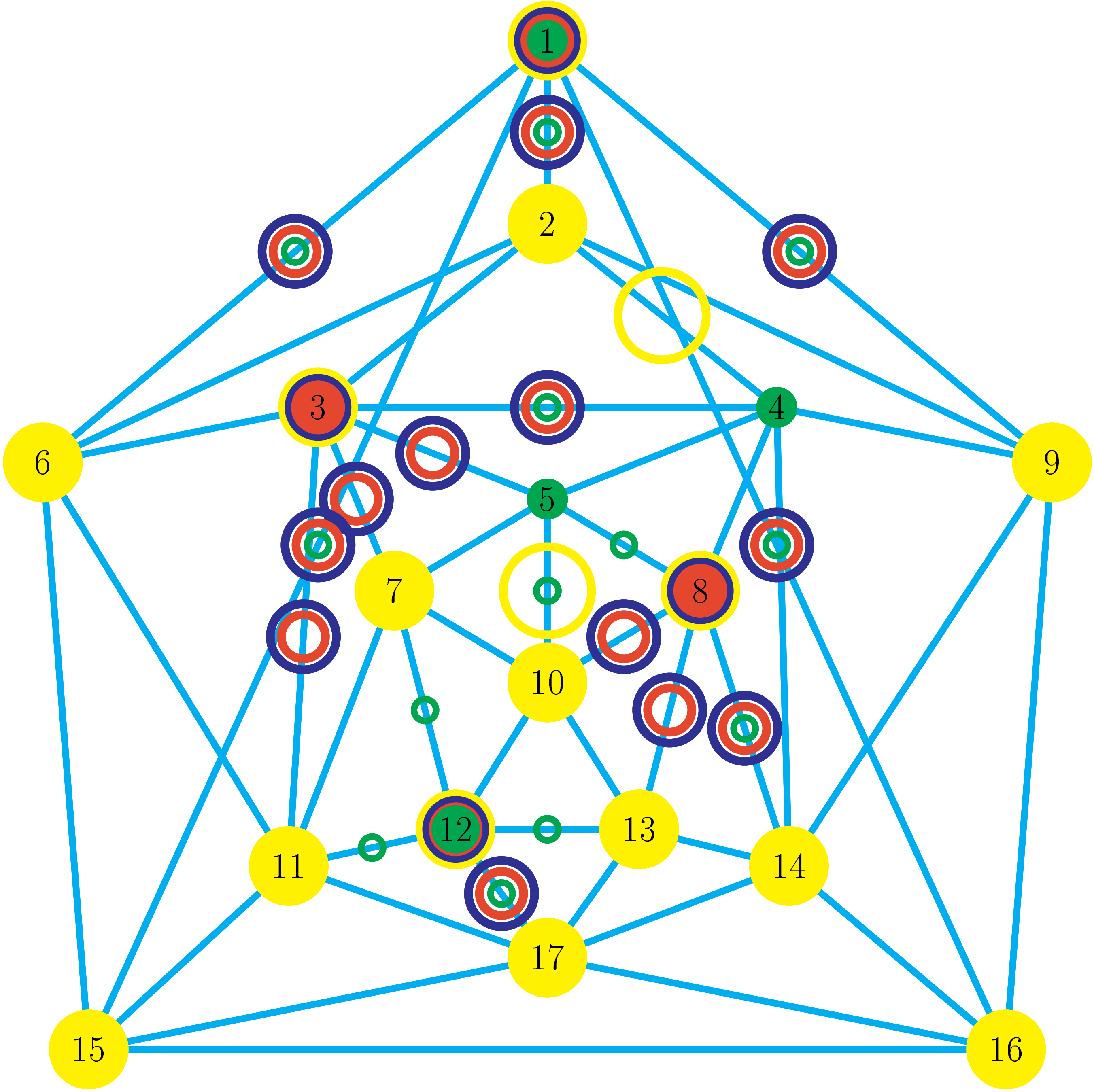


Figure 95.

instruction 185: unplace edge 8-13 Green Checker
instruction 186: place edge 12-13 Green Checker

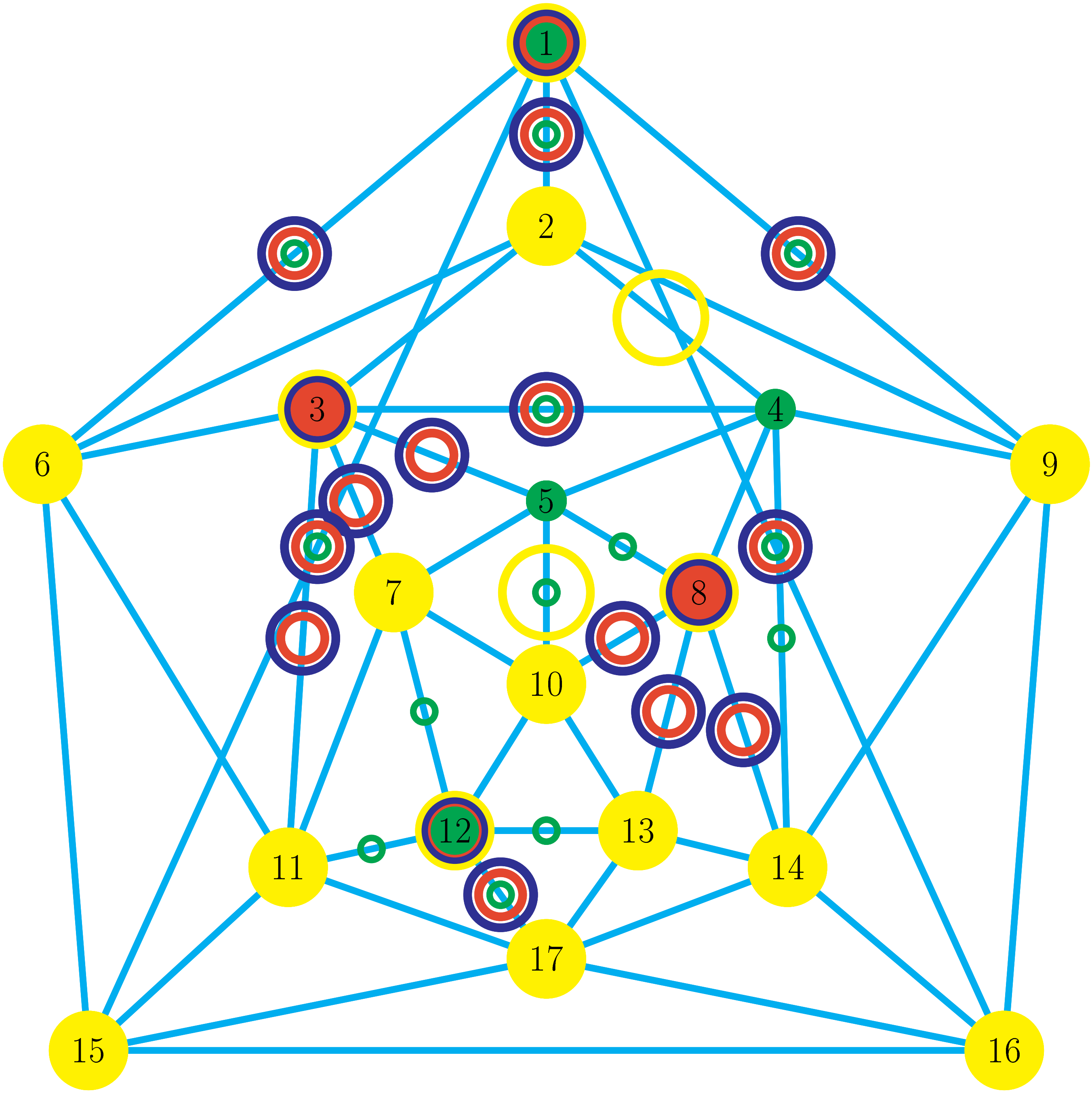


Figure 96.

instruction 187: unplace edge 8-14 Green Checker
instruction 188: place edge 4-14 Green Checker

Property B is restored and we can now proceed to find the next alternating chain.

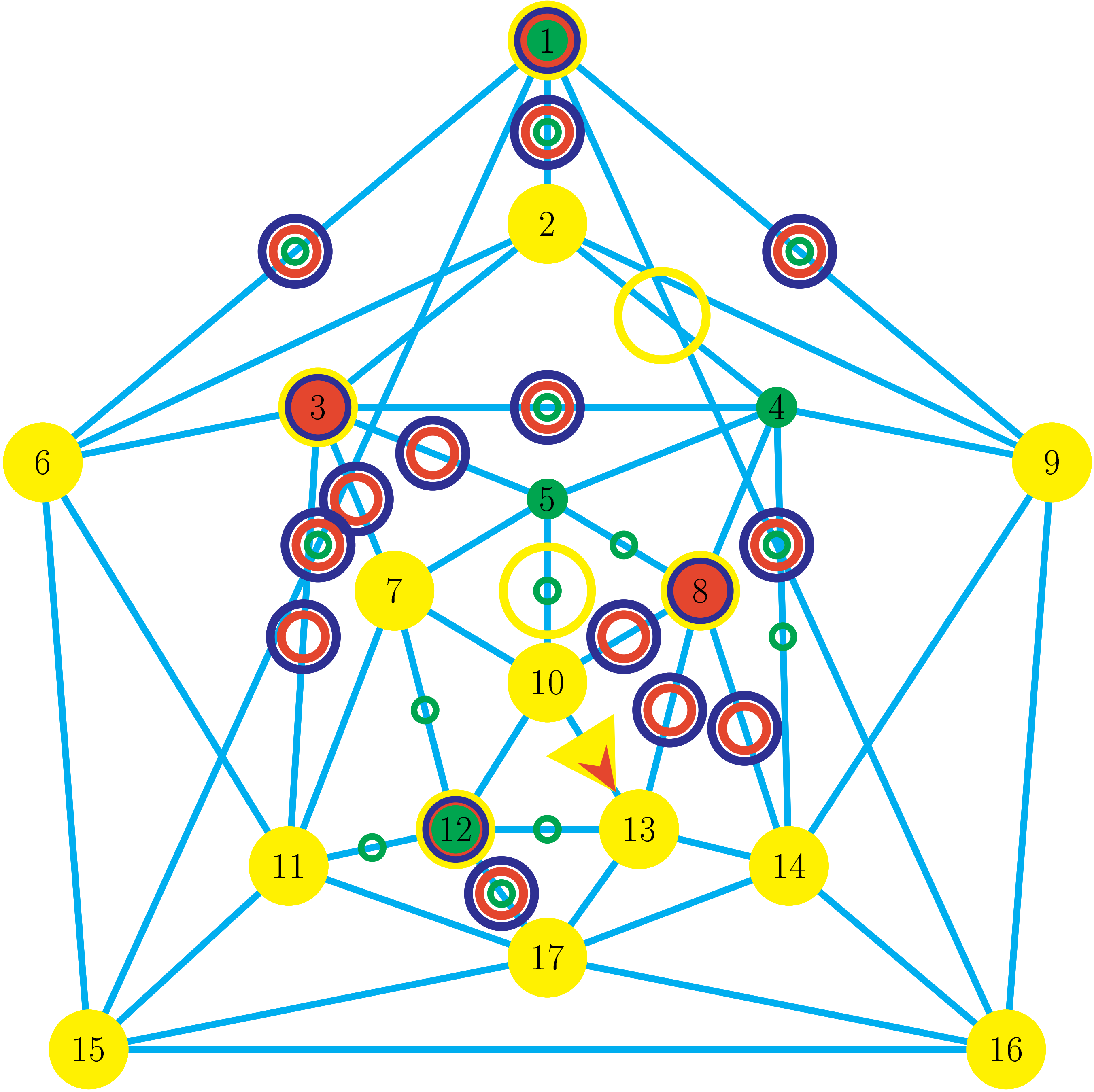


Figure 97.

instruction 189: place edge 10->13 Yellow DeletionArrow
instruction 190: place edge 10->13 Red InsertionArrow

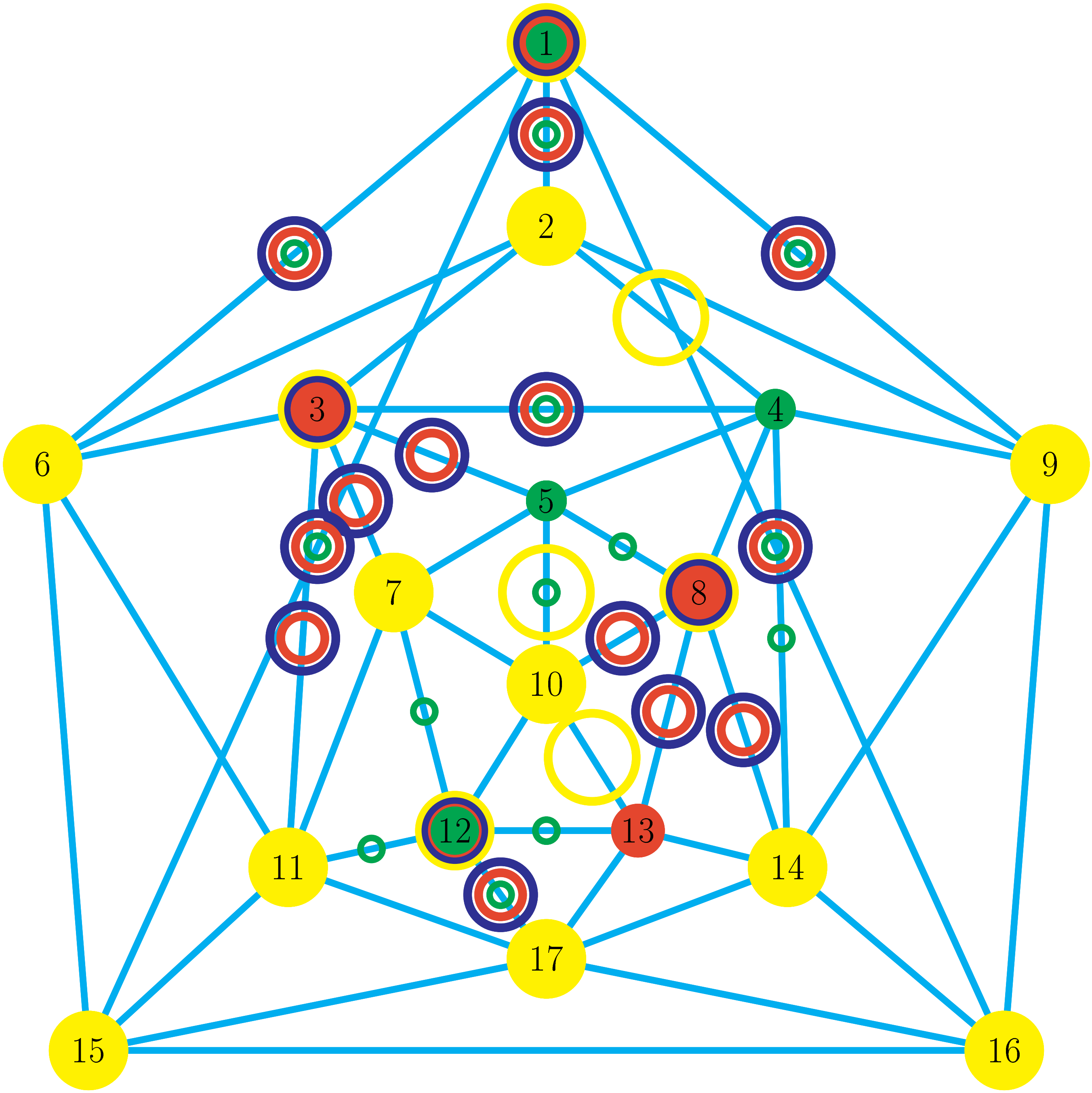


Figure 98.

instruction 191: place edge 10-13 Yellow Checker
instruction 192: place vertex 13 Red Checker;
instruction 193: unplace vertex 13 Yellow Checker;
instruction 194: unplace edge 10->13 Yellow DeletionArrow
instruction 195: unplace edge 10->13 Red InsertionArrow

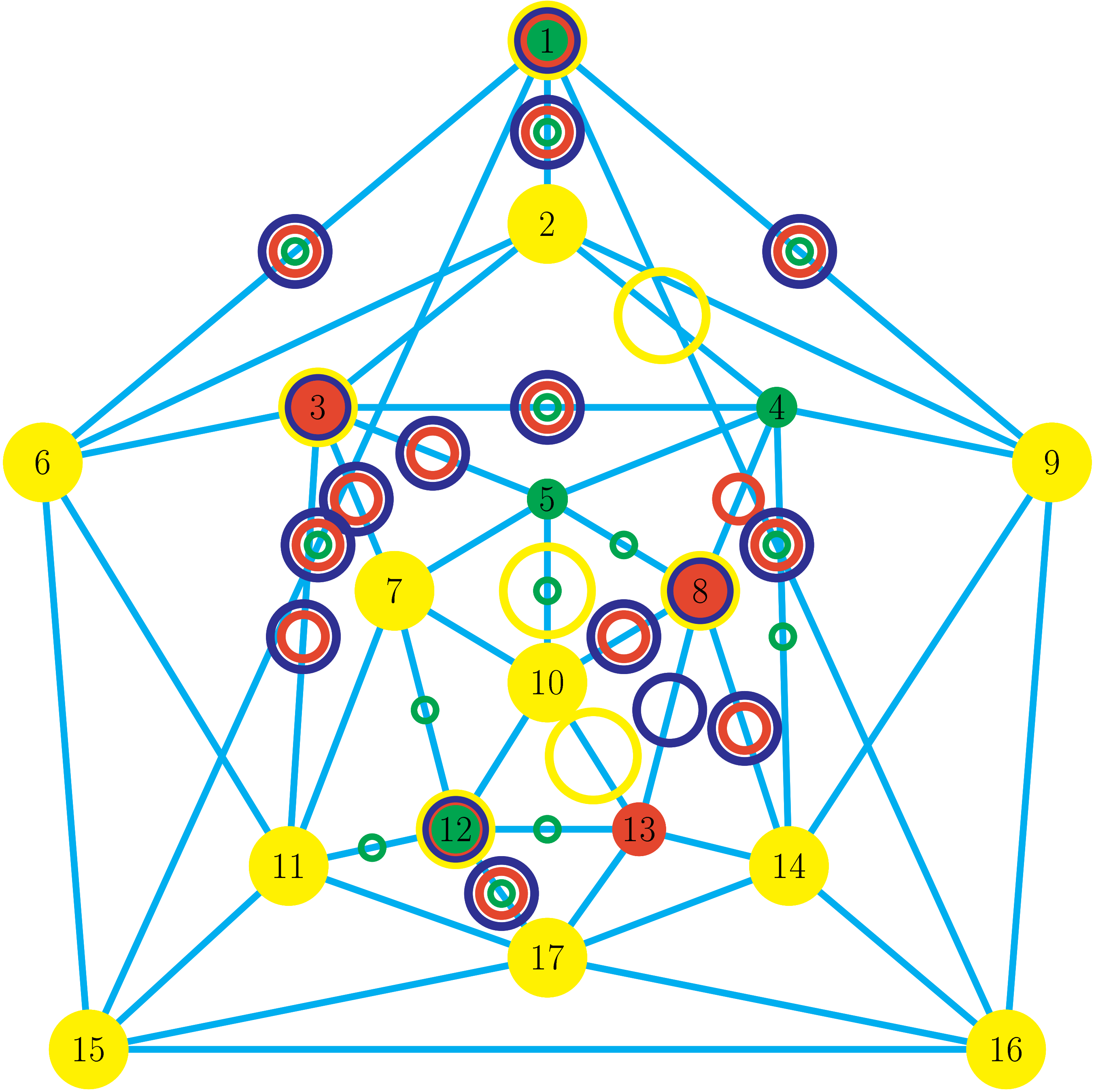


Figure 99.

instruction 196: unplace edge 8-13 Red Checker
instruction 197: place edge 8-4 Red Checker

The Red insertion arrow on edge 10→13 could have bumped out the Red Checker on vertex 8, but we bumped out the Red Checker on edge 13→8 instead, just for variations sake. Properties A and B hold so we can now find the next alternating chain.

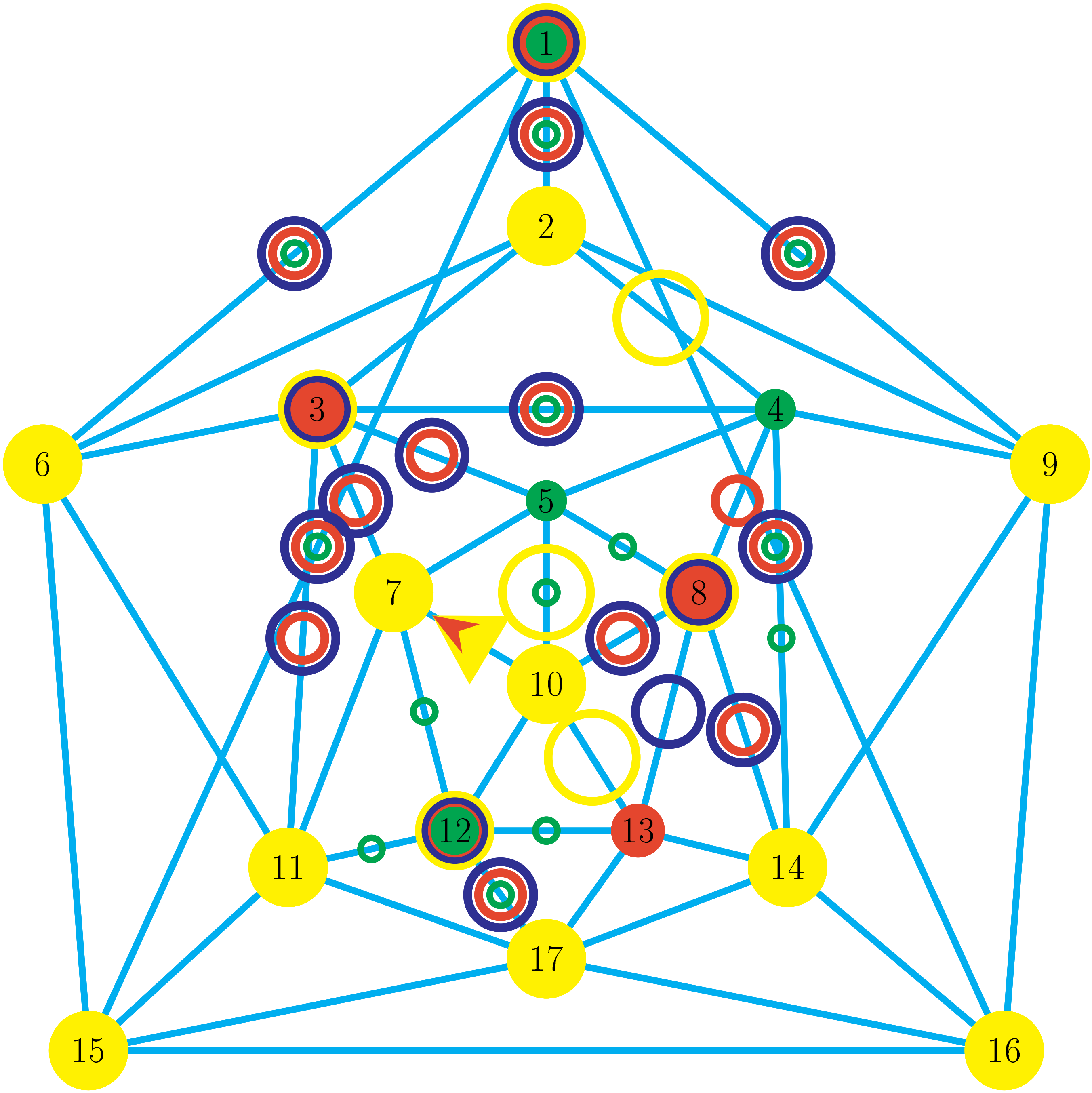


Figure 100.

instruction 198: place edge 10->7 Yellow DeletionArrow
instruction 199: place edge 10->7 Red InsertionArrow

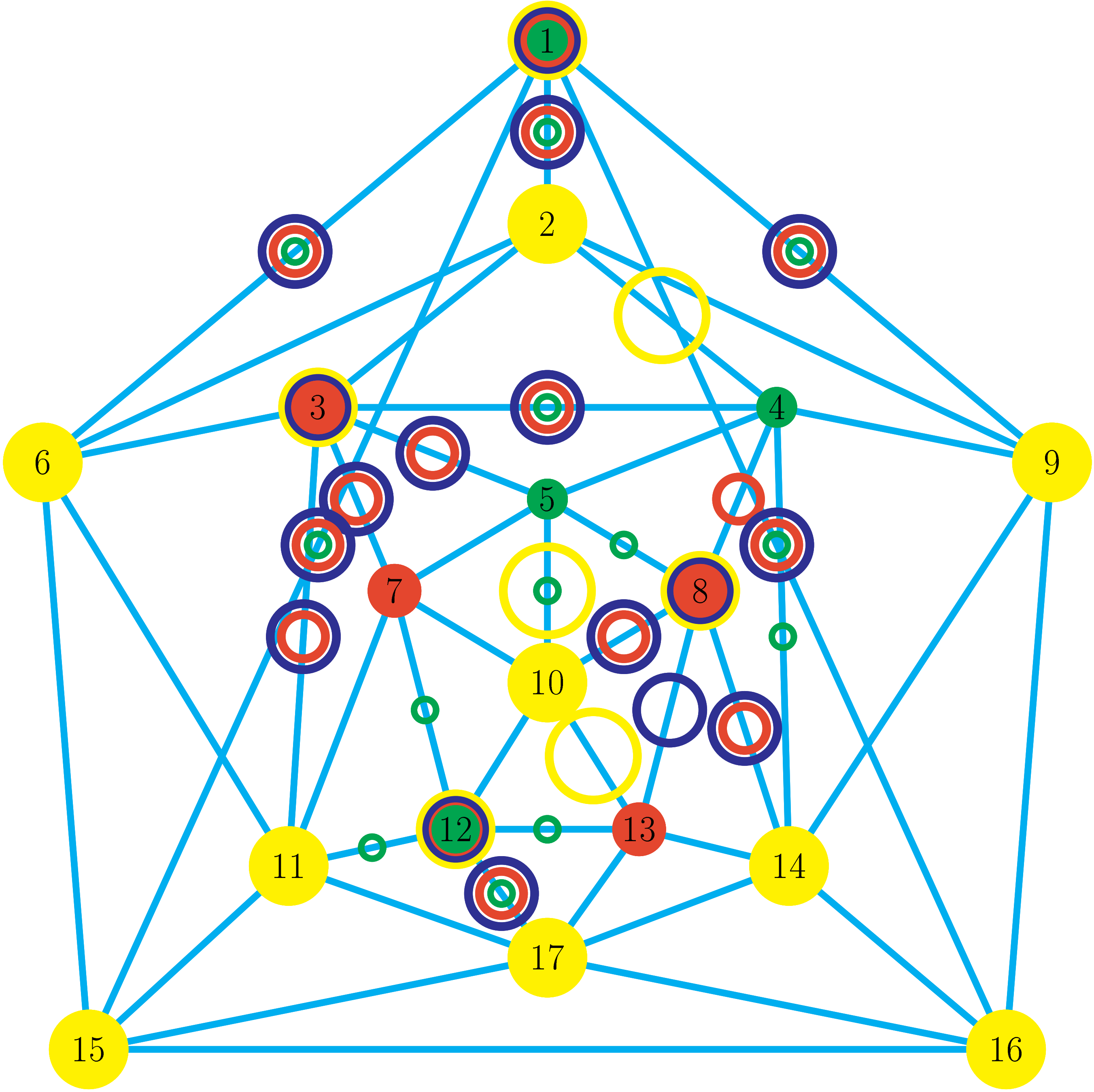


Figure 101.

instruction 200: unplace edge 10->7 Yellow DeletionArrow
instruction 201: unplace edge 10->7 Red InsertionArrow
instruction 202: place vertex 7 Red Checker;
instruction 203: unplace vertex 7 Yellow Checker;

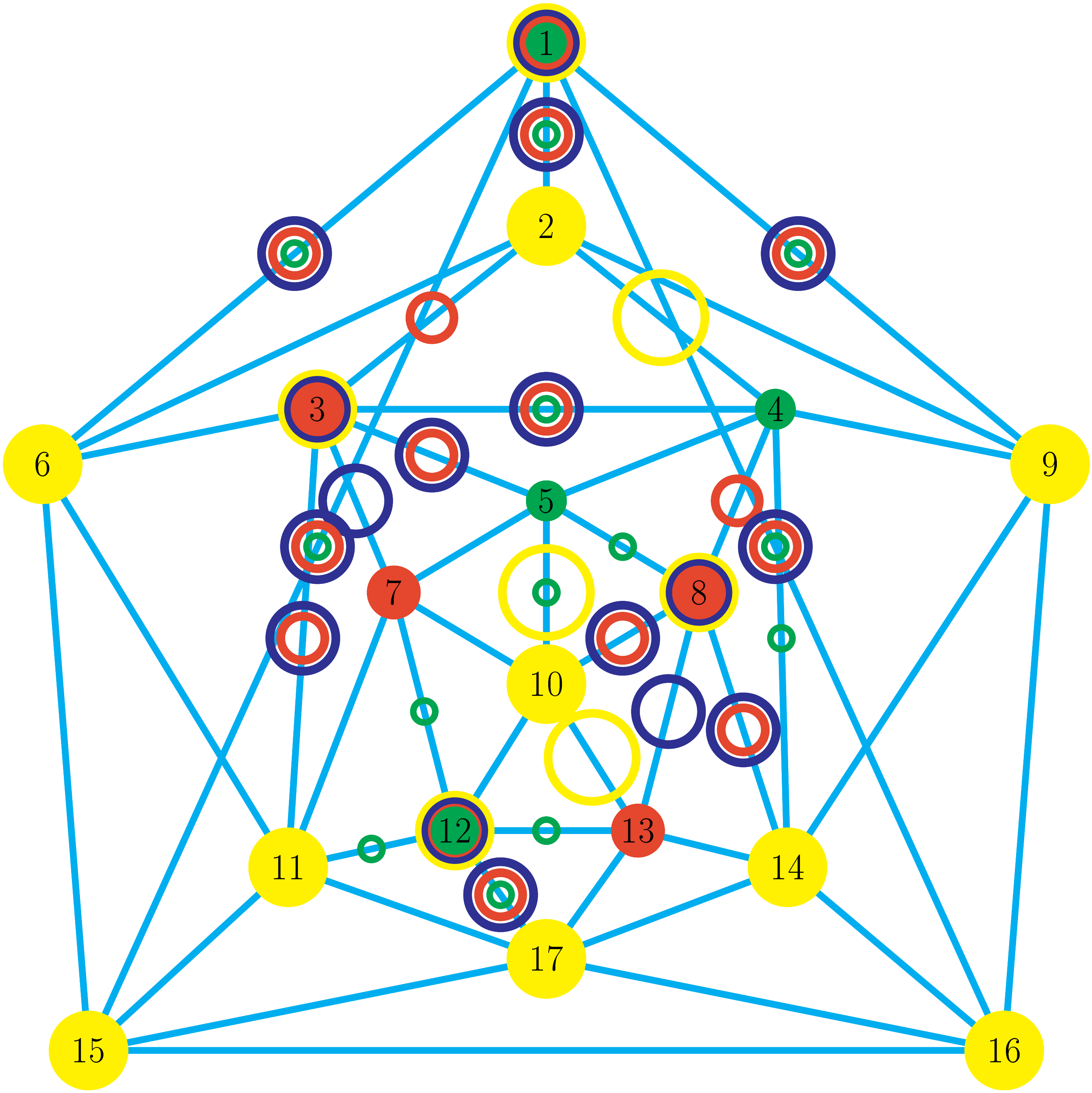


Figure 102.

instruction 204: unplace edge 3-7 Red Checker

instruction 205: place edge 3-2 Red Checker

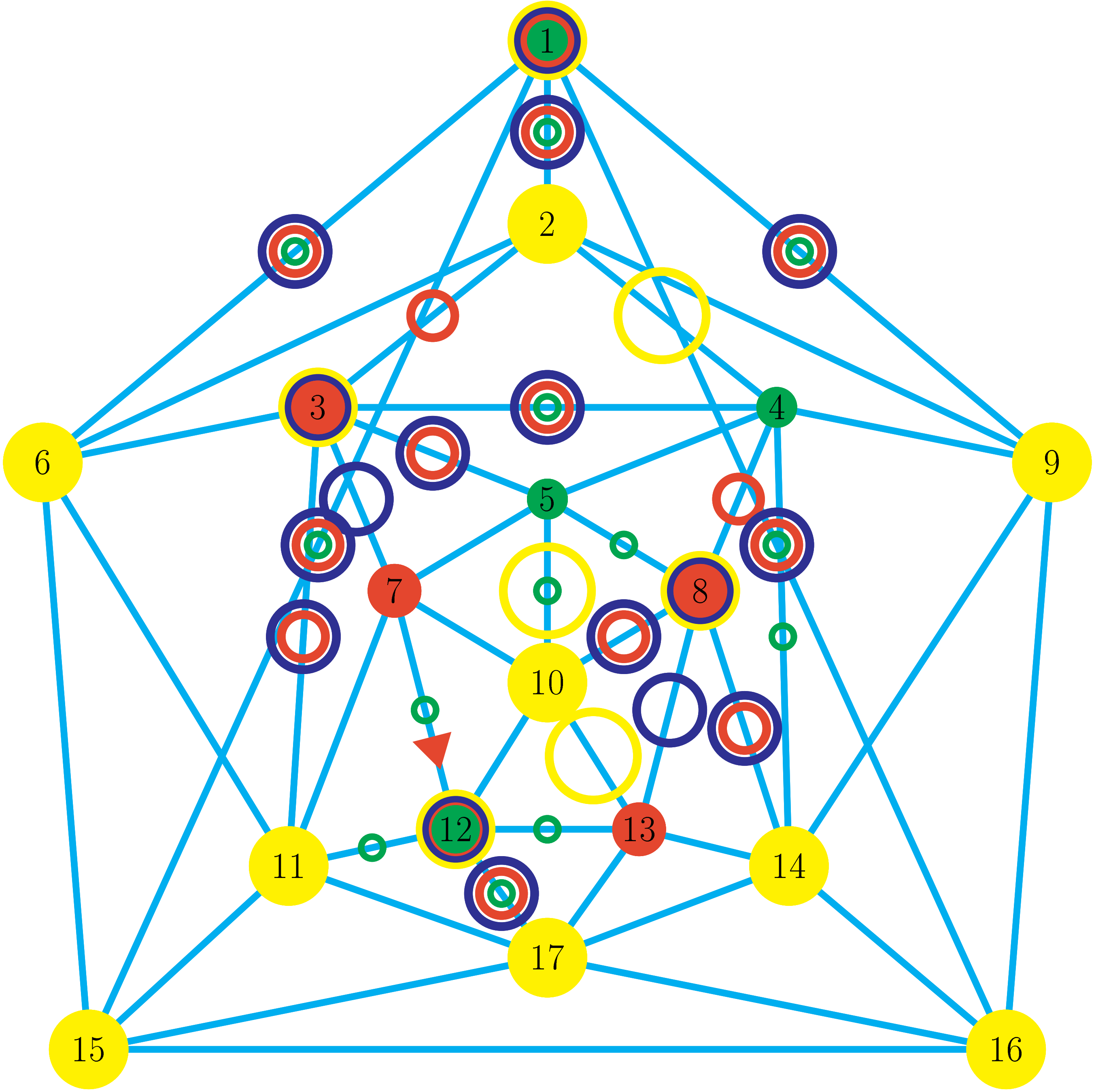


Figure 103.

instruction 206: place edge 7->12 Red DeletionArrow

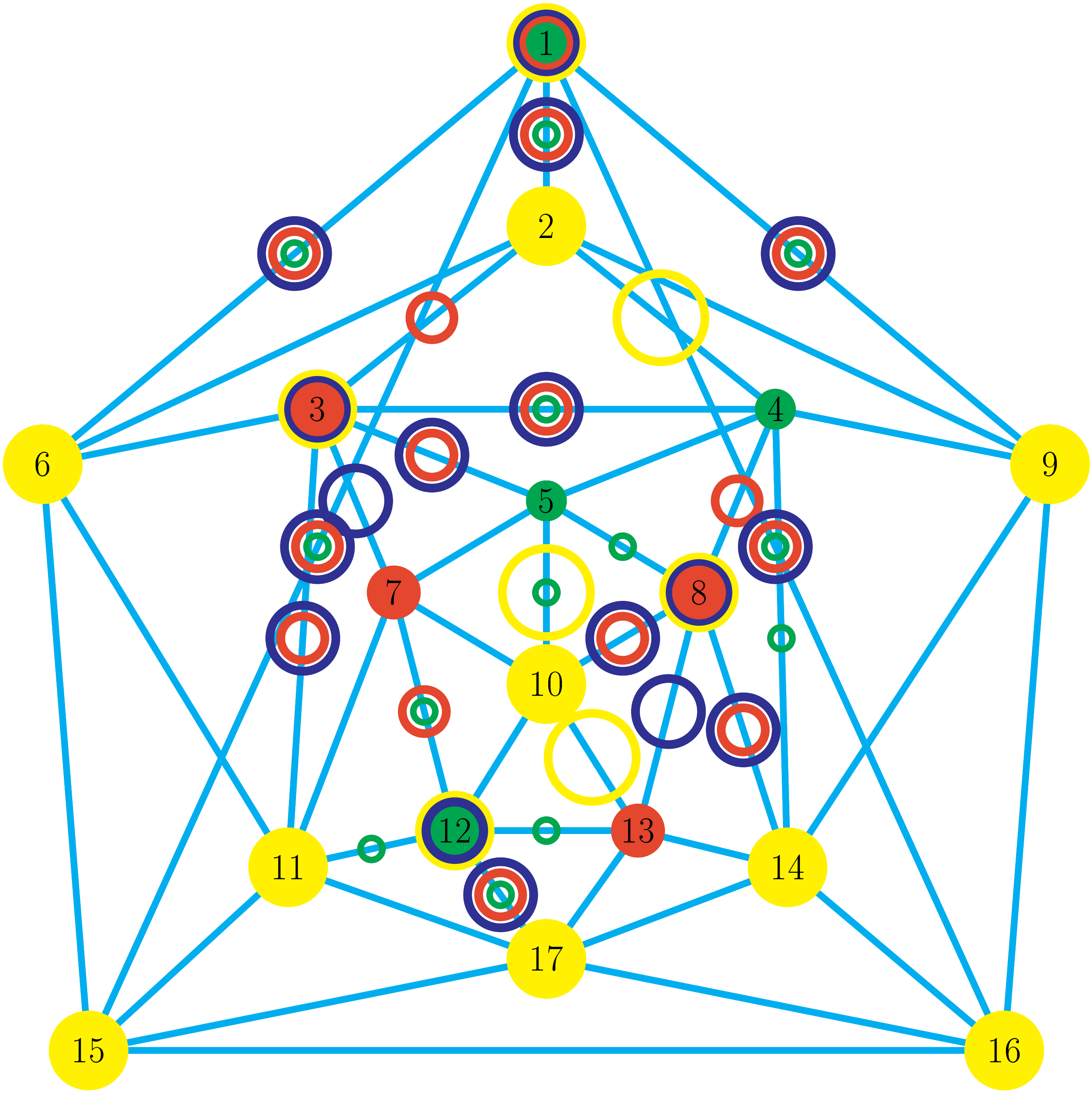


Figure 104.

instruction 207: unplace edge 7->12 Red DeletionArrow
instruction 208: unplace vertex 12 Red Checker;
instruction 209: place edge 7-12 Red Checker

Property B fails to hold on edge 12-17, because of the Red Checker removed from vertex 12.

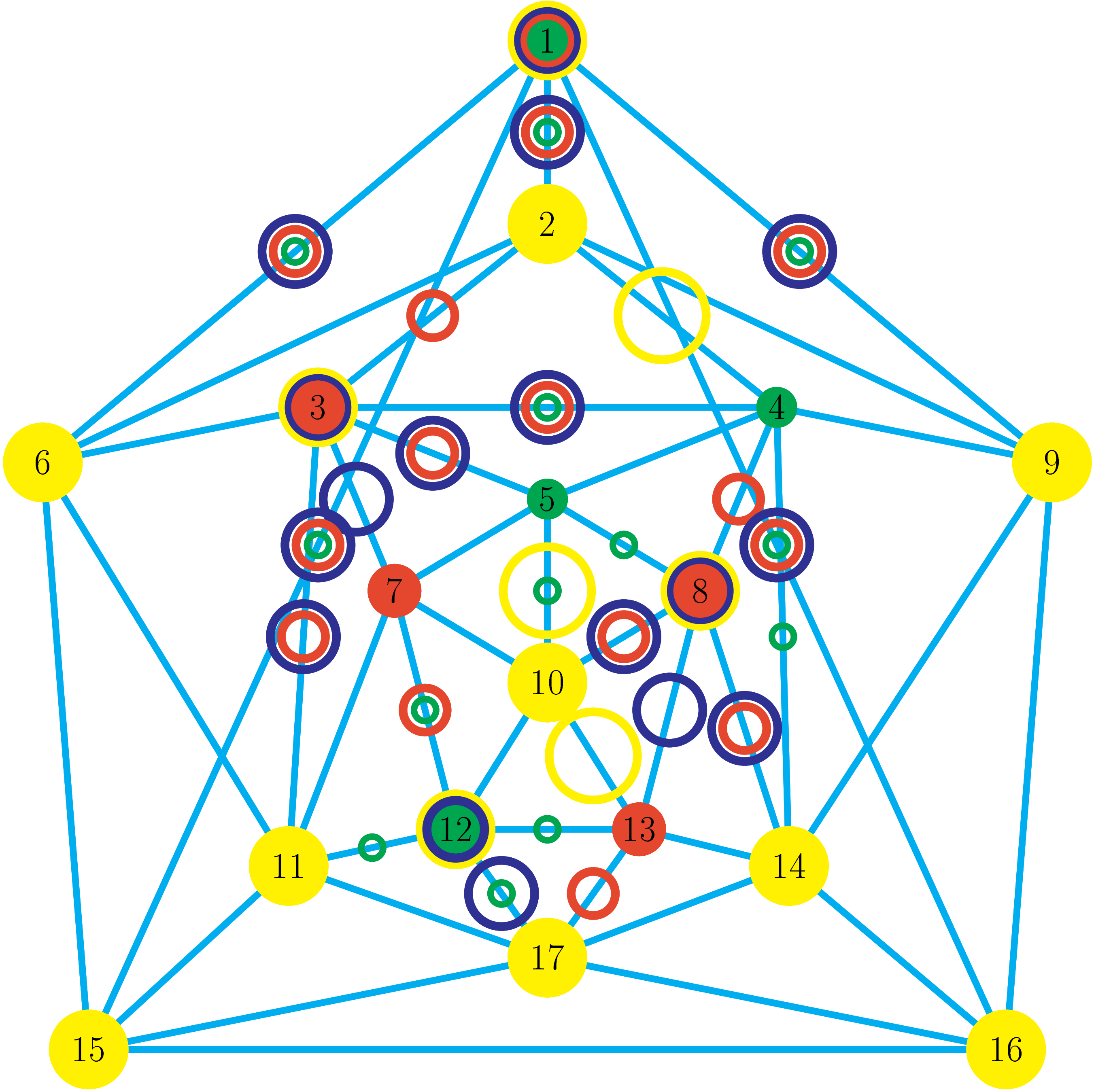


Figure 105.
instruction 210: unplace edge 12-17 Red Checker
instruction 211: place edge 13-17 Red Checker

Property B is restored and we can now proceed to find the next alternating chain.

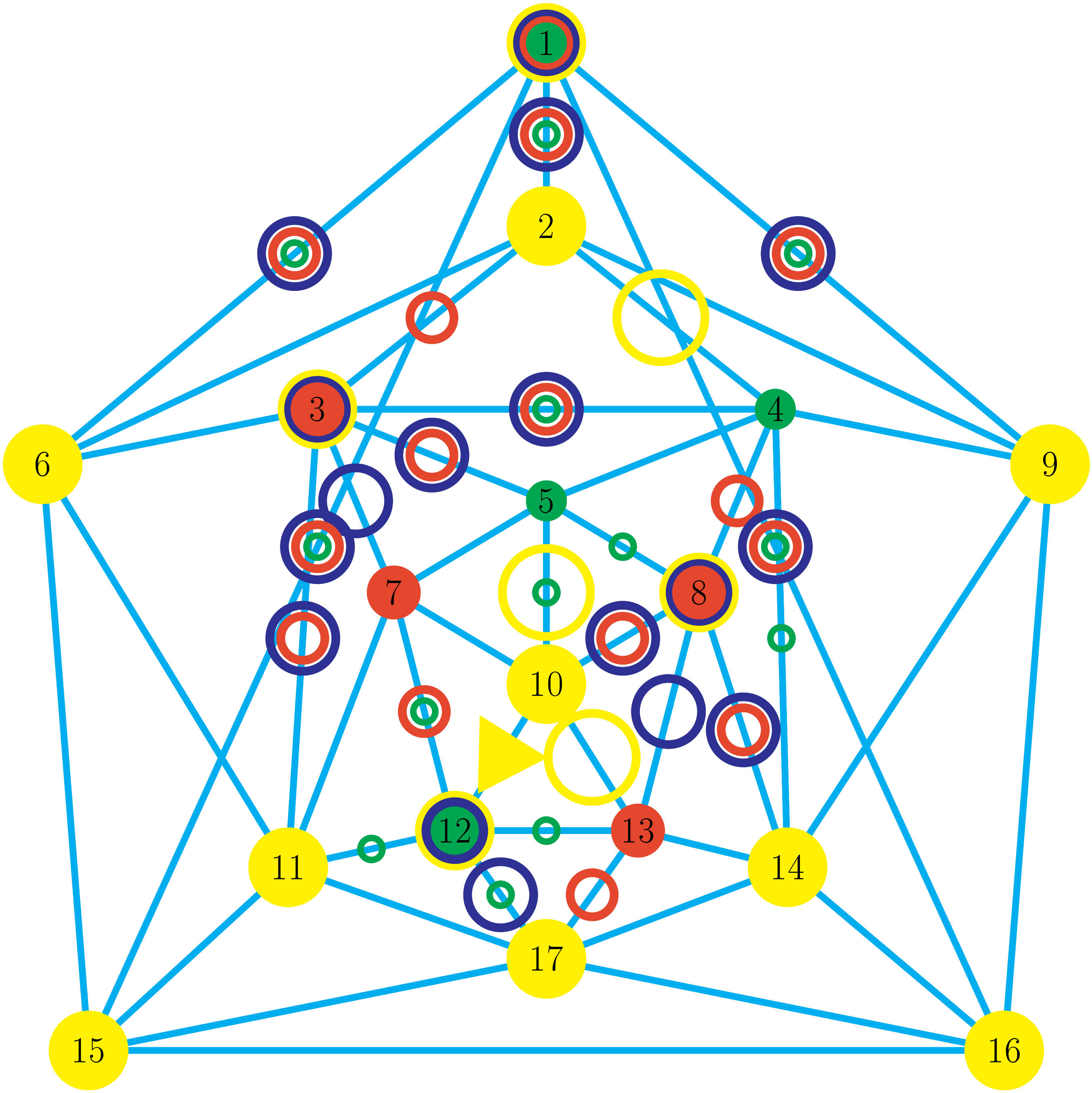


Figure 106.

instruction 212: place edge 10->12 Yellow DeletionArrow

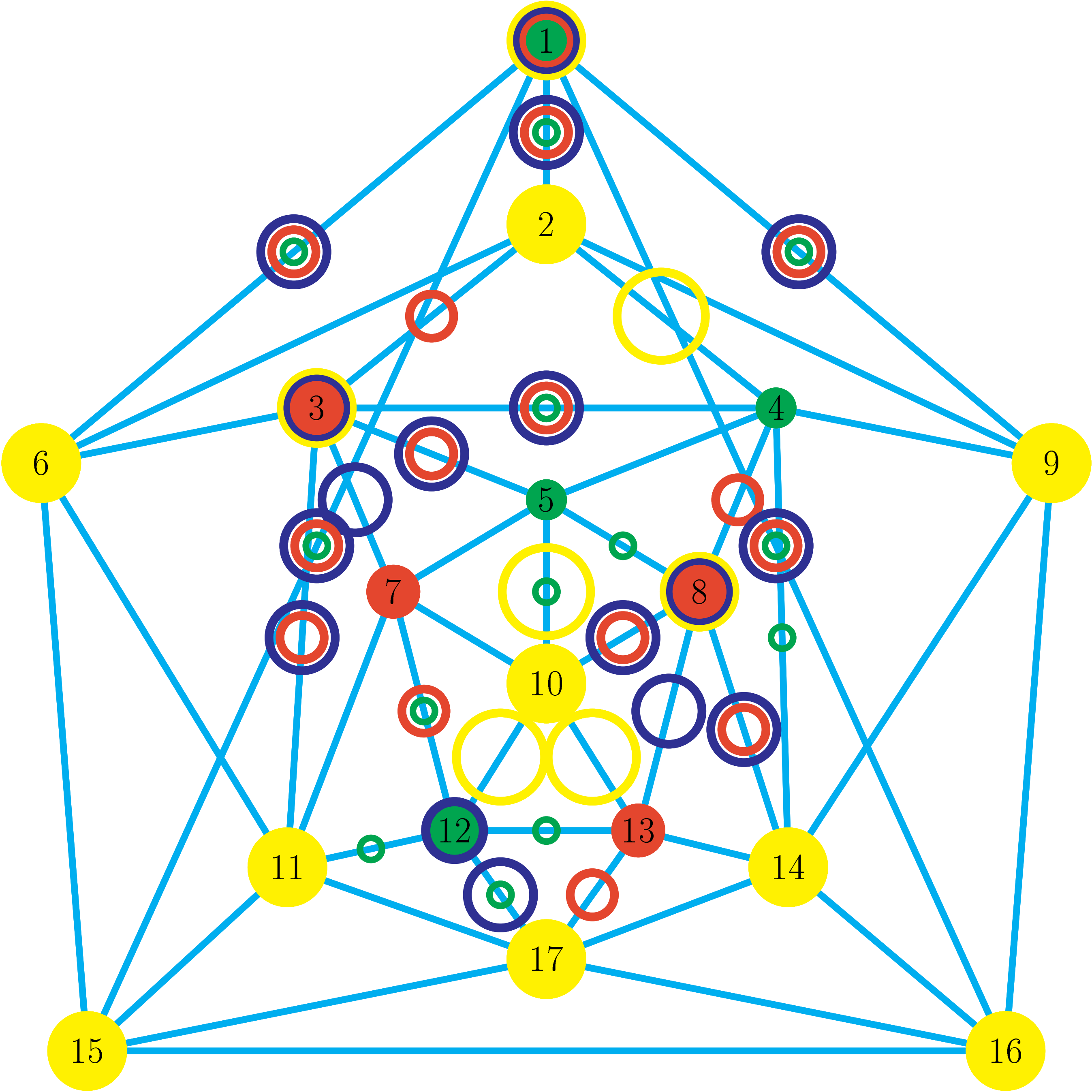


Figure 107.

instruction 213: unplace edge 10->12 Yellow DeletionArrow
instruction 214: unplace vertex 12 Yellow Checker;
instruction 215: place edge 10-12 Yellow Checker

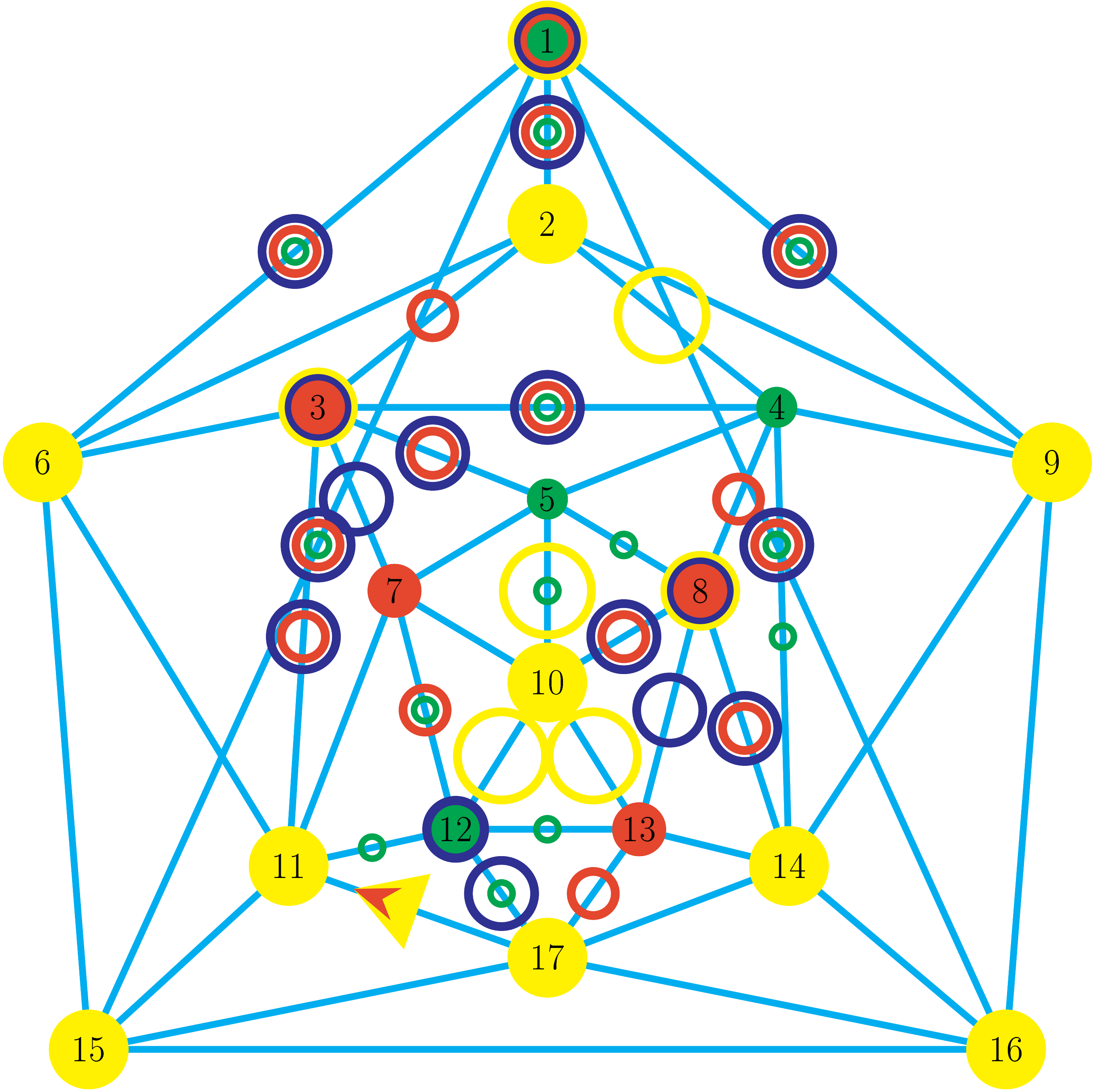


Figure 108.

instruction 216: place edge 17->11 Yellow DeletionArrow

instruction 217: place edge 17->11 Red InsertionArrow

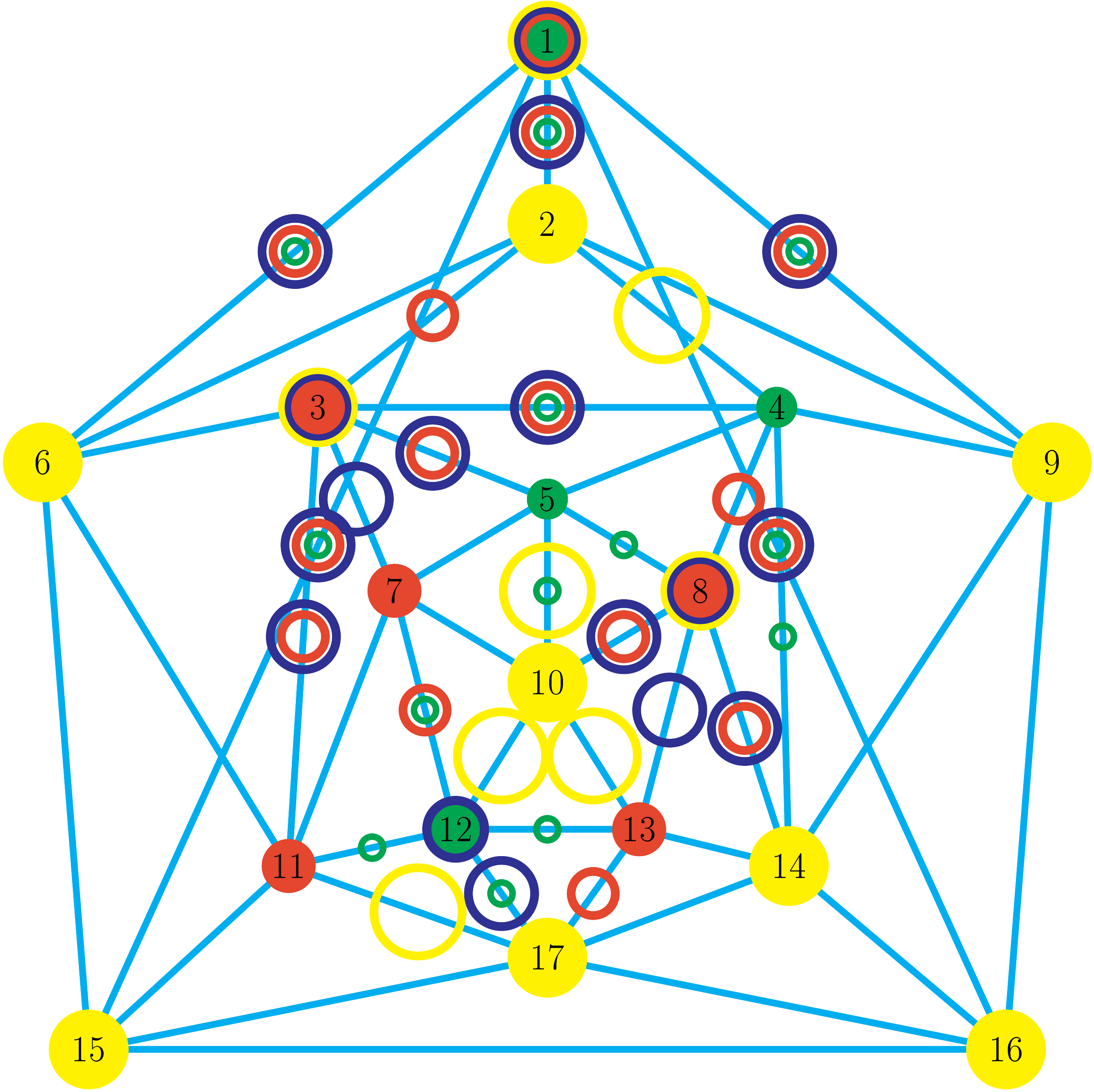


FIGURE 109.

instruction 218: unplace edge 17->11 Yellow DeletionArrow
instruction 219: unplace edge 17->11 Red InsertionArrow
instruction 220: place edge 17-11 Yellow Checker
instruction 221: unplace vertex 11 Yellow Checker;
instruction 222: place vertex 11 Red Checker;

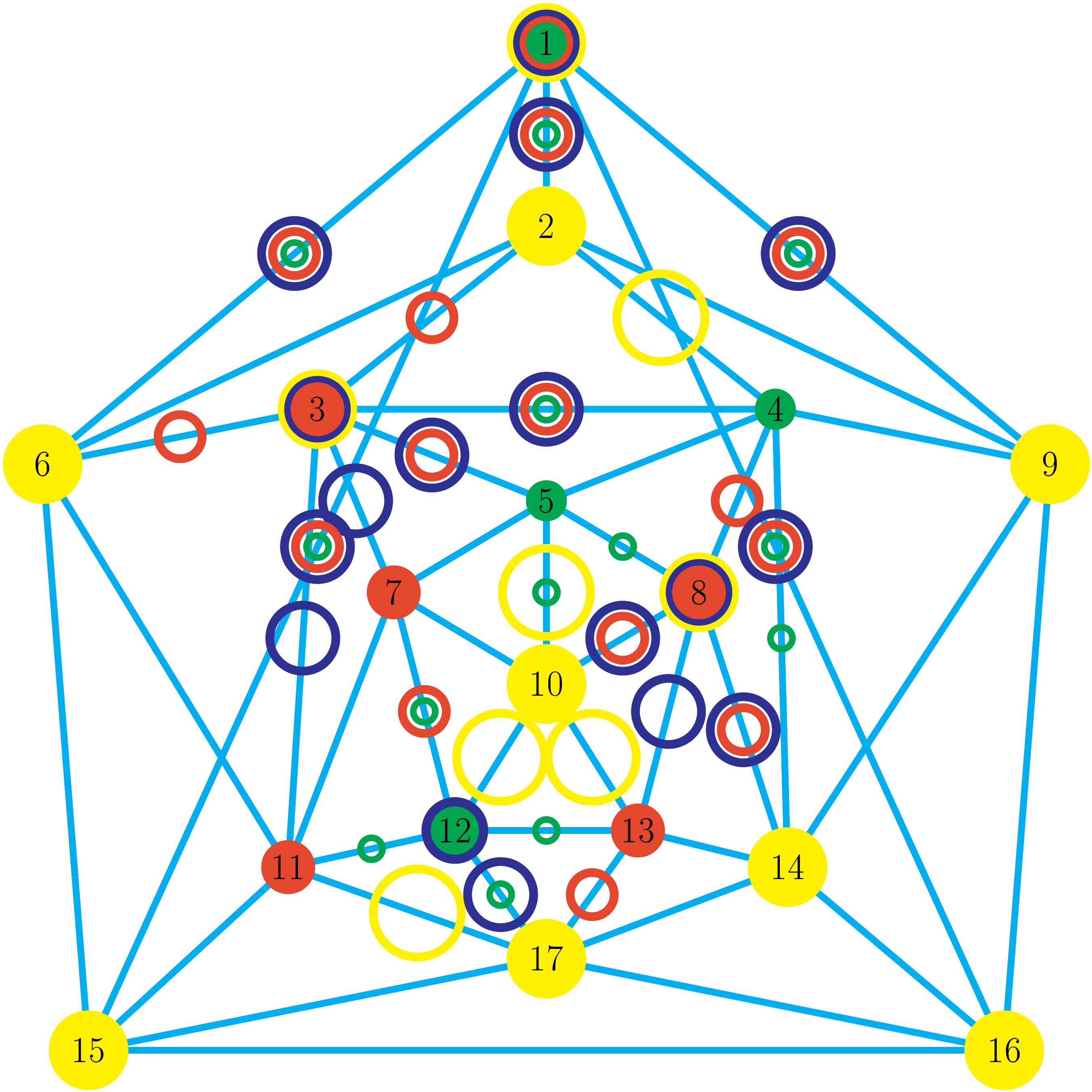


Figure 110.

instruction 223: unplace edge 3-11 Red Checker
instruction 224: place edge 3-6 Red Checker

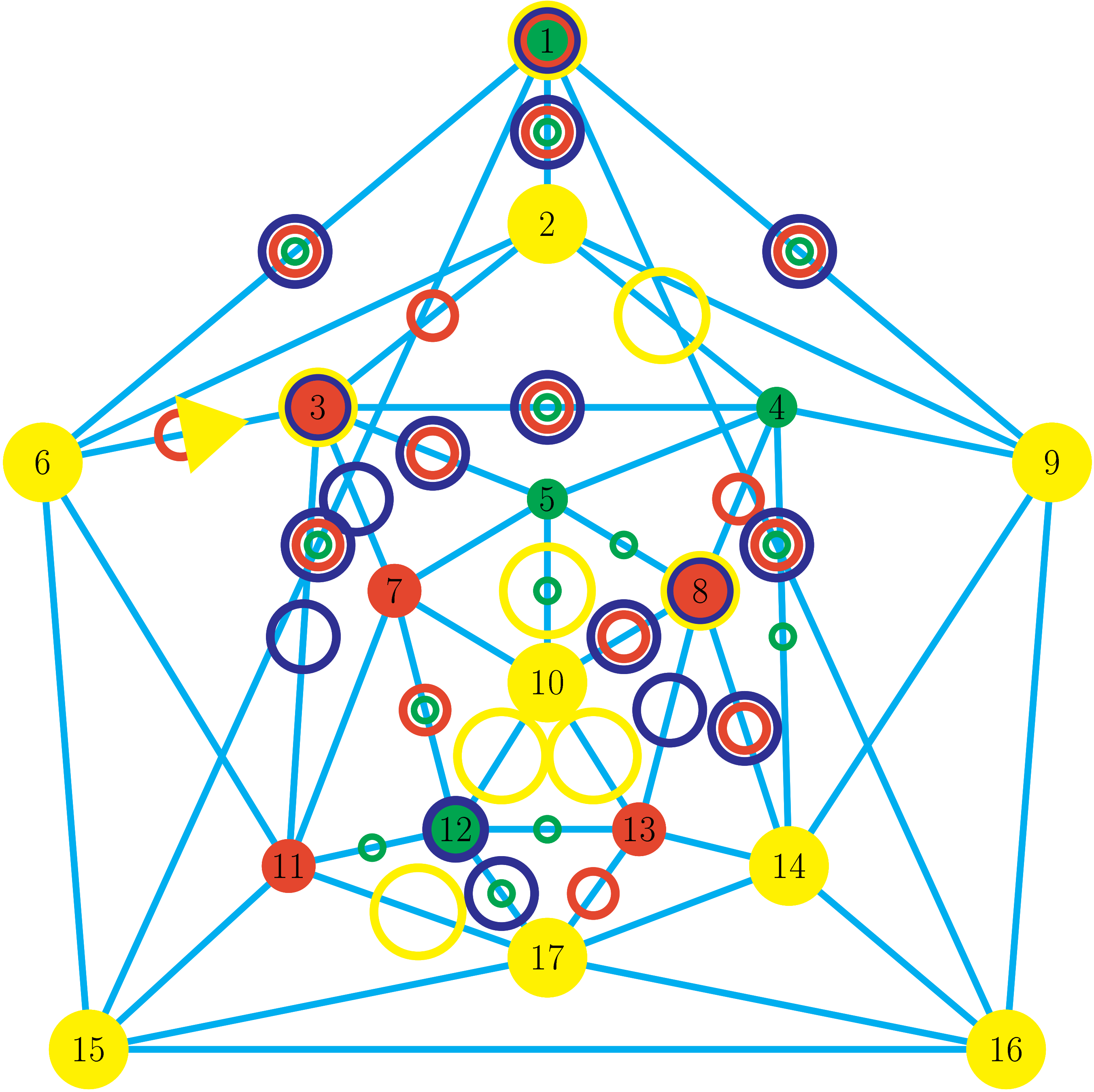


Figure 111.

instruction 225: place edge 6->3 Yellow DeletionArrow

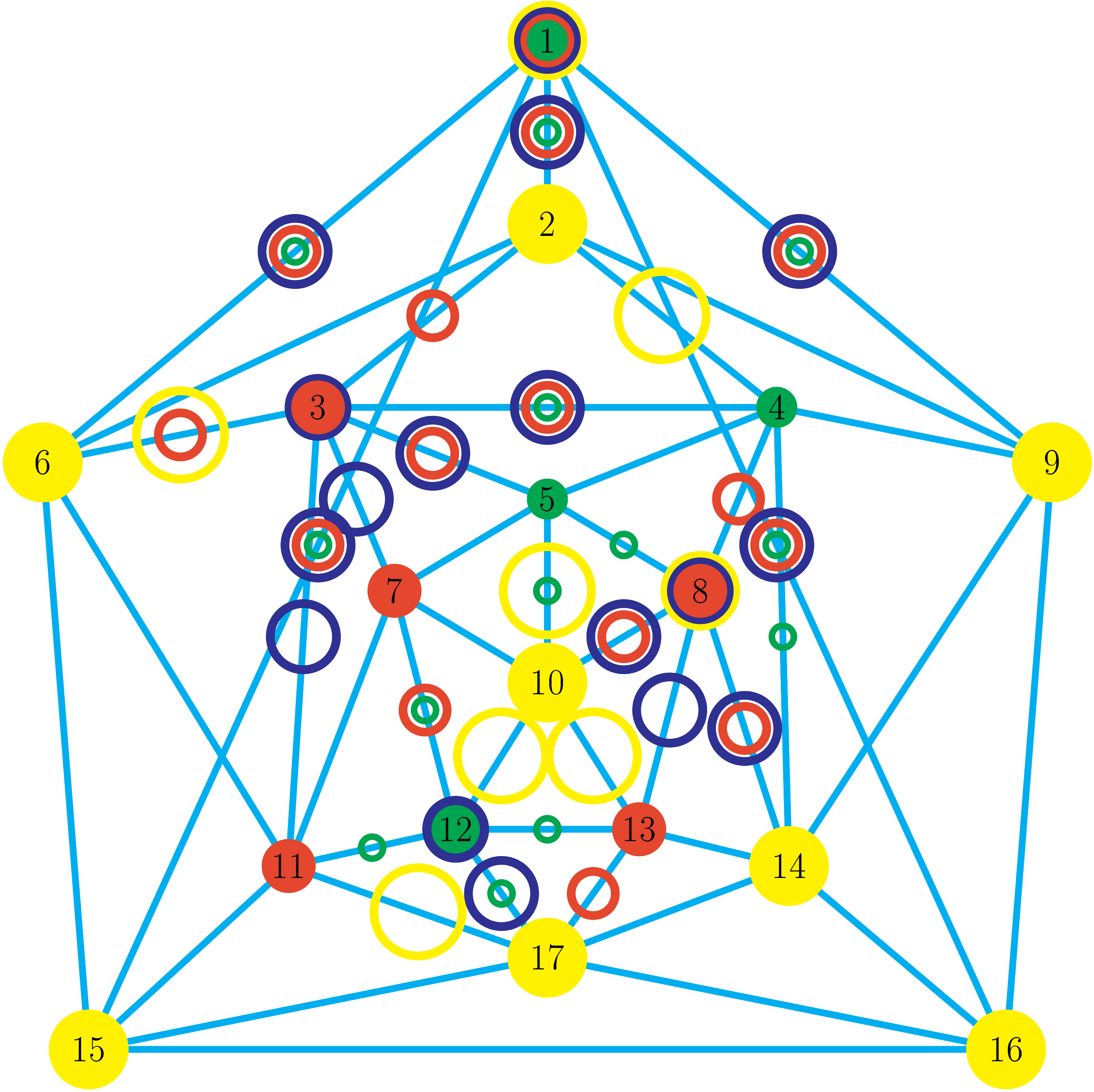


Figure 112.

instruction 226: unplace edge 6->3 Yellow DeletionArrow
instruction 227: unplace vertex 3 Yellow Checker;
instruction 228: place edge 6-3 Yellow Checker

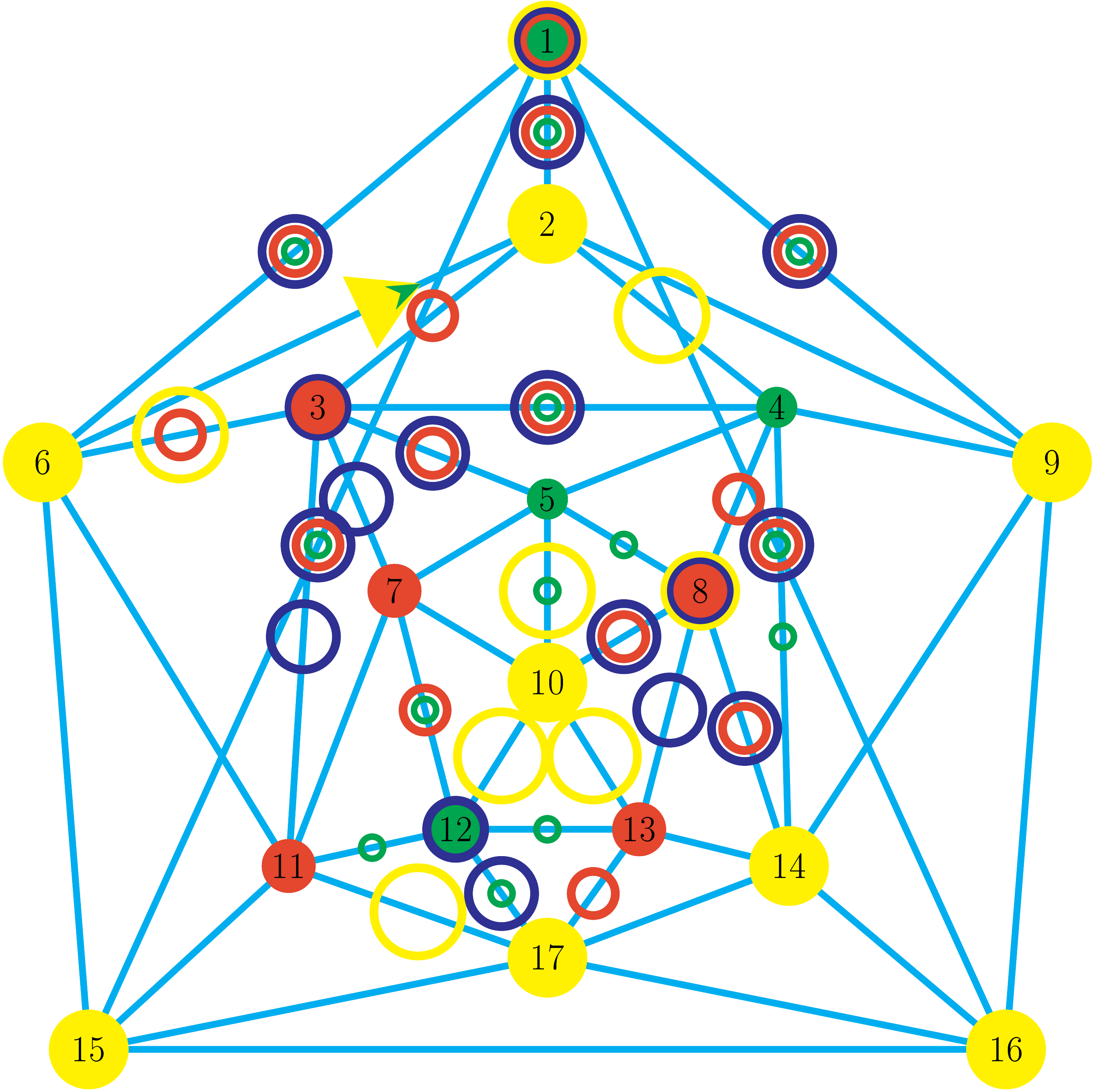


Figure 113.

instruction 229: place edge 6->2 Yellow DeletionArrow
instruction 230: place edge 6->2 Green InsertionArrow

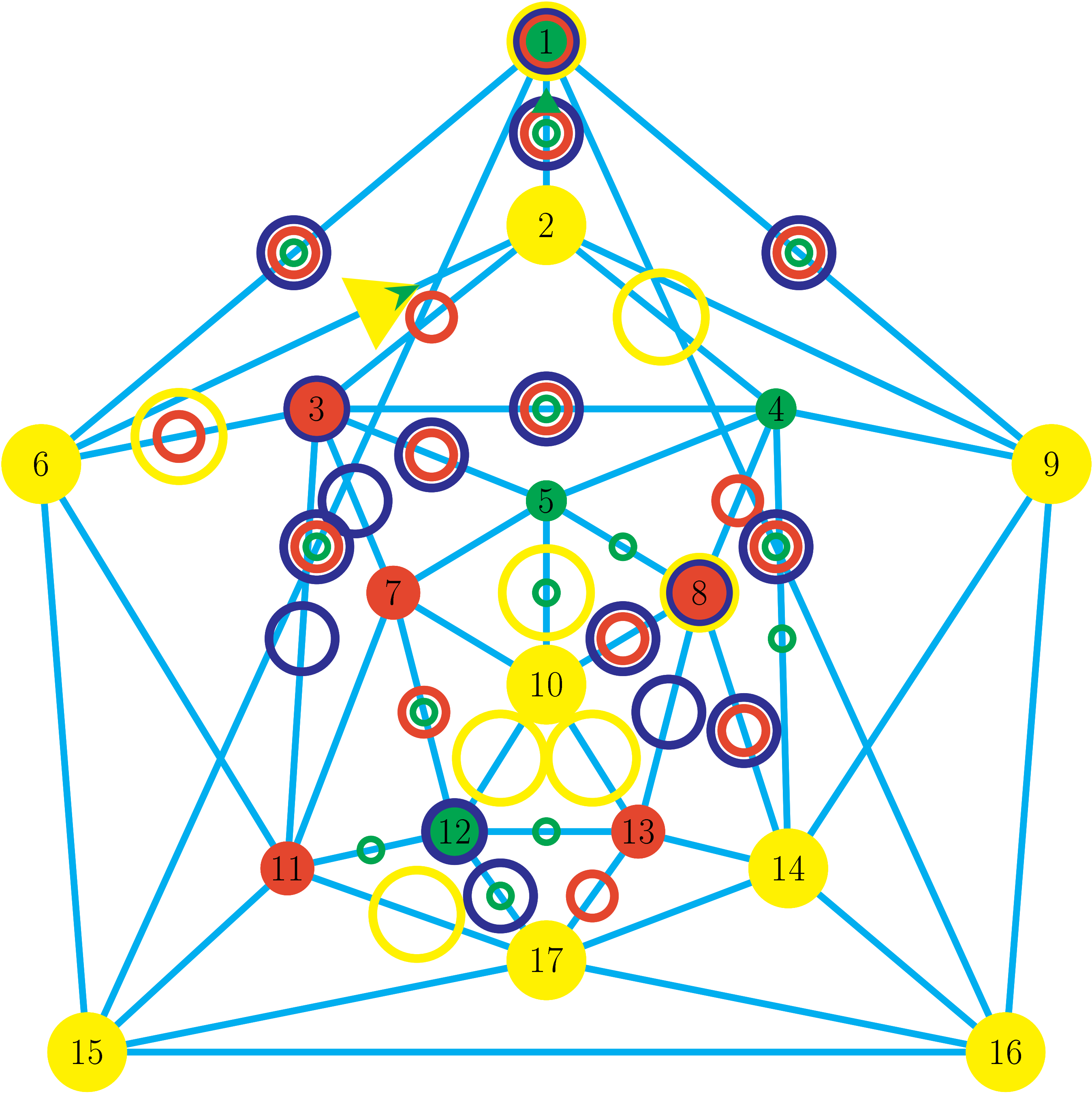


Figure 114.

instruction 231: place edge 2->1 Green DeletionArrow

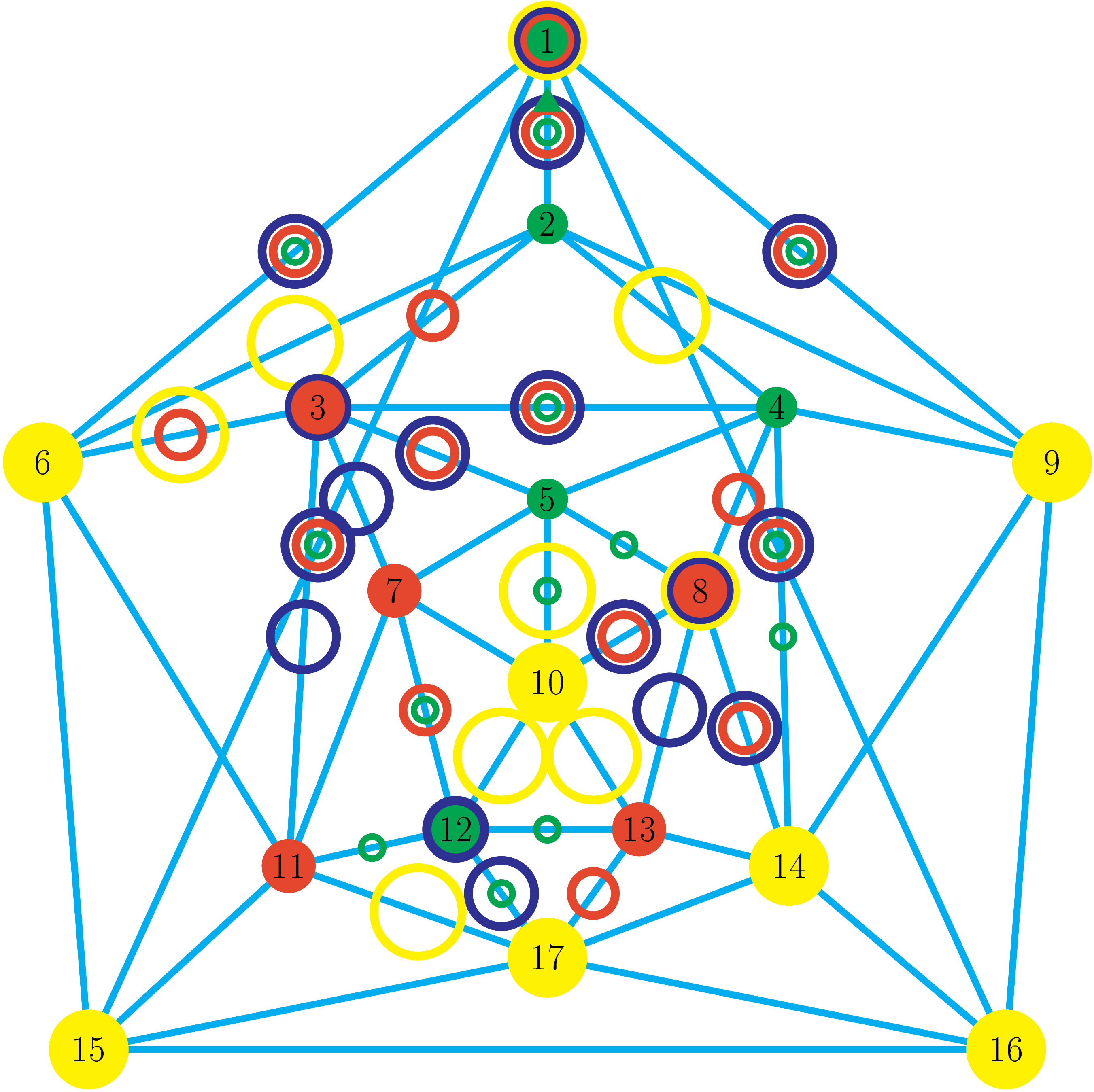


Figure 115.

instruction 232: unplace edge 6->2 Yellow DeletionArrow

instruction 233: unplace edge 6->2 Green InsertionArrow

instruction 234: unplace vertex 2 Yellow Checker;

instruction 235: place vertex 2 Green Checker;

instruction 236: place edge 6-2 Yellow Checker

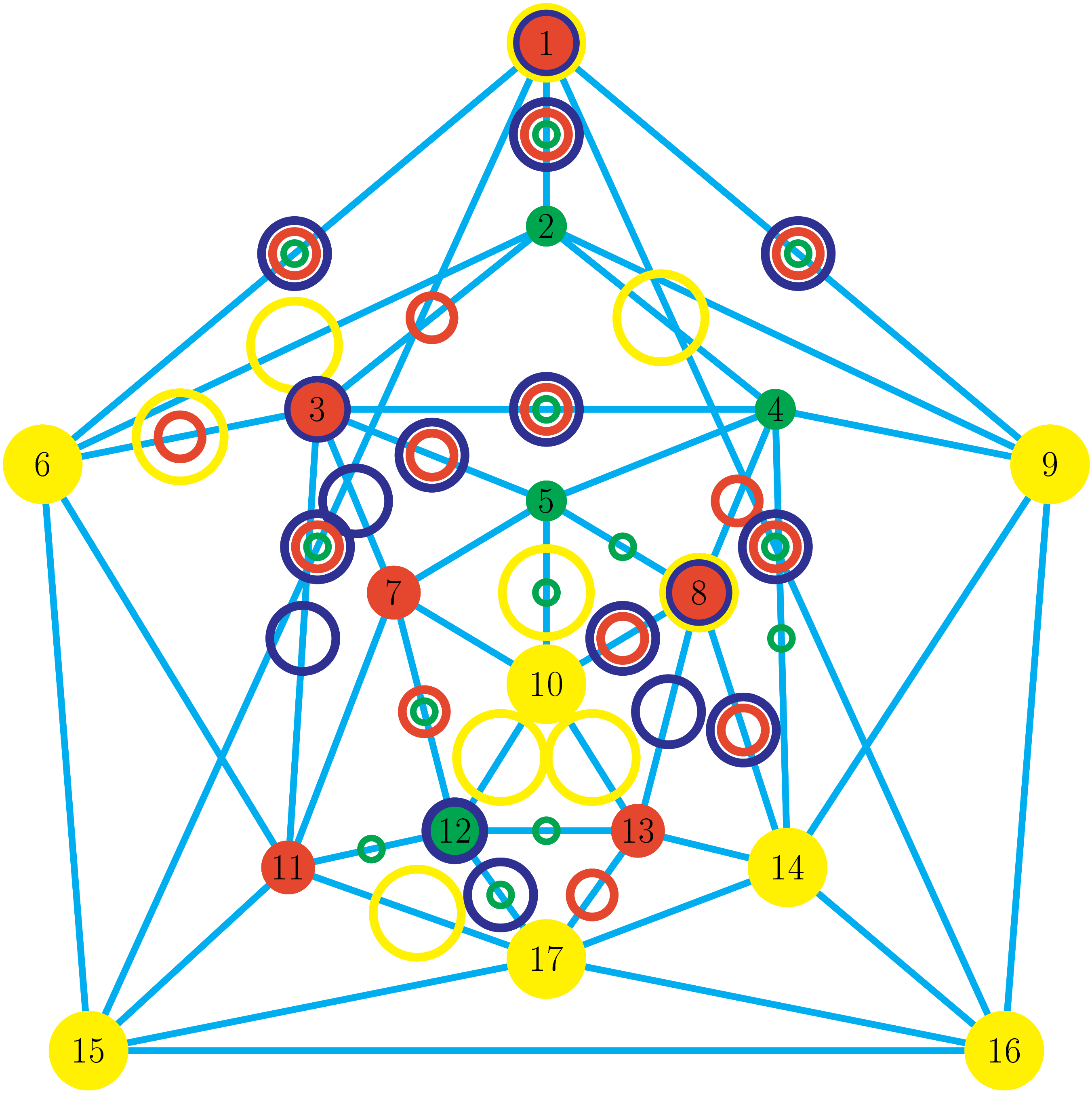


Figure 116.

instruction 237: unplace edge 2->1 Green DeletionArrow
instruction 238: unplace vertex 1 Green Checker;

Property B fails, it is restored as follows.

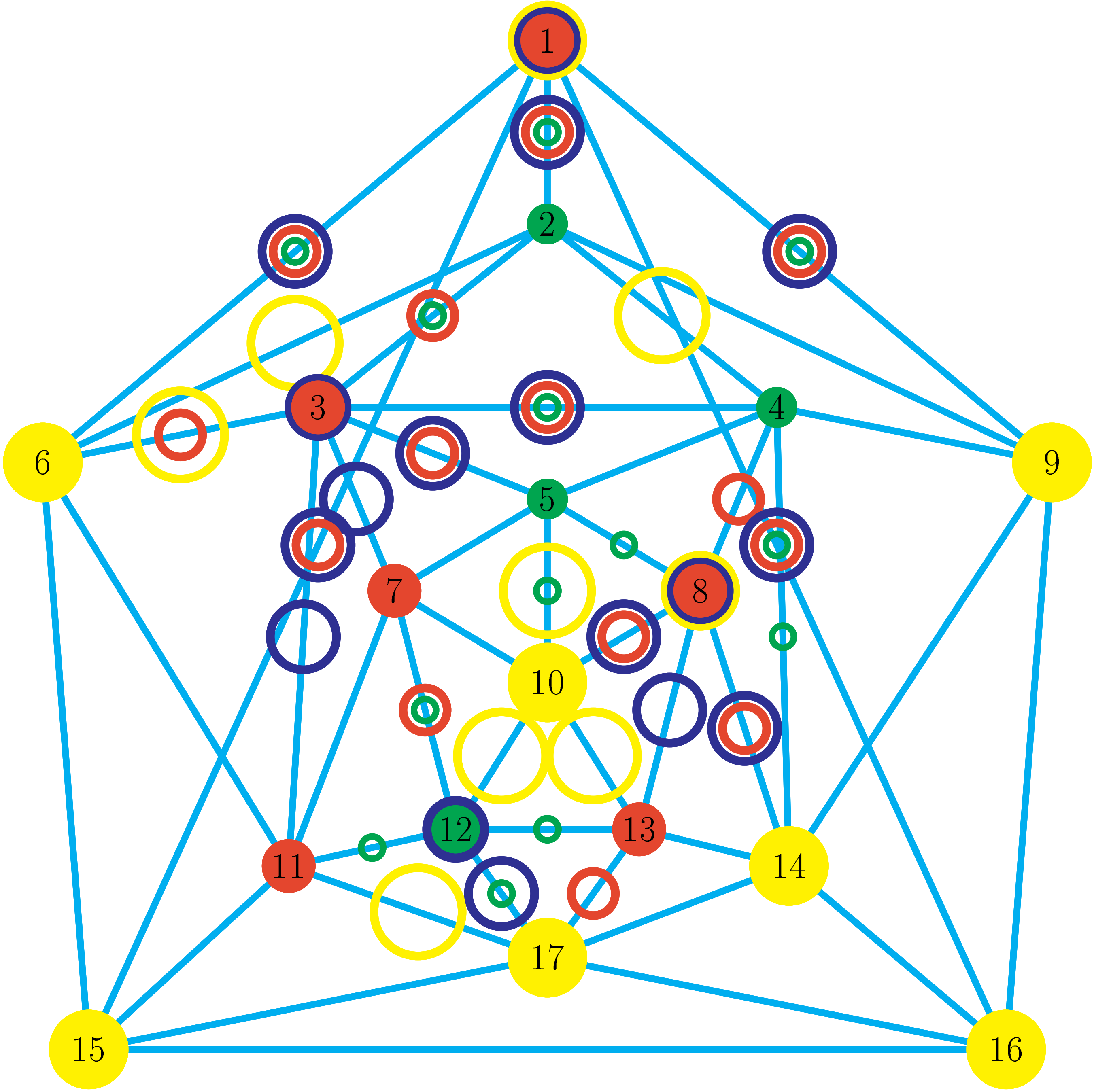


Figure 117.

instruction 239: unplace edge 1-15 Green Checker
instruction 240: place edge 2-3 Green Checker

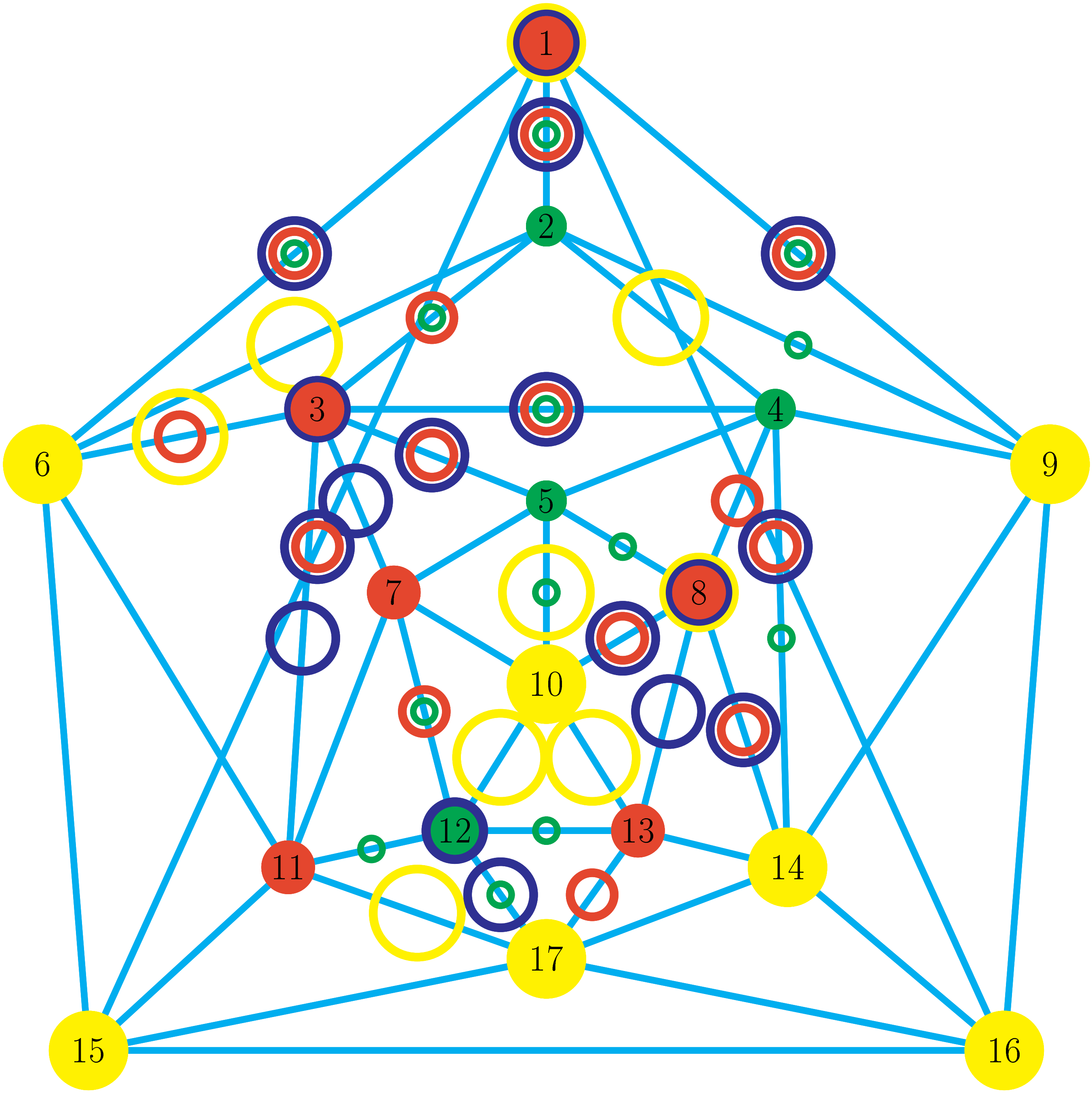


Figure 118.

instruction 241: unplace edge 1-16 Green Checker

instruction 242: place edge 2-9 Green Checker

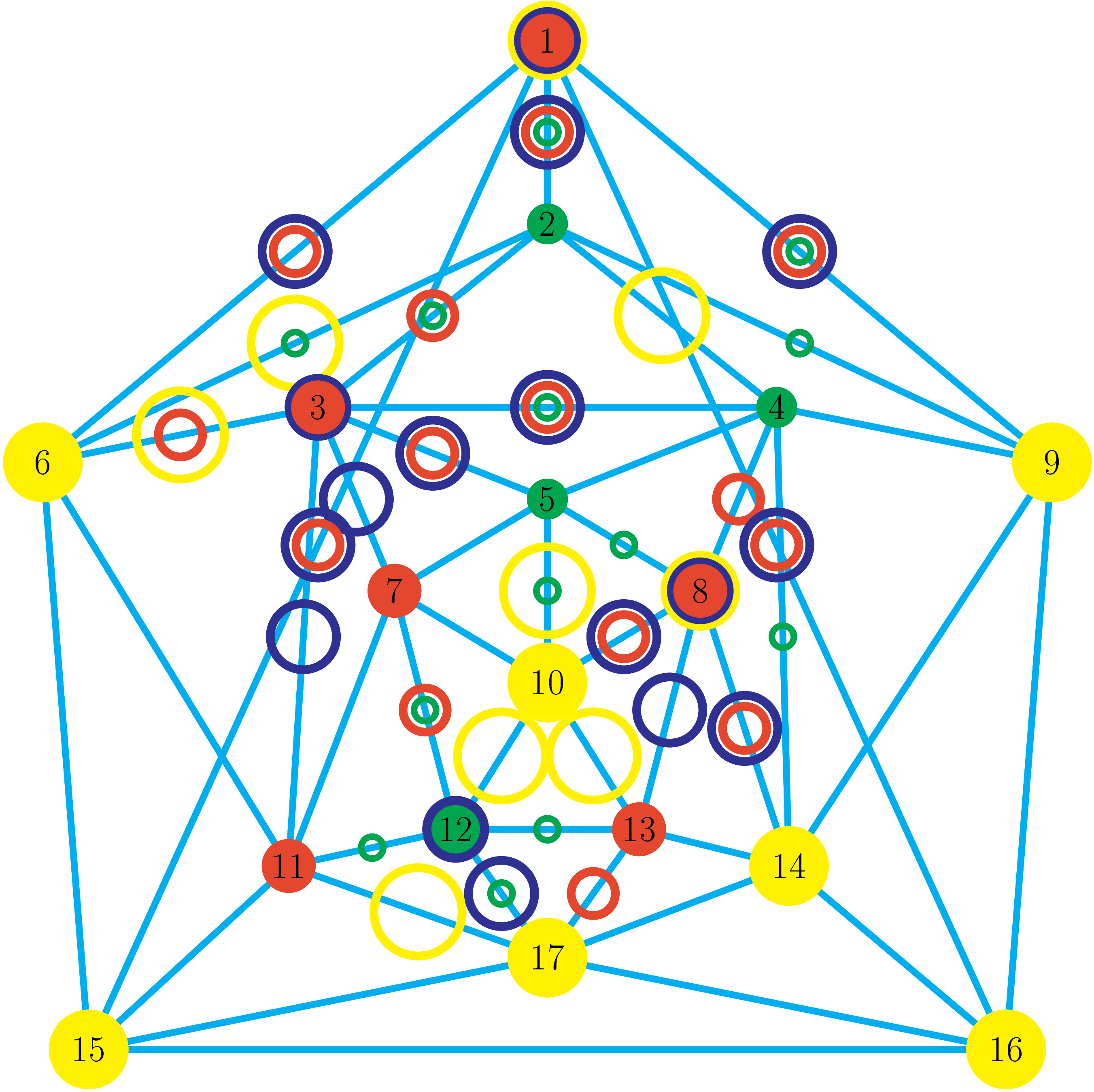


Figure 119.

instruction 243: unplace edge 6-1 Green Checker
instruction 244: place edge 6-2 Green Checker

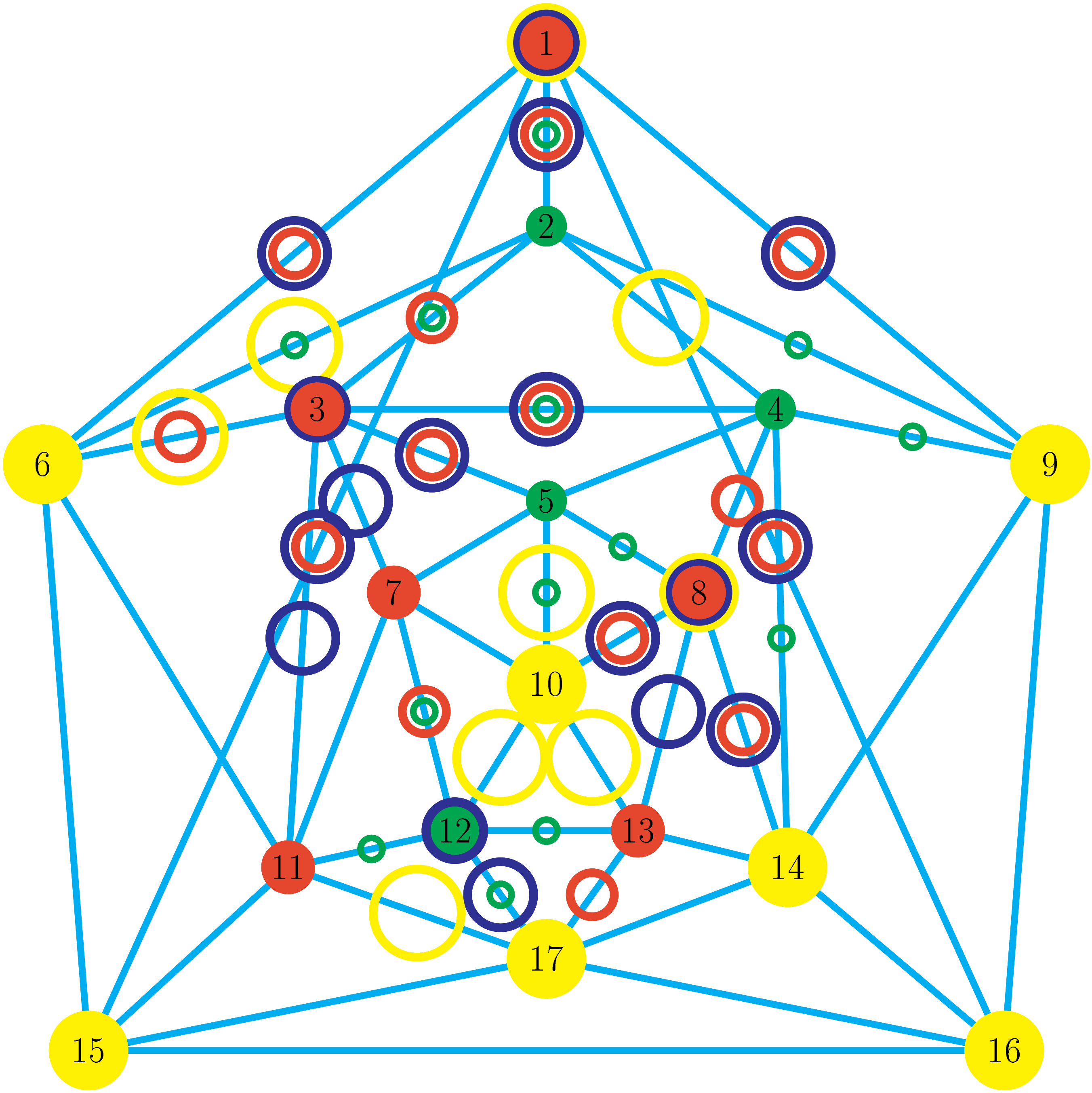


Figure 120.

instruction 245: unplace edge 9-1 Green Checker
instruction 246: place edge 9-4 Green Checker

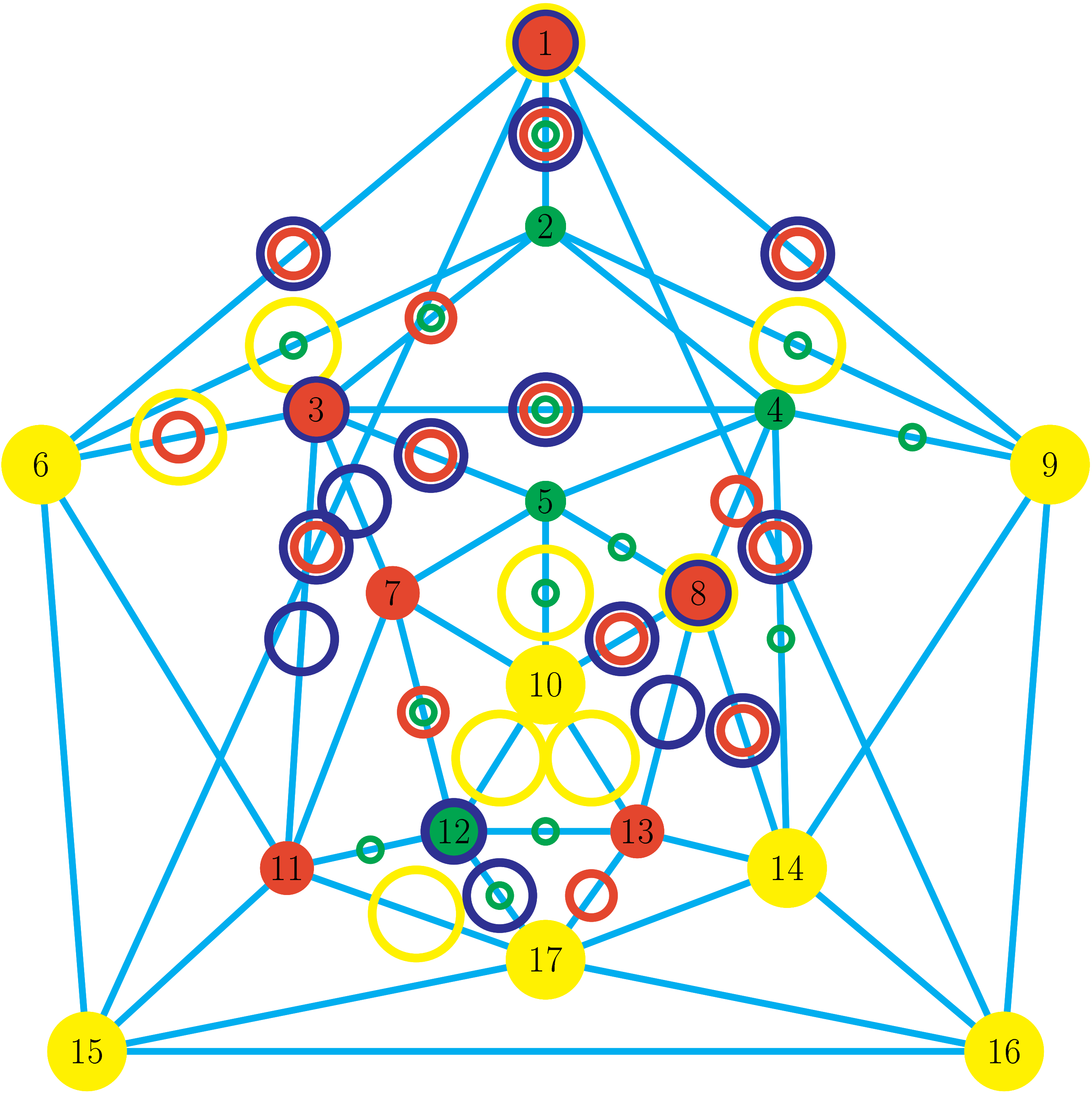


Figure 121.

instruction 247: unplace edge 2-4 Yellow Checker
instruction 248: place edge 2-9 Yellow Checker

Property B is restored and we can now proceed to find the next alternating chain.

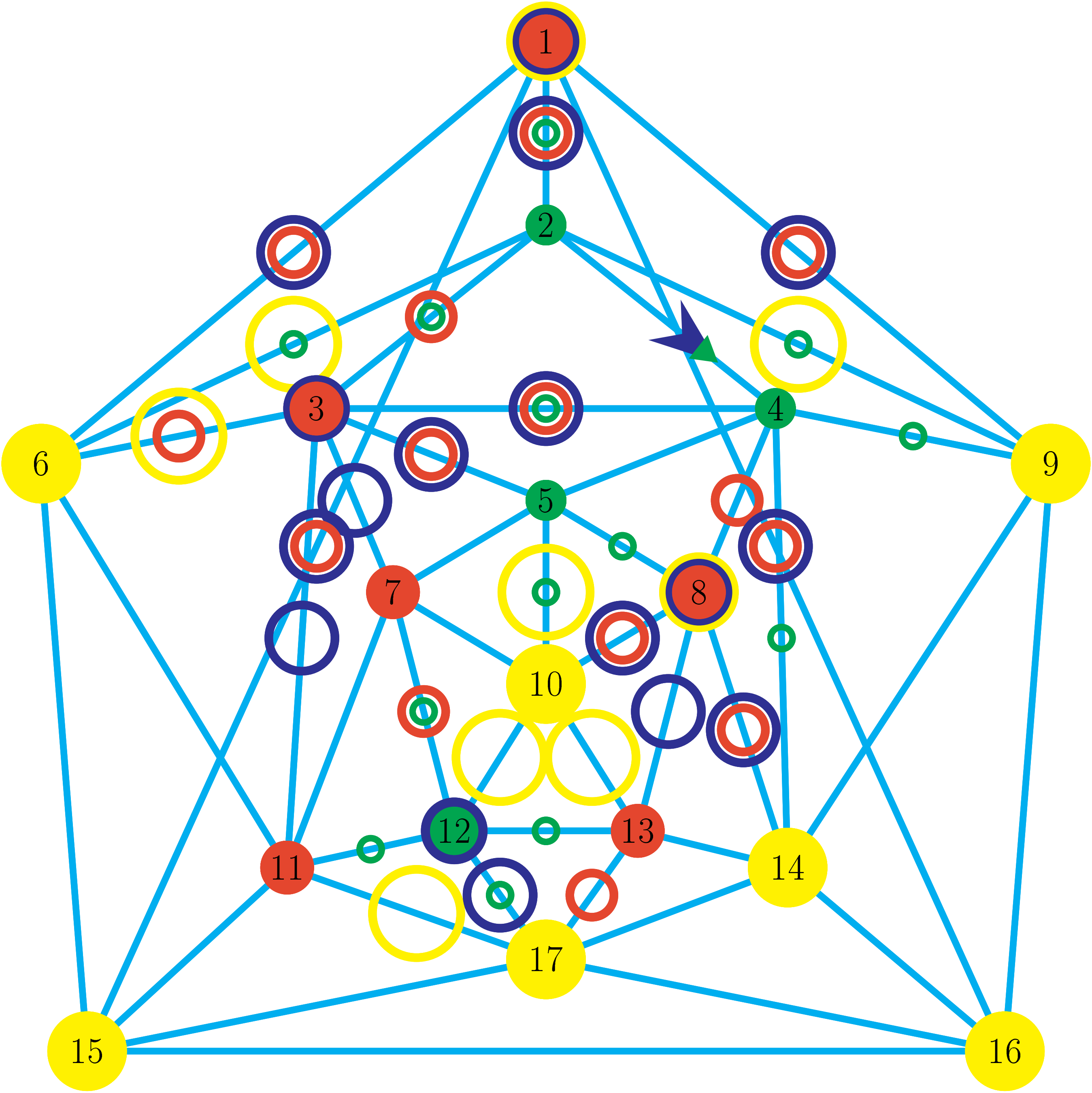


Figure 122.

instruction 249: place edge 2->4 Green DeletionArrow

instruction 250: place edge 2->4 Blue InsertionArrow

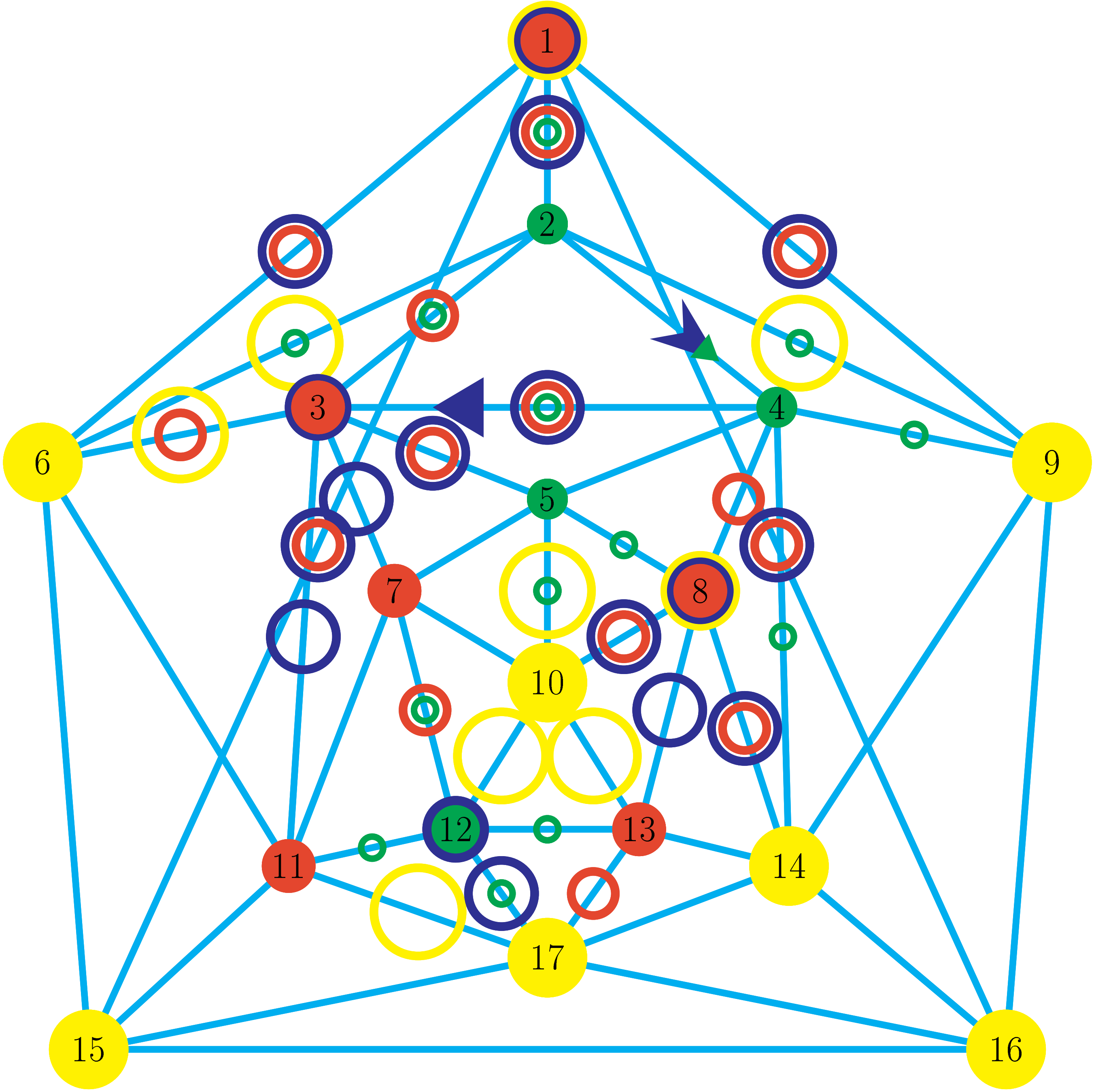


Figure 123.

instruction 251: place edge 4->3 Blue DeletionArrow

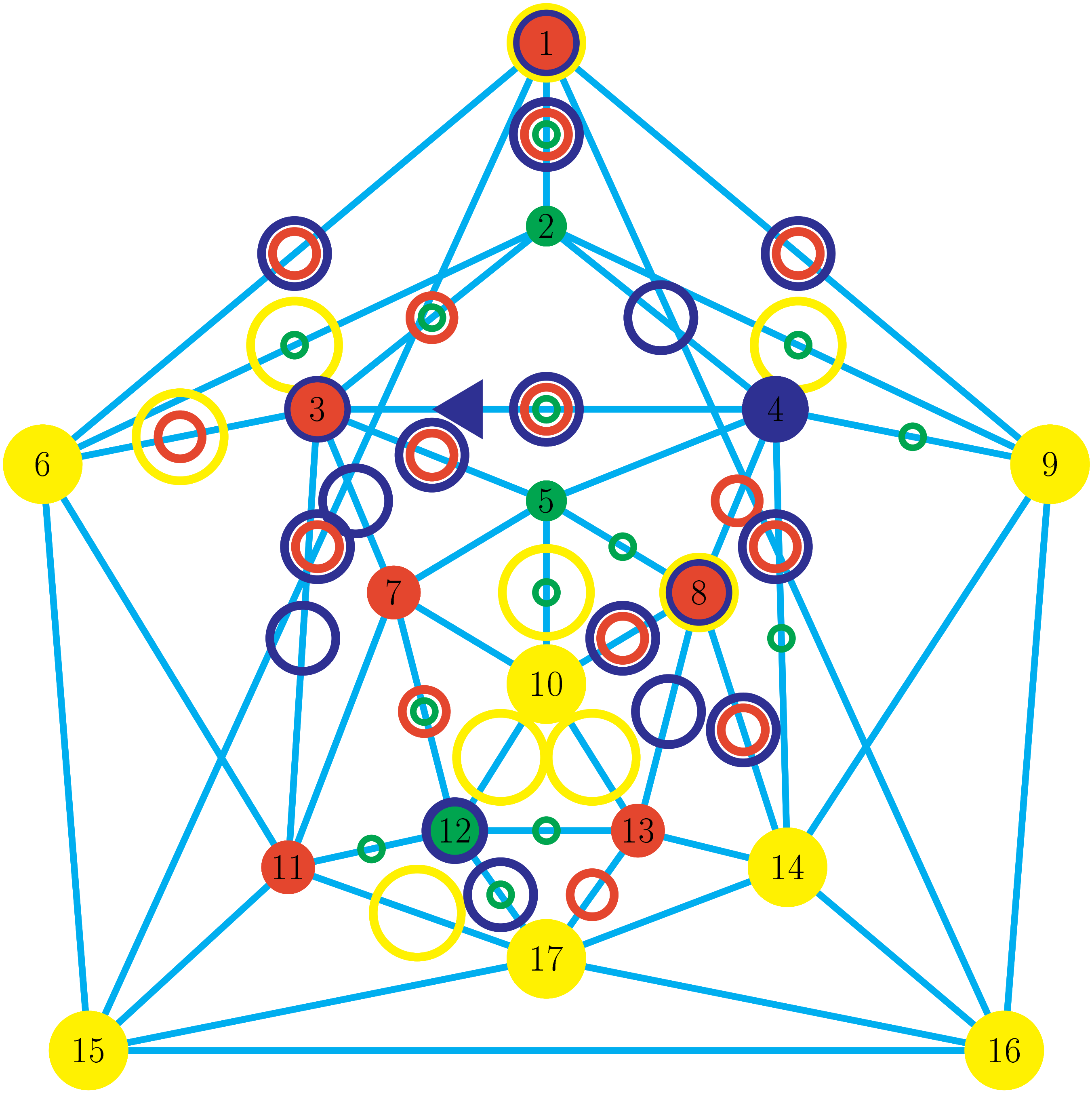


Figure 124.

instruction 252: unplace edge 2->4 Green DeletionArrow
instruction 253: unplace edge 2->4 Blue InsertionArrow
instruction 254: unplace vertex 4 Green Checker;
instruction 255: place vertex 4 Blue Checker;
instruction 256: place edge 2-4 Blue Checker

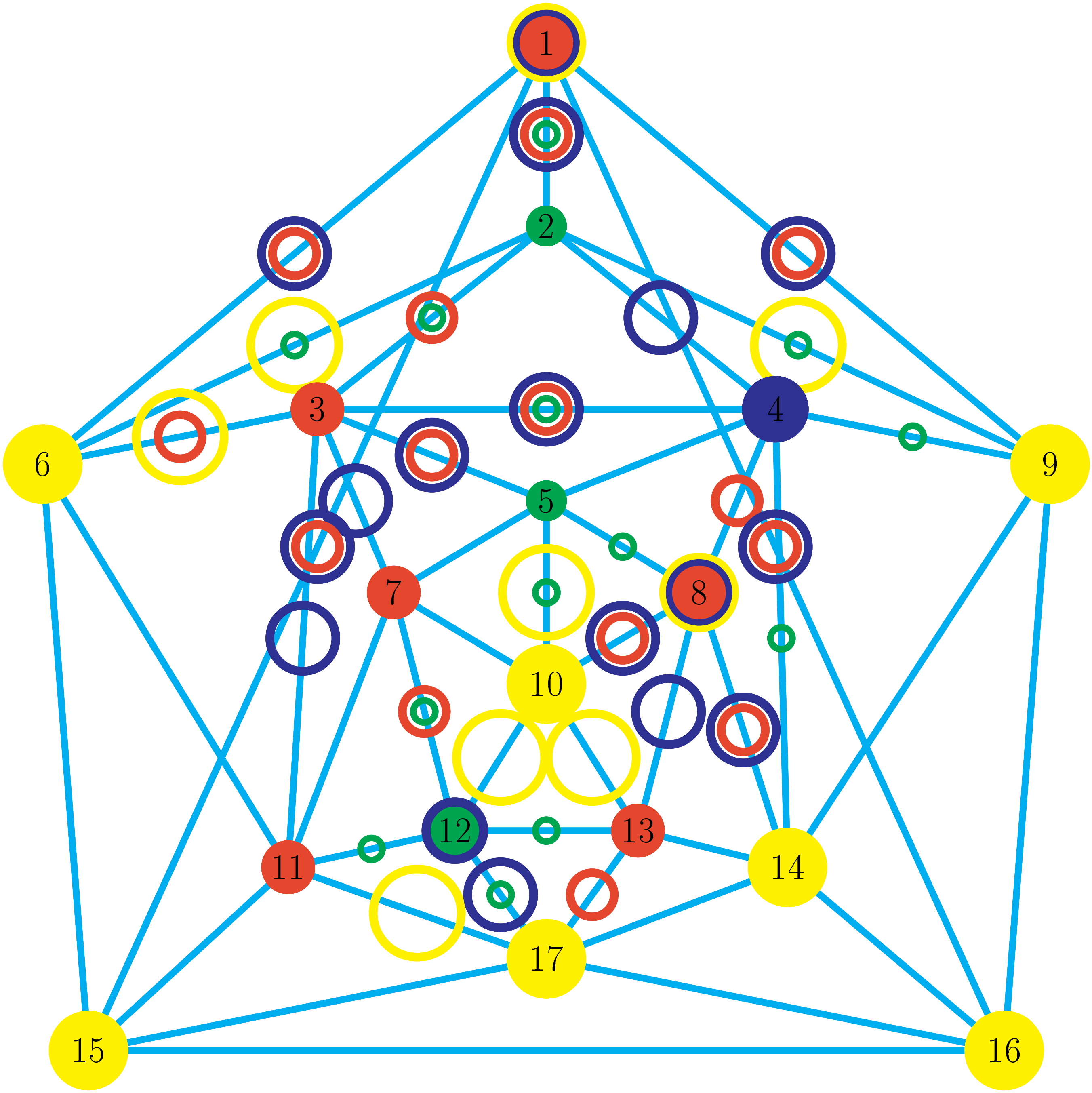


Figure 125.

instruction 257: unplace edge 4->3 Blue DeletionArrow
instruction 258: unplace vertex 3 Blue Checker;

Property B must be restored.

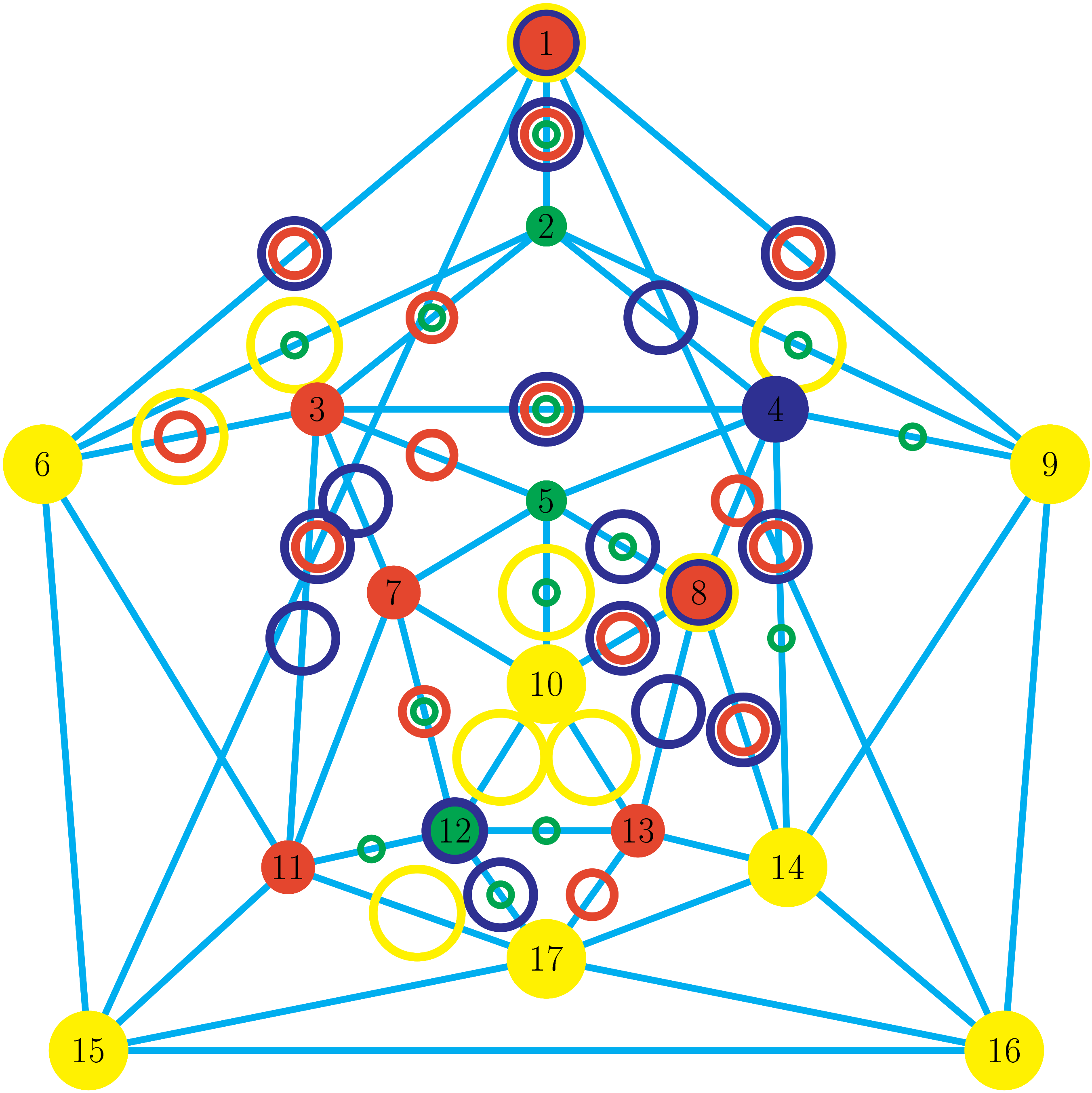


Figure 126.

instruction 259: unplace edge 3-5 Blue Checker
instruction 260: place edge 5-8 Blue Checker

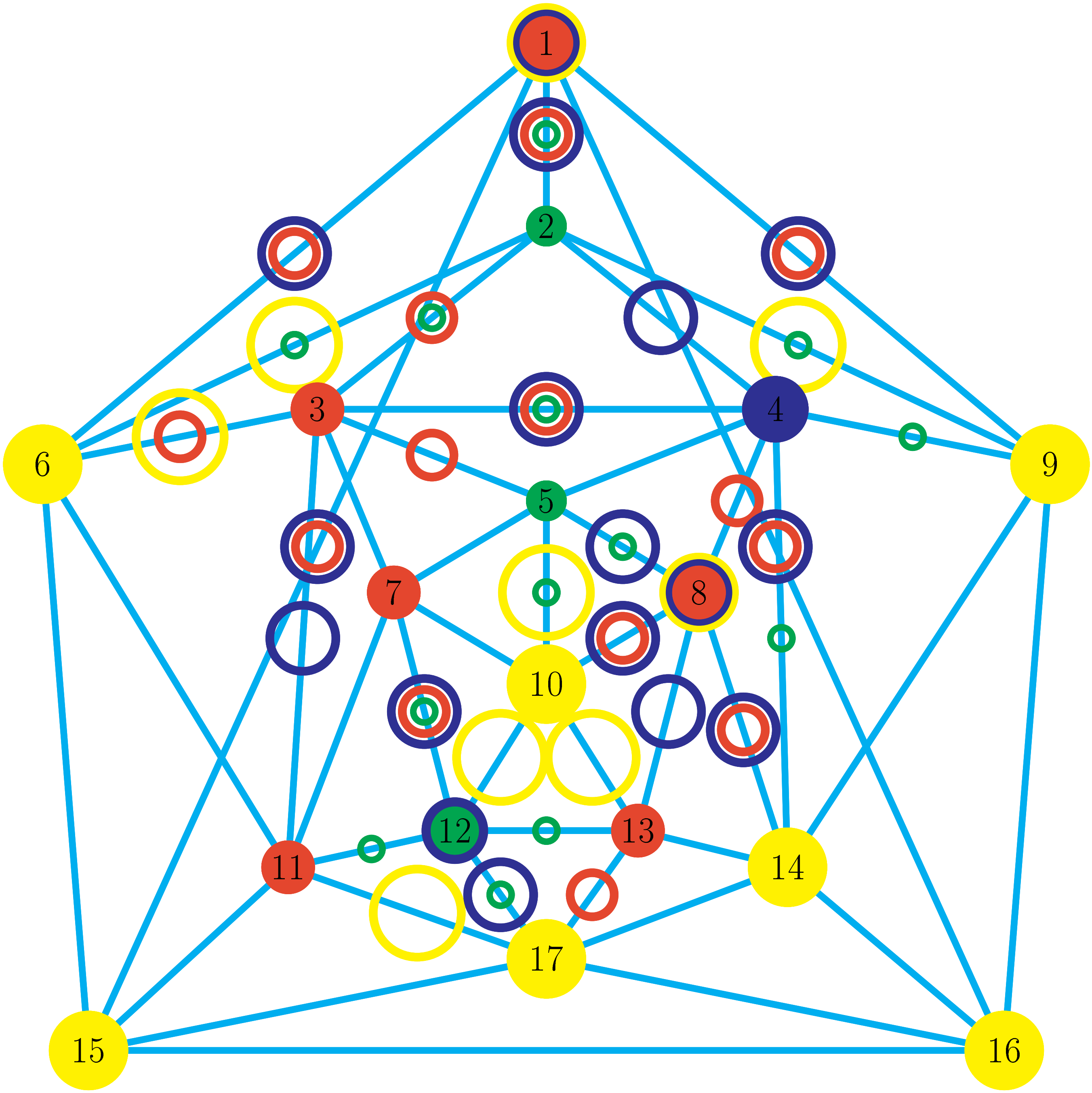


Figure 127.

instruction 261: unplace edge 3-7 Blue Checker

instruction 262: place edge 7-12 Blue Checker

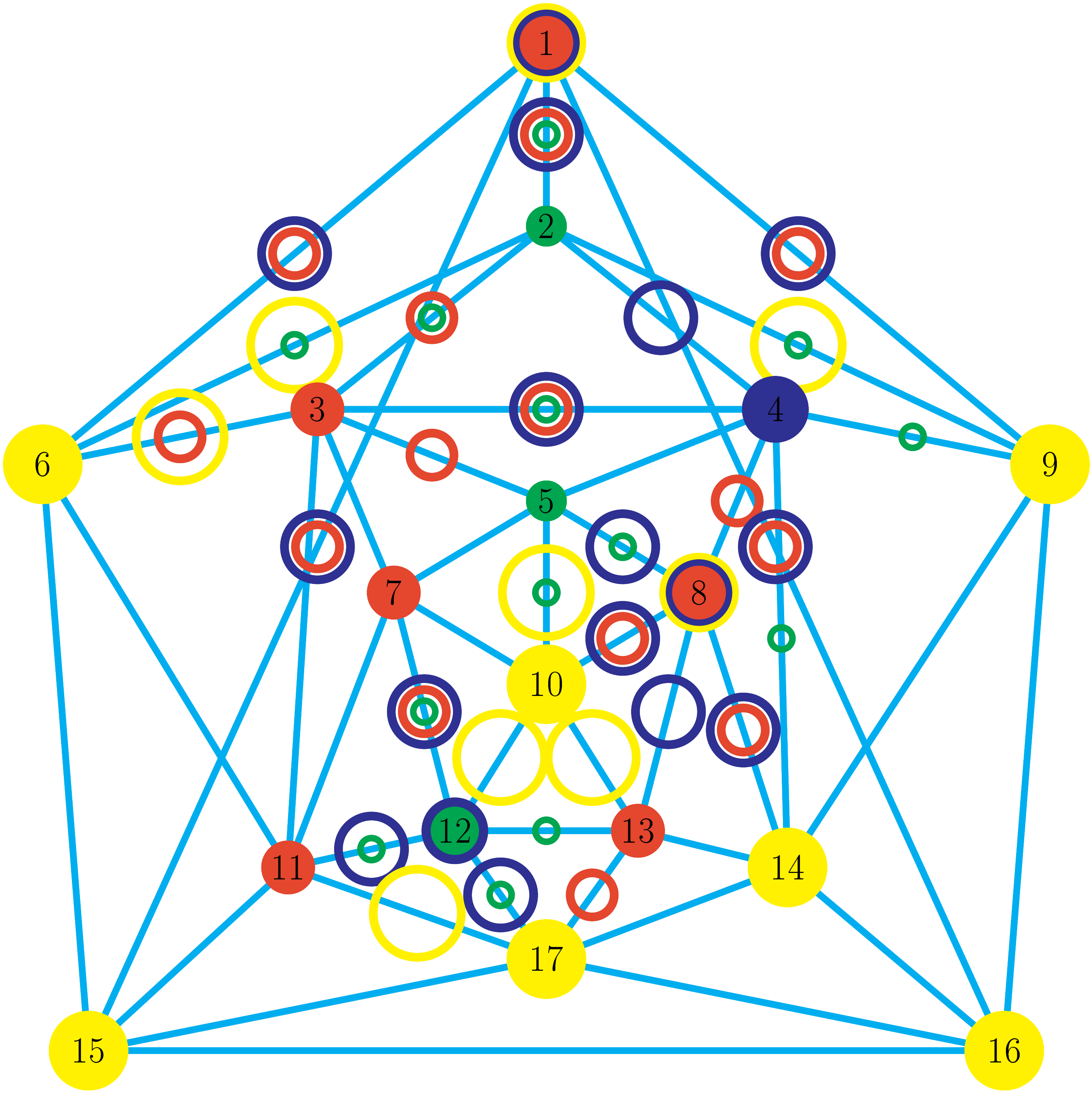


Figure 128.

instruction 263: unplace edge 3-11 Blue Checker
instruction 264: place edge 11-12 Blue Checker

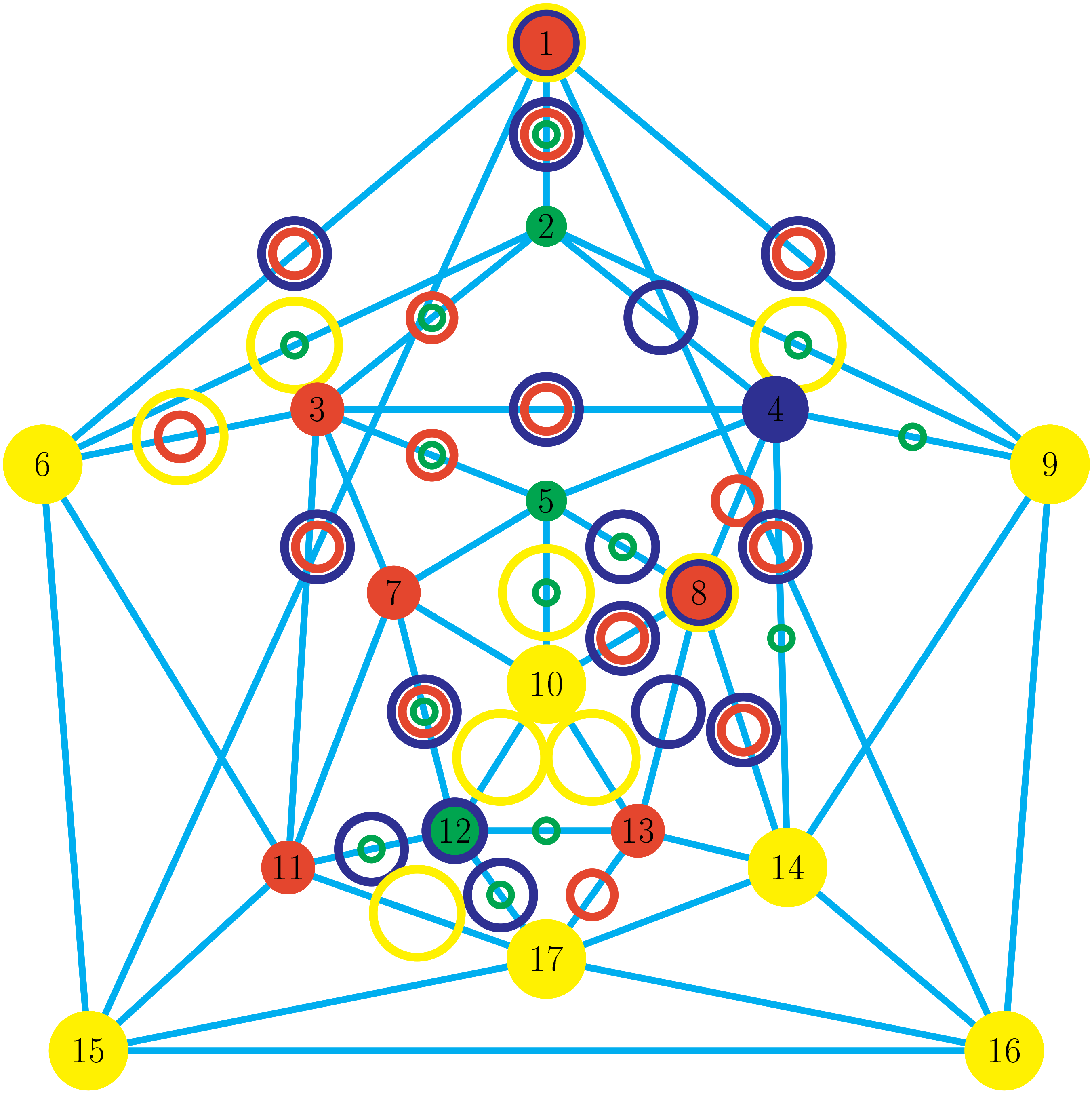


Figure 129.

instruction 265: unplace edge 3-4 Green Checker

instruction 266: place edge 3-5 Green Checker

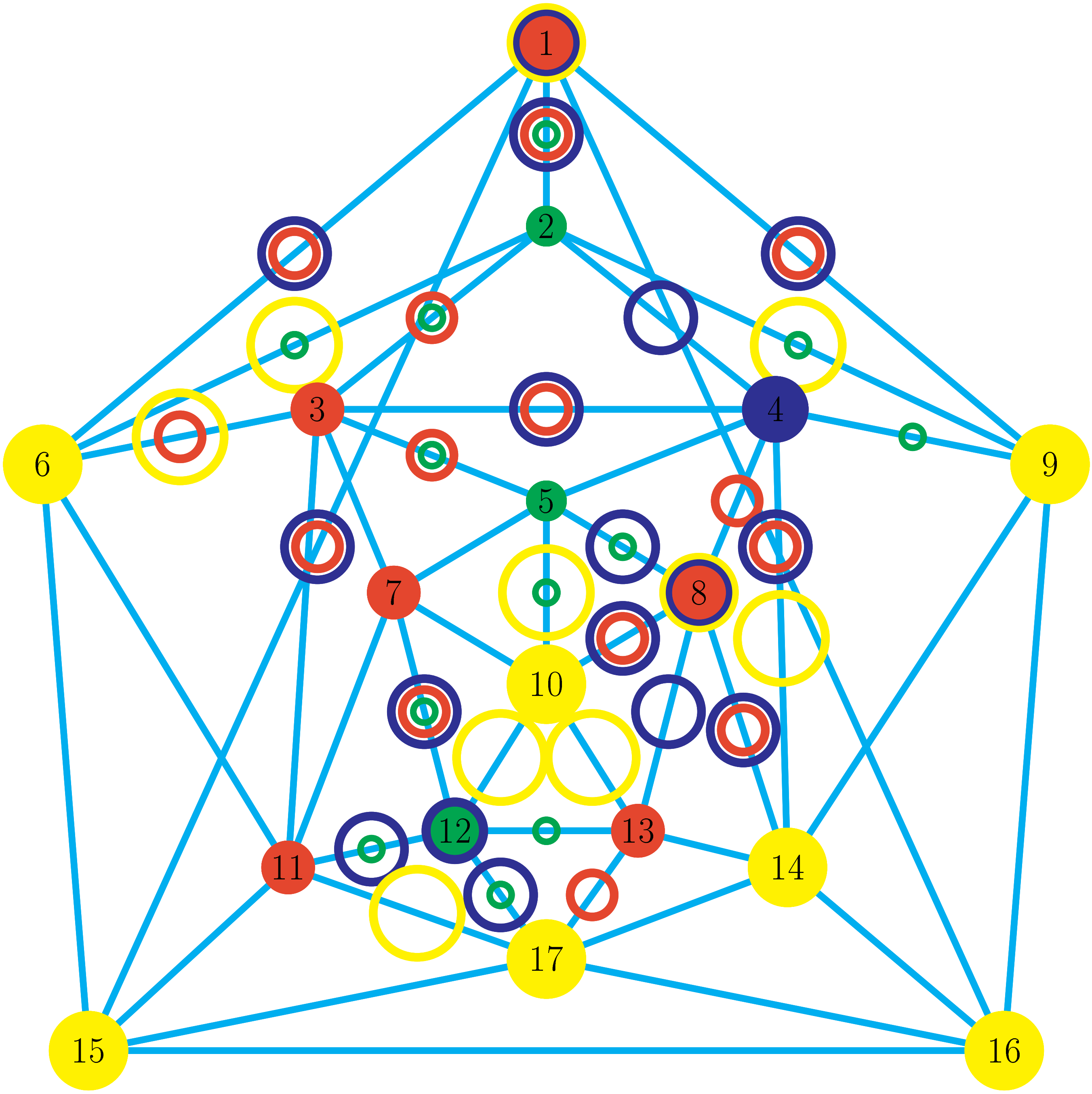


Figure 130.
instruction 267: unplace edge 4-14 Green Checker
instruction 268: place edge 4-14 Yellow Checker

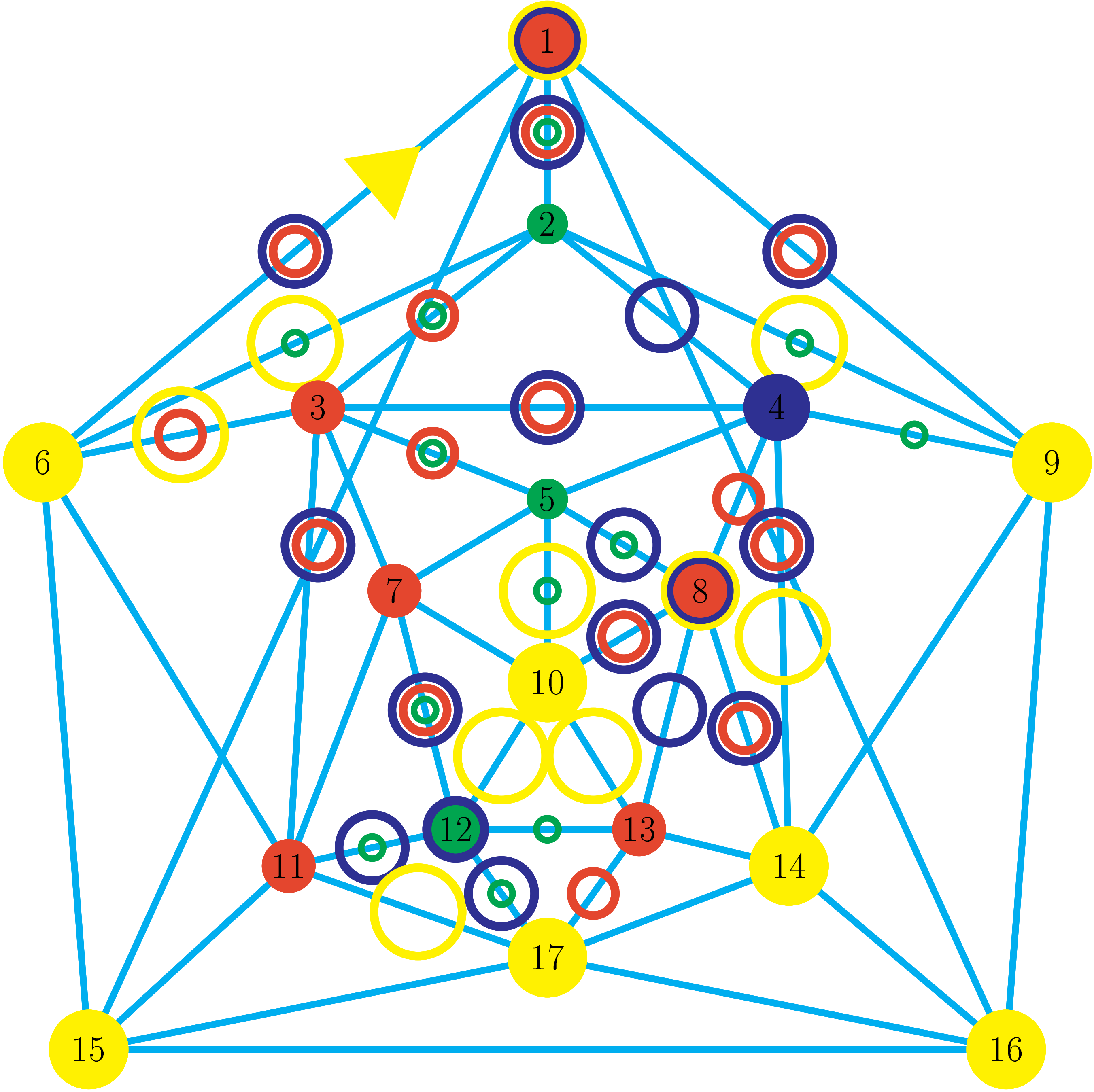


Figure 131.

instruction 269: place edge 6->1 Yellow DeletionArrow

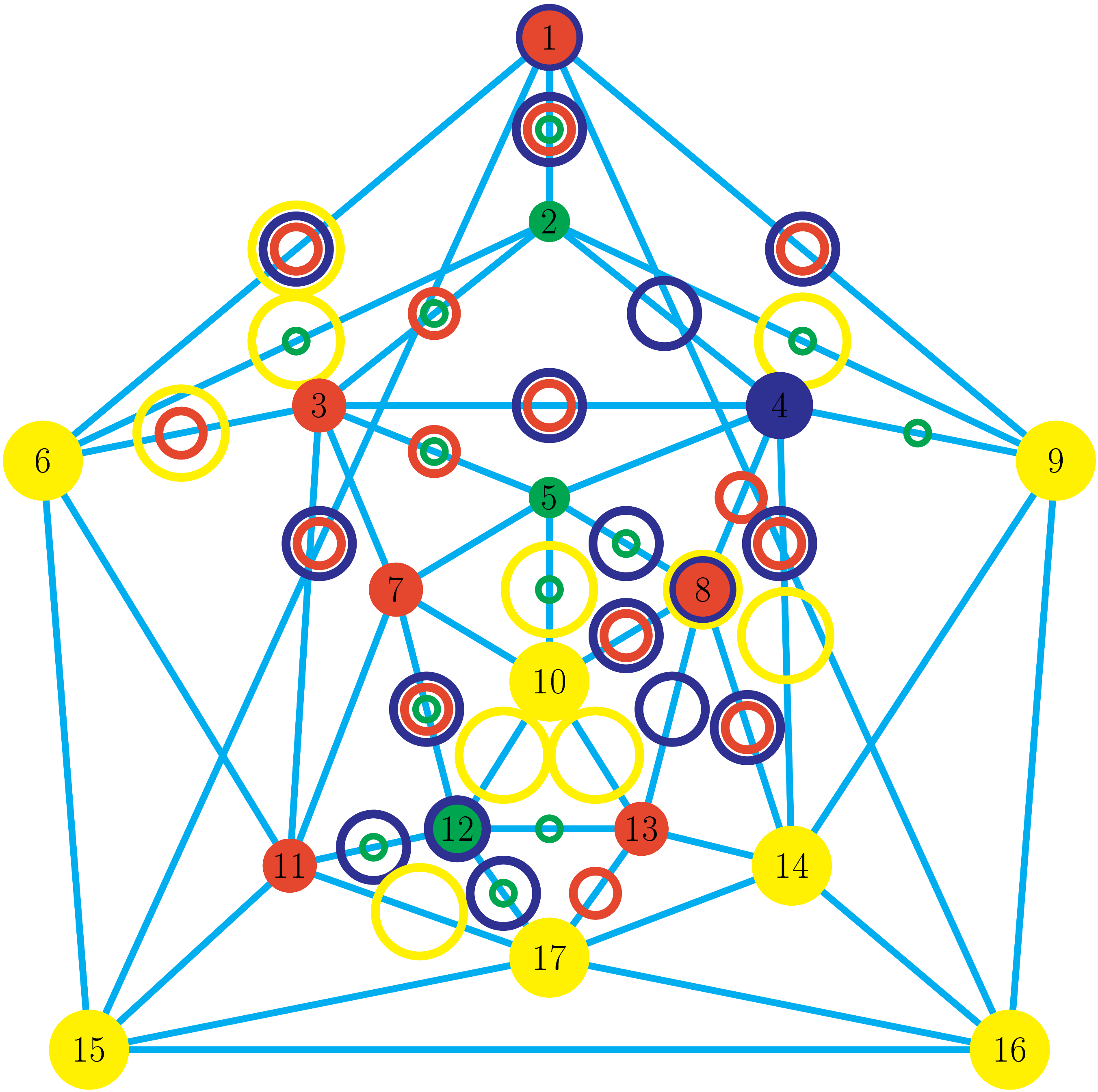


Figure 132.

instruction 270: unplace edge 6->1 Yellow DeletionArrow
instruction 271: unplace vertex 1 Yellow Checker;
instruction 272: place edge 6-1 Yellow Checker

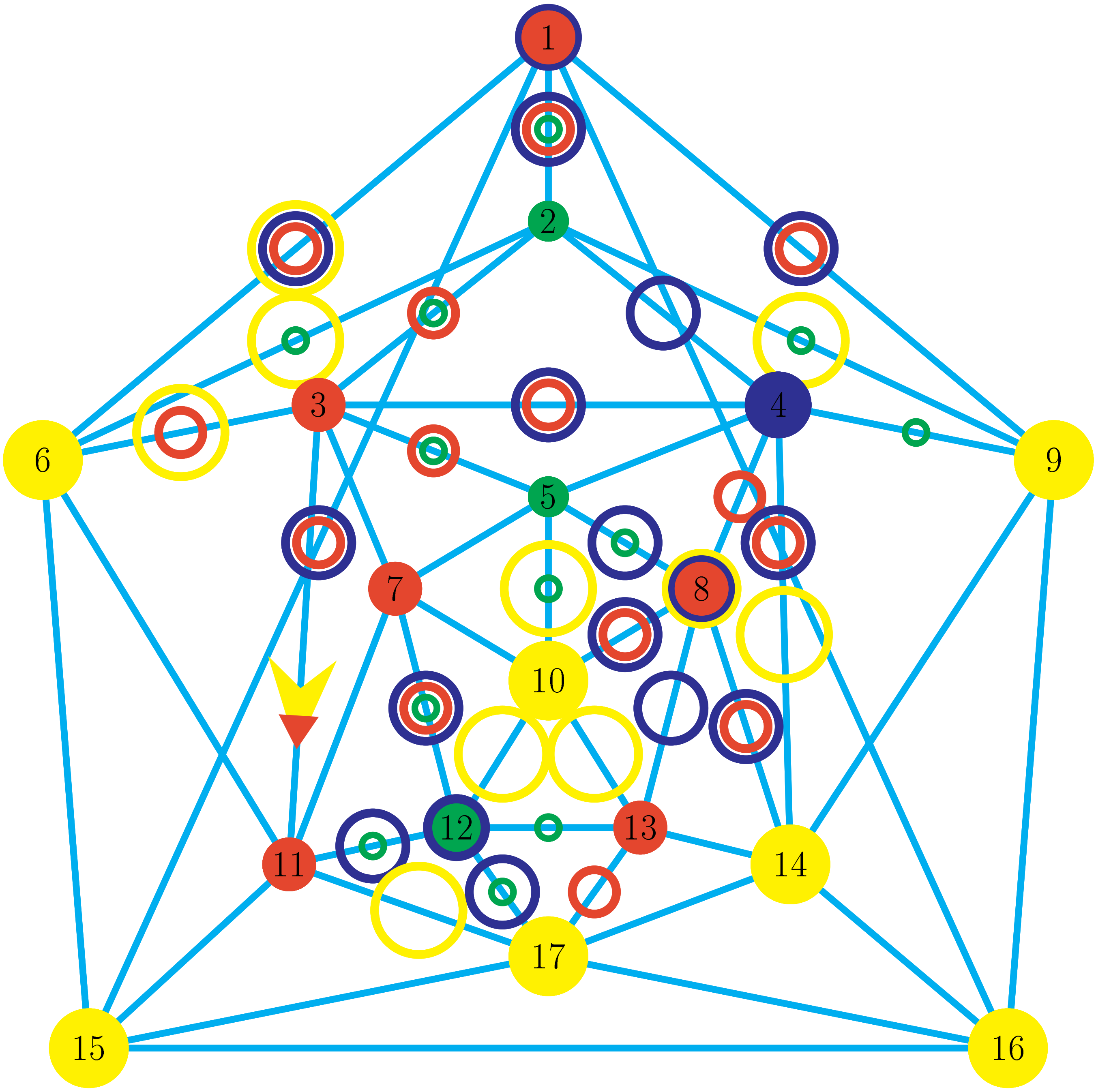


Figure 133.

instruction 273: place edge 3->11 Red DeletionArrow
instruction 274: place edge 3->11 Yellow InsertionArrow

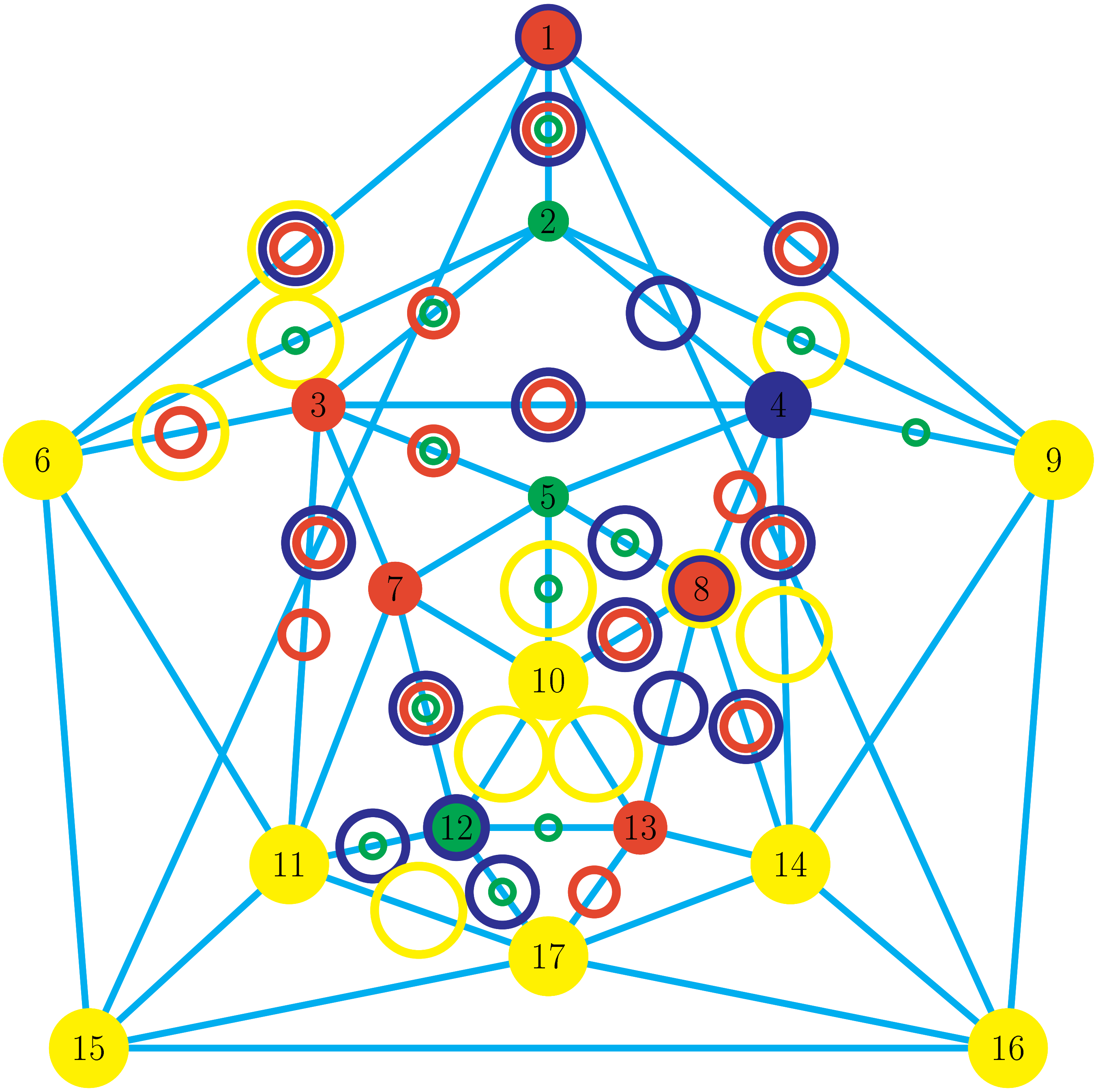


Figure 134.

instruction 275: unplace edge 3->11 Red DeletionArrow
instruction 276: unplace edge 3->11 Yellow InsertionArrow
instruction 277: unplace vertex 11 Red Checker;
instruction 278: place edge 3-11 Red Checker

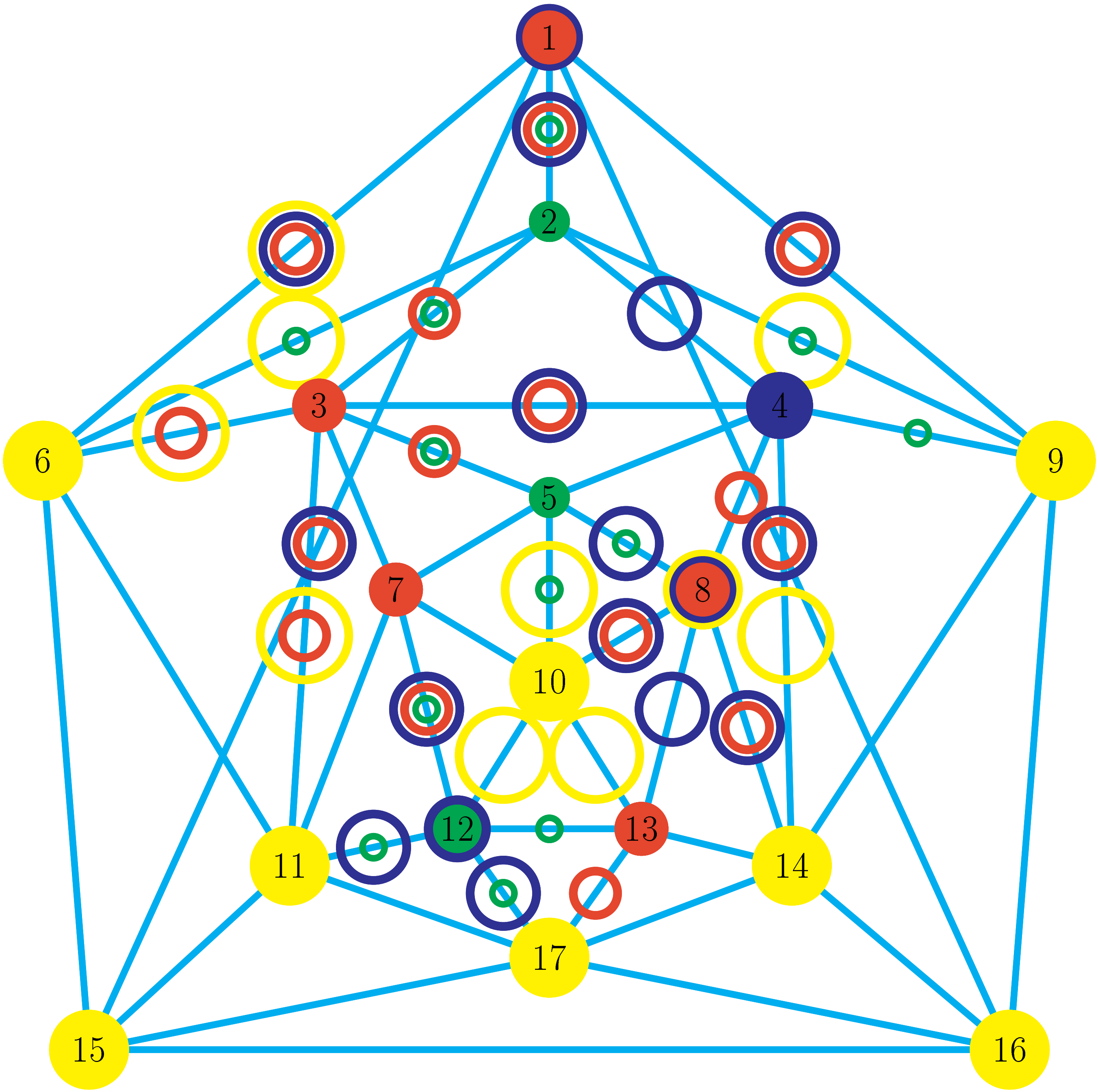


Figure 135.

instruction 279: unplace edge 11-17 Yellow Checker
instruction 280: place edge 11-3 Yellow Checker

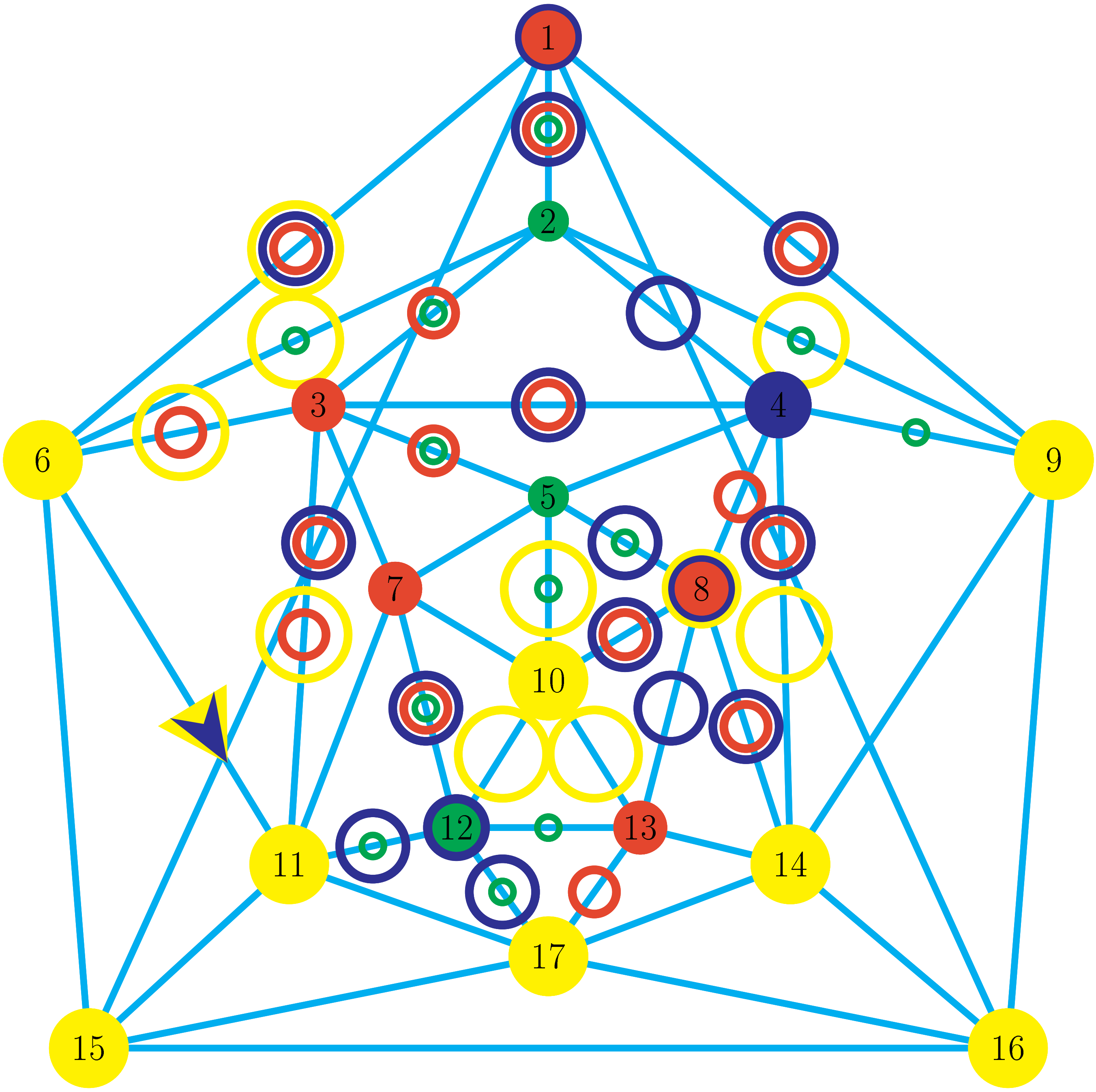


FIGURE 136.

instruction 281: place edge 6->11 Yellow DeletionArrow
instruction 282: place edge 6->11 Blue InsertionArrow

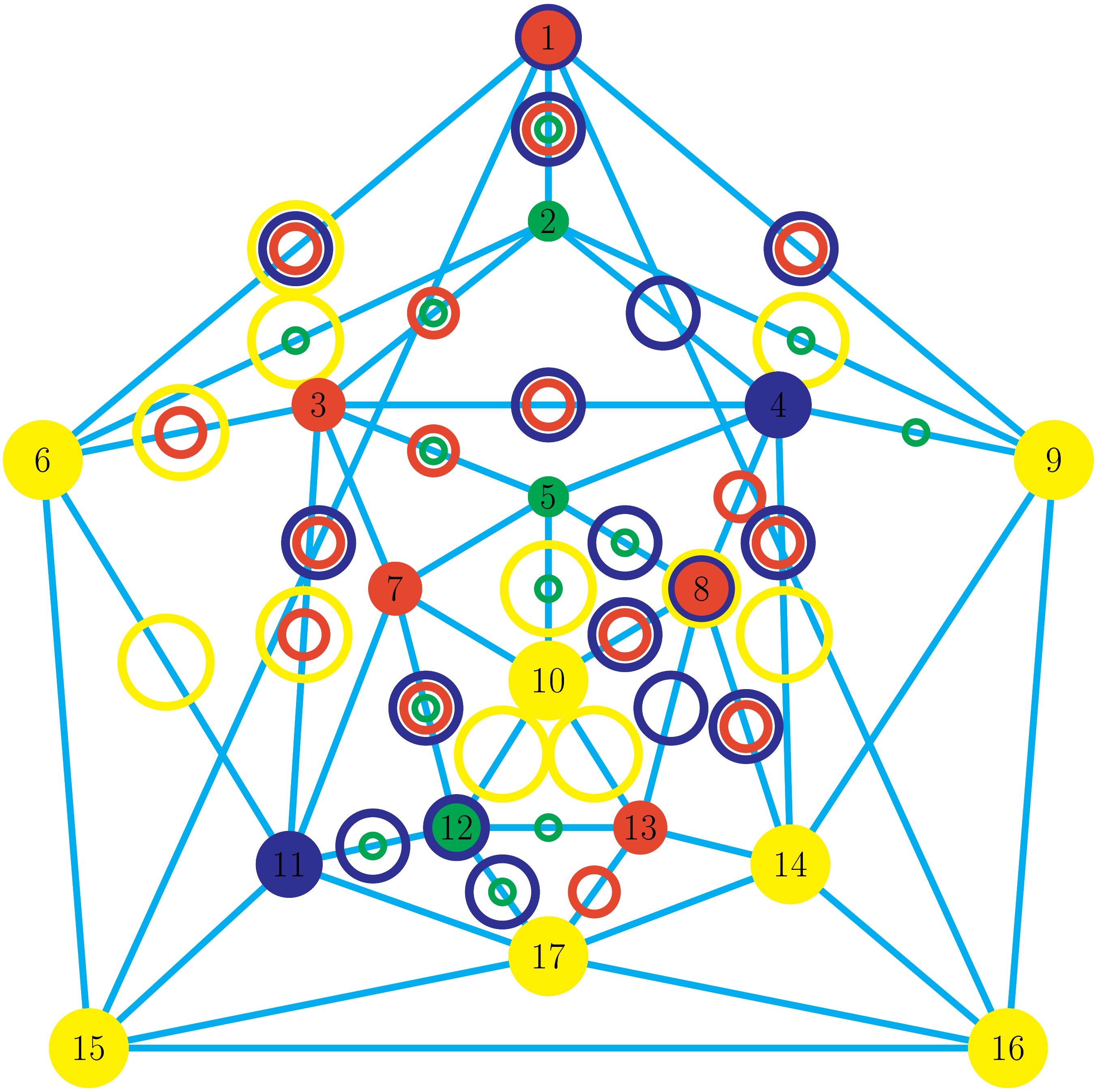


Figure 137.

instruction 283: unplace edge 6->11 Yellow DeletionArrow

instruction 284: unplace edge 6->11 Blue InsertionArrow

instruction 285: unplace vertex 11 Yellow Checker;

instruction 286: place vertex 11 Blue Checker;

instruction 287: place edge 6-11 Yellow Checker

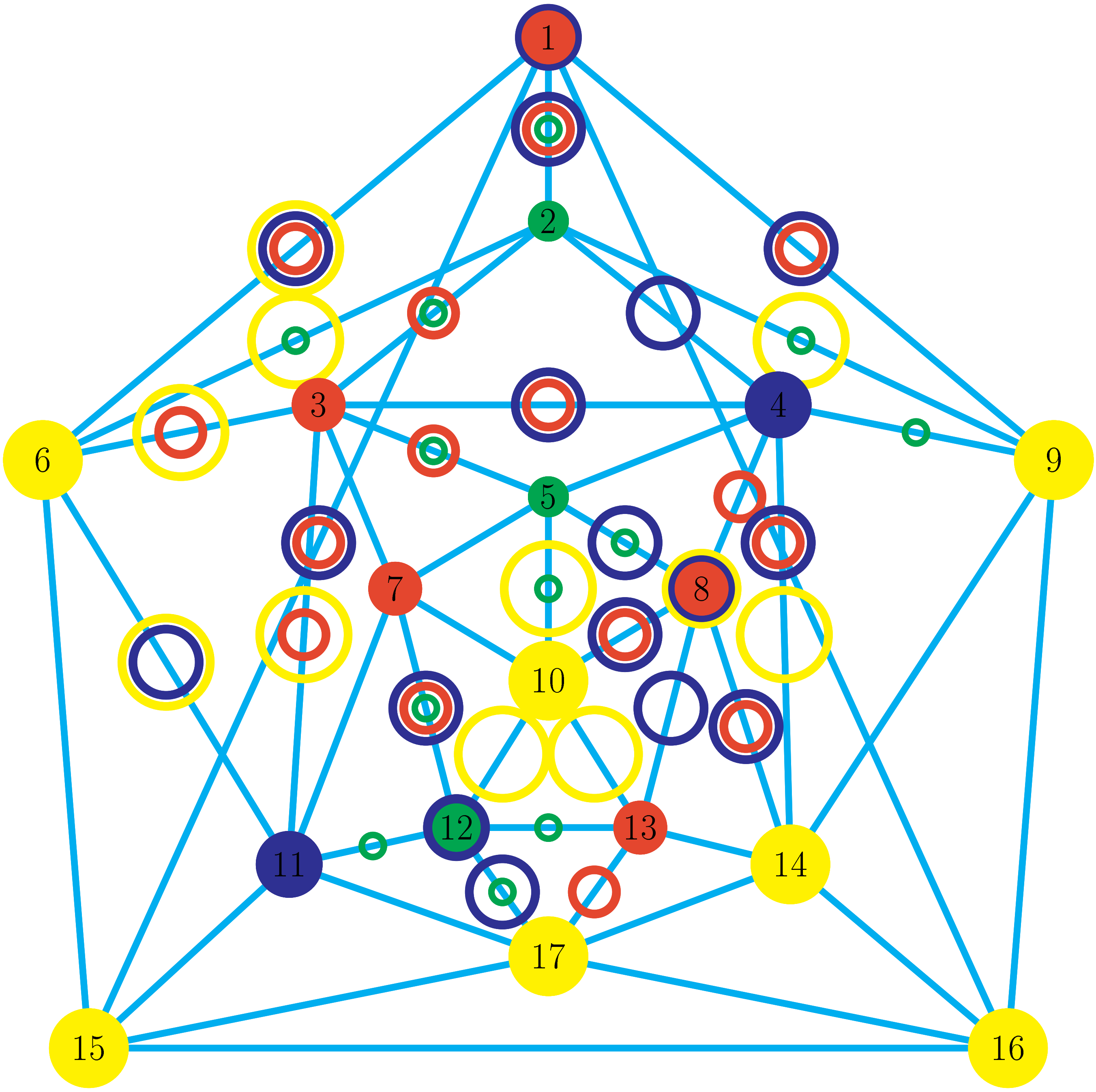


Figure 138.

instruction 288: unplace edge 11-12 Blue Checker

instruction 289: place edge 11-6 Blue Checker

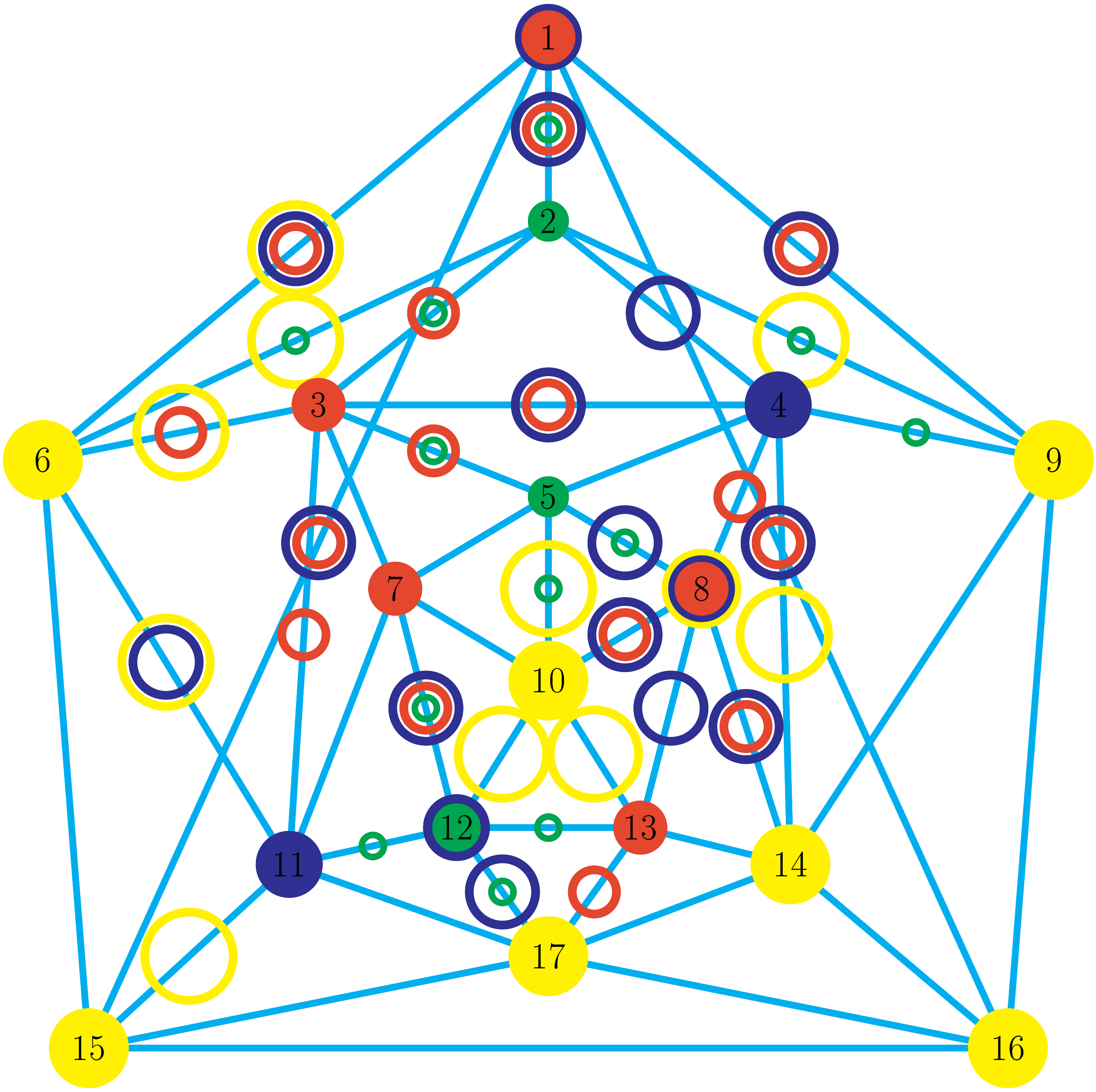


Figure 139.

instruction 290: unplace edge 3-11 Yellow Checker

instruction 291: place edge 11-15 Yellow Checker

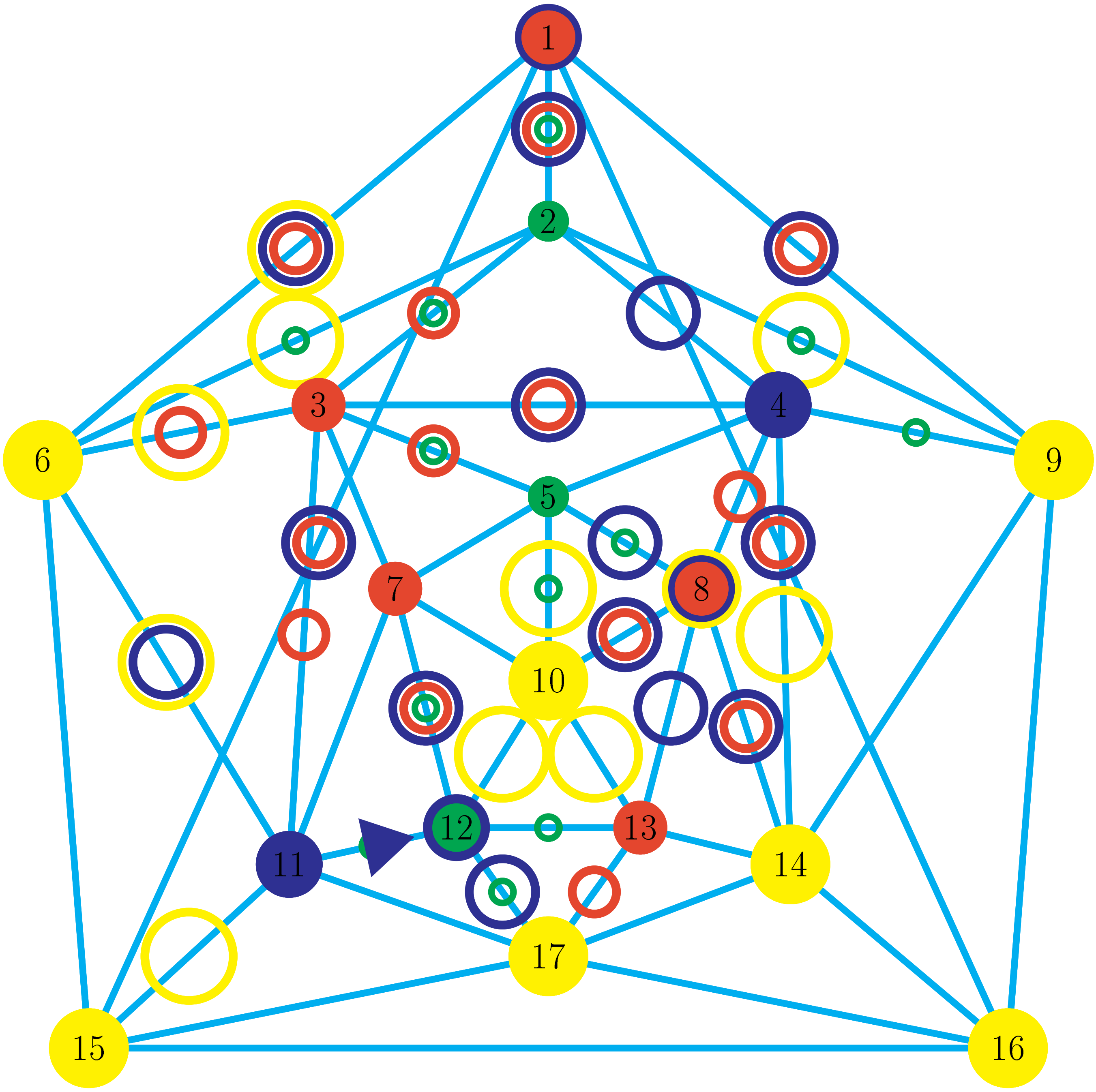


Figure 140.

instruction 292: place edge 11->12 Blue DeletionArrow

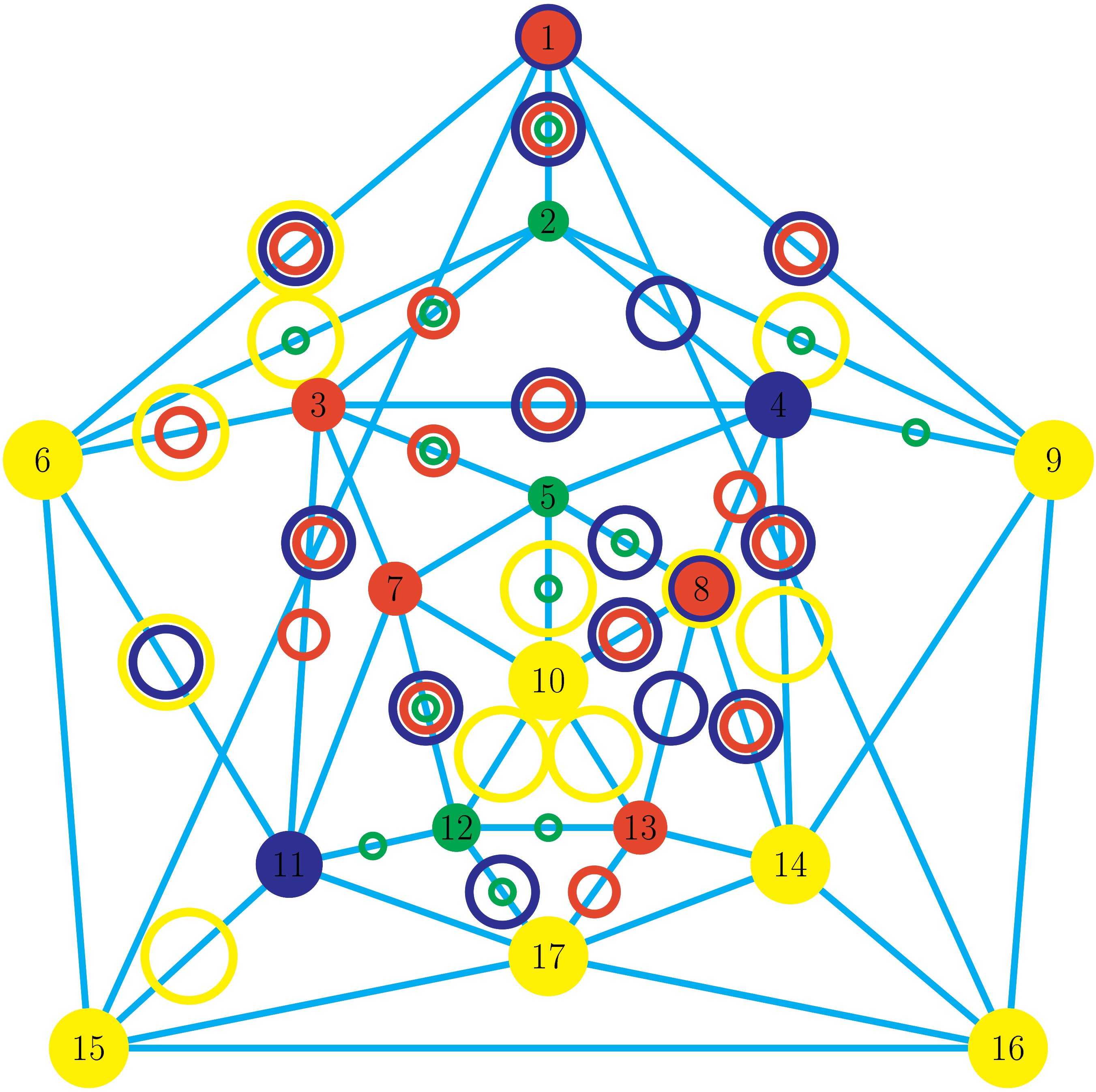


Figure 141.

instruction 293: unplace edge 11->12 Blue DeletionArrow
instruction 294: unplace vertex 12 Blue Checker;

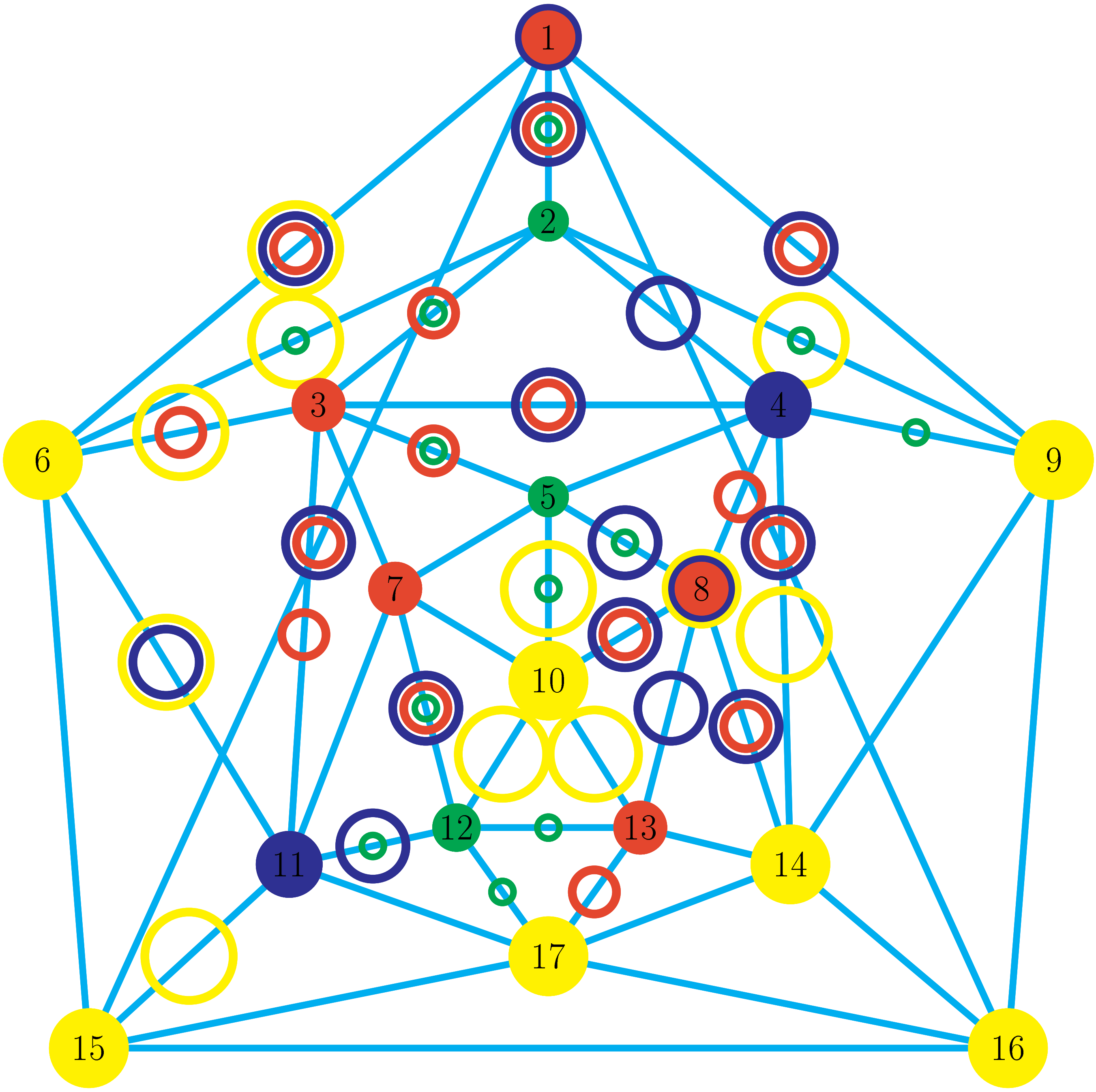


Figure 142.

instruction 295: unplace edge 12-17 Blue Checker

instruction 296: place edge 12-11 Blue Checker

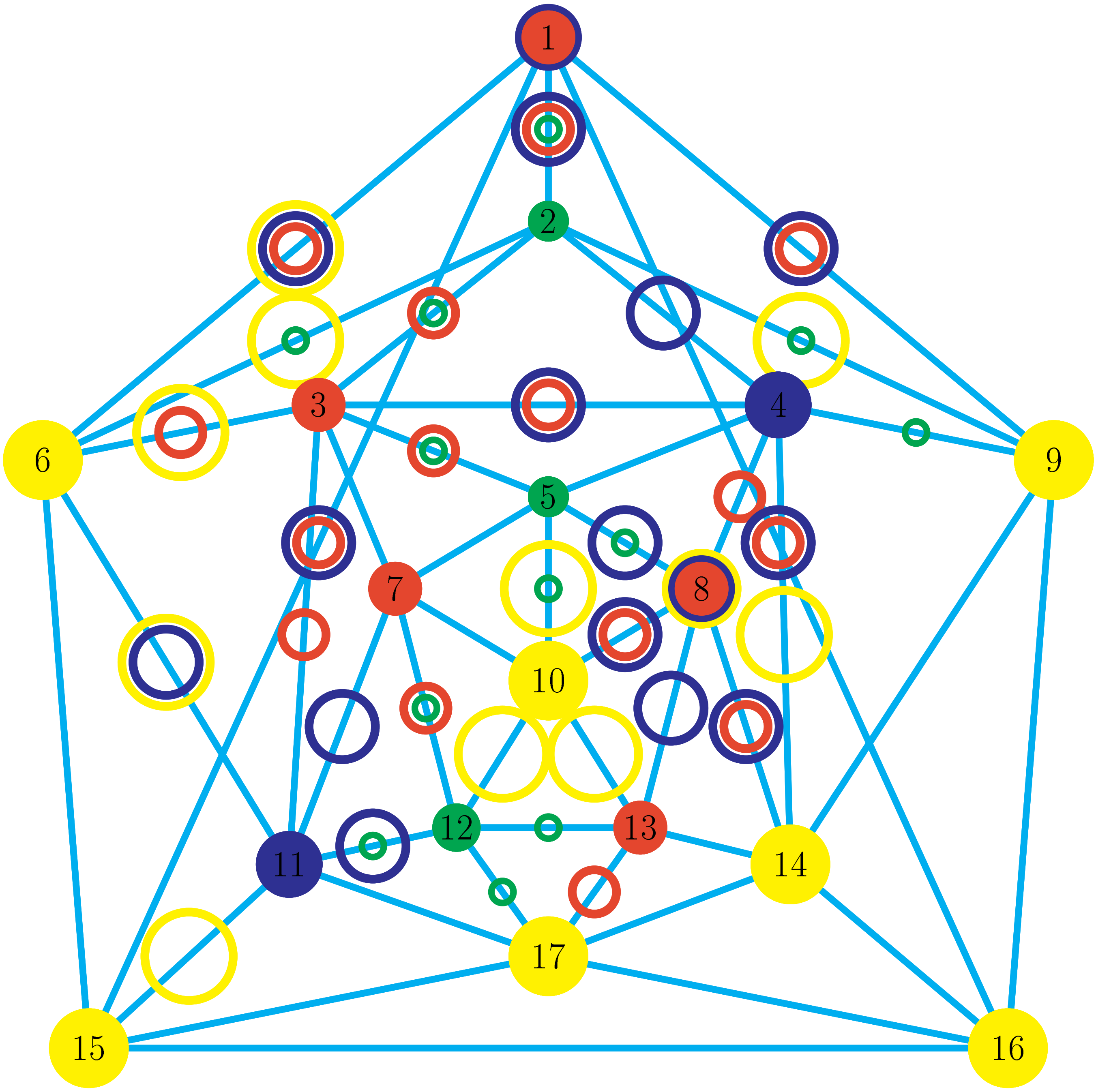


Figure 143.

instruction 297: unplace edge 7-12 Blue Checker

instruction 298: place edge 7-11 Blue Checker

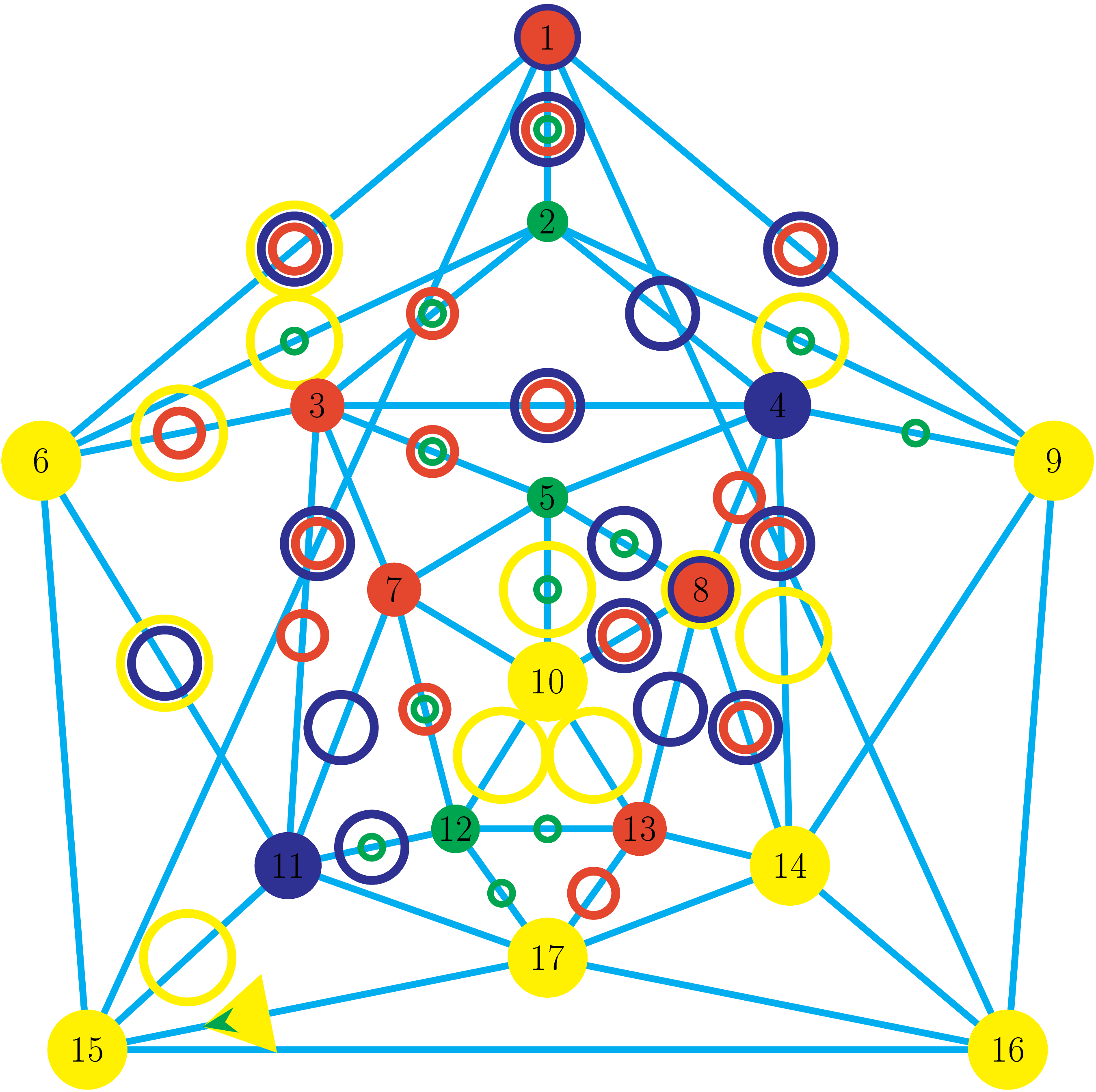


Figure 144.

instruction 299: place edge 17->15 Yellow DeletionArrow

instruction 300: place edge 17->15 Green InsertionArrow

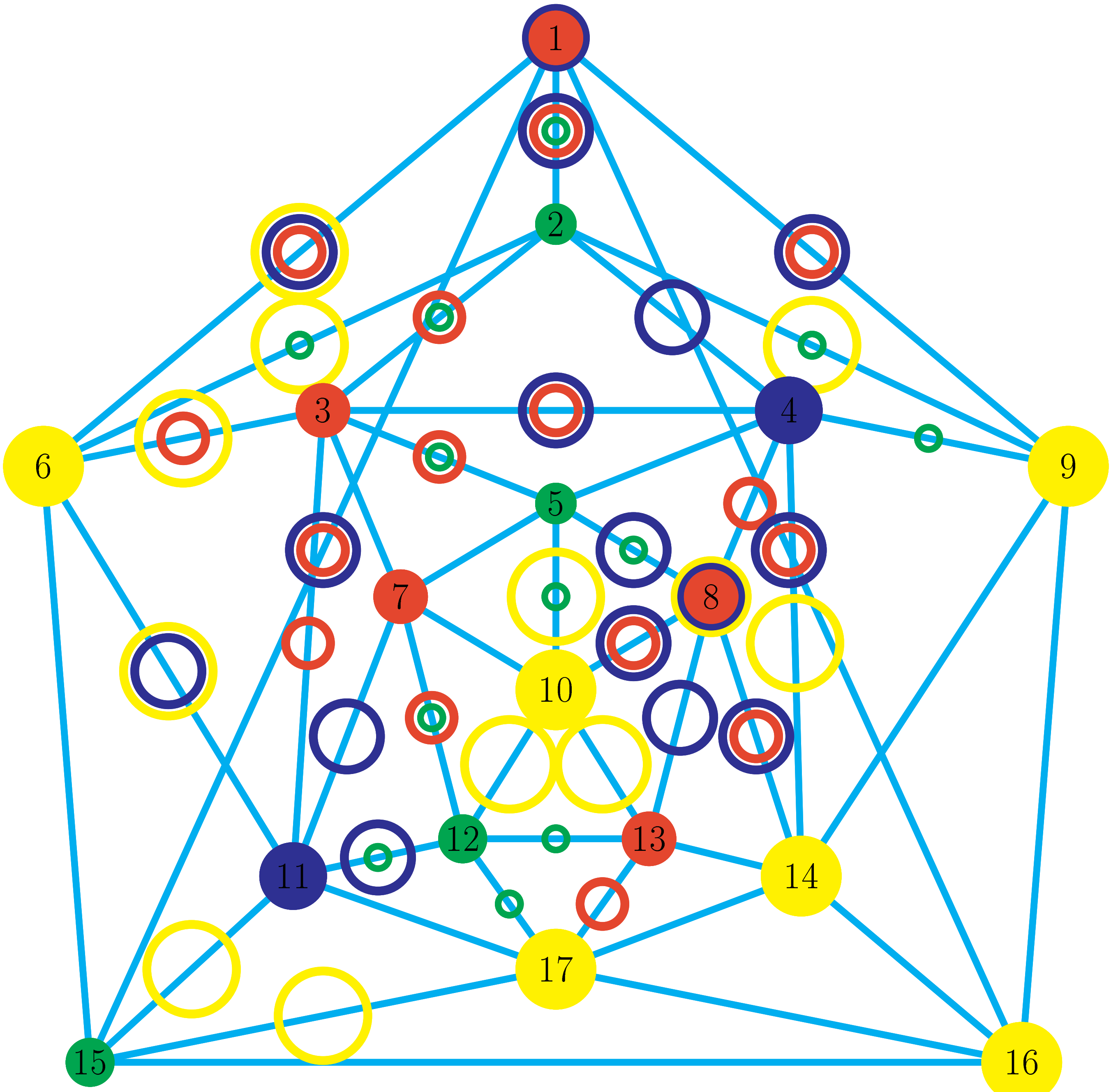


Figure 145.

instruction 301: unplace edge 17->15 Yellow DeletionArrow

instruction 302: unplace edge 17->15 Green InsertionArrow

instruction 303: place edge 17-15 Yellow Checker

instruction 304: unplace vertex 15 Yellow Checker;

instruction 305: place vertex 15 Green Checker;

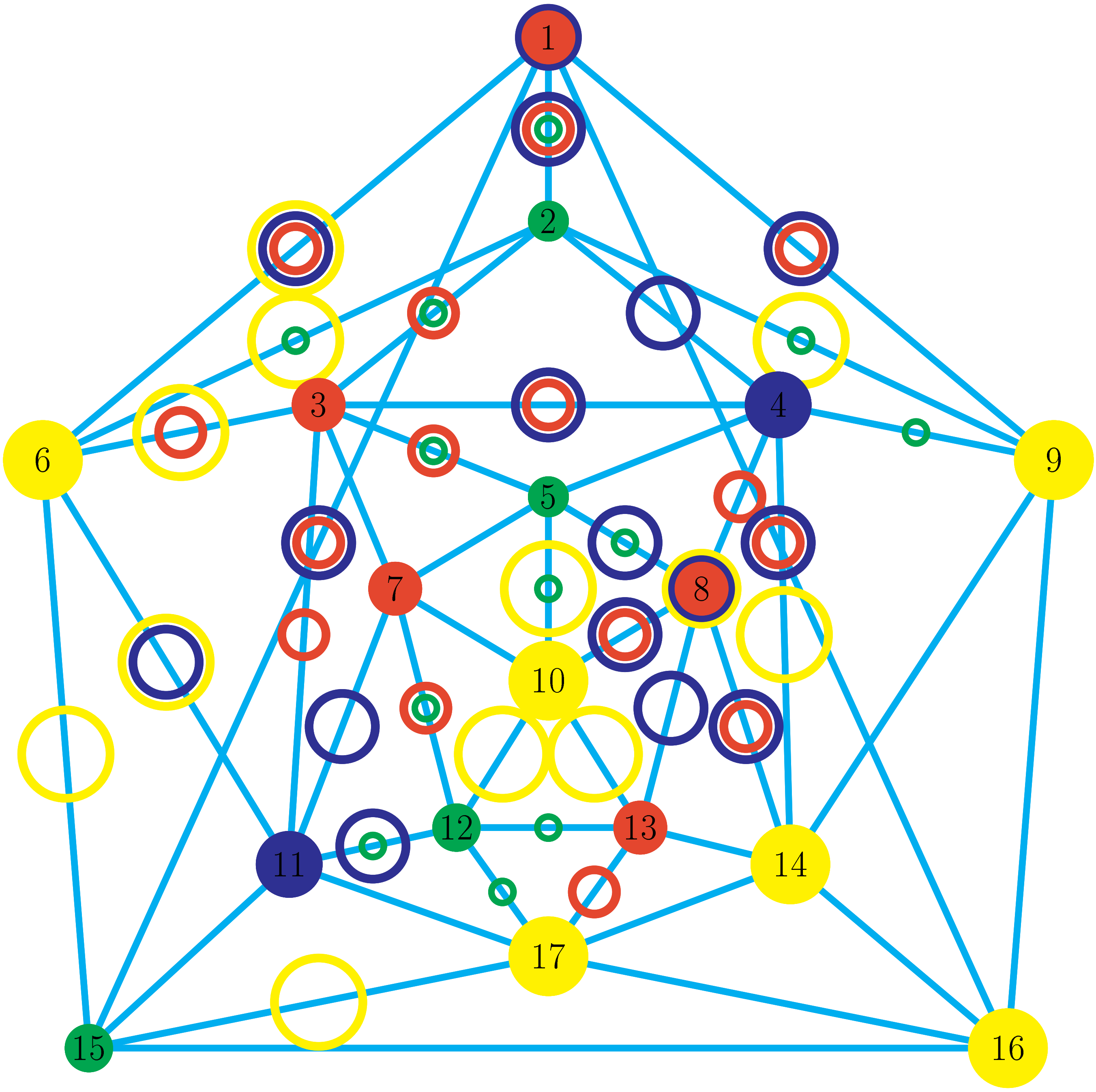


Figure 146.

instruction 306: unplace edge 11-15 Yellow Checker

instruction 307: place edge 15-6 Yellow Checker

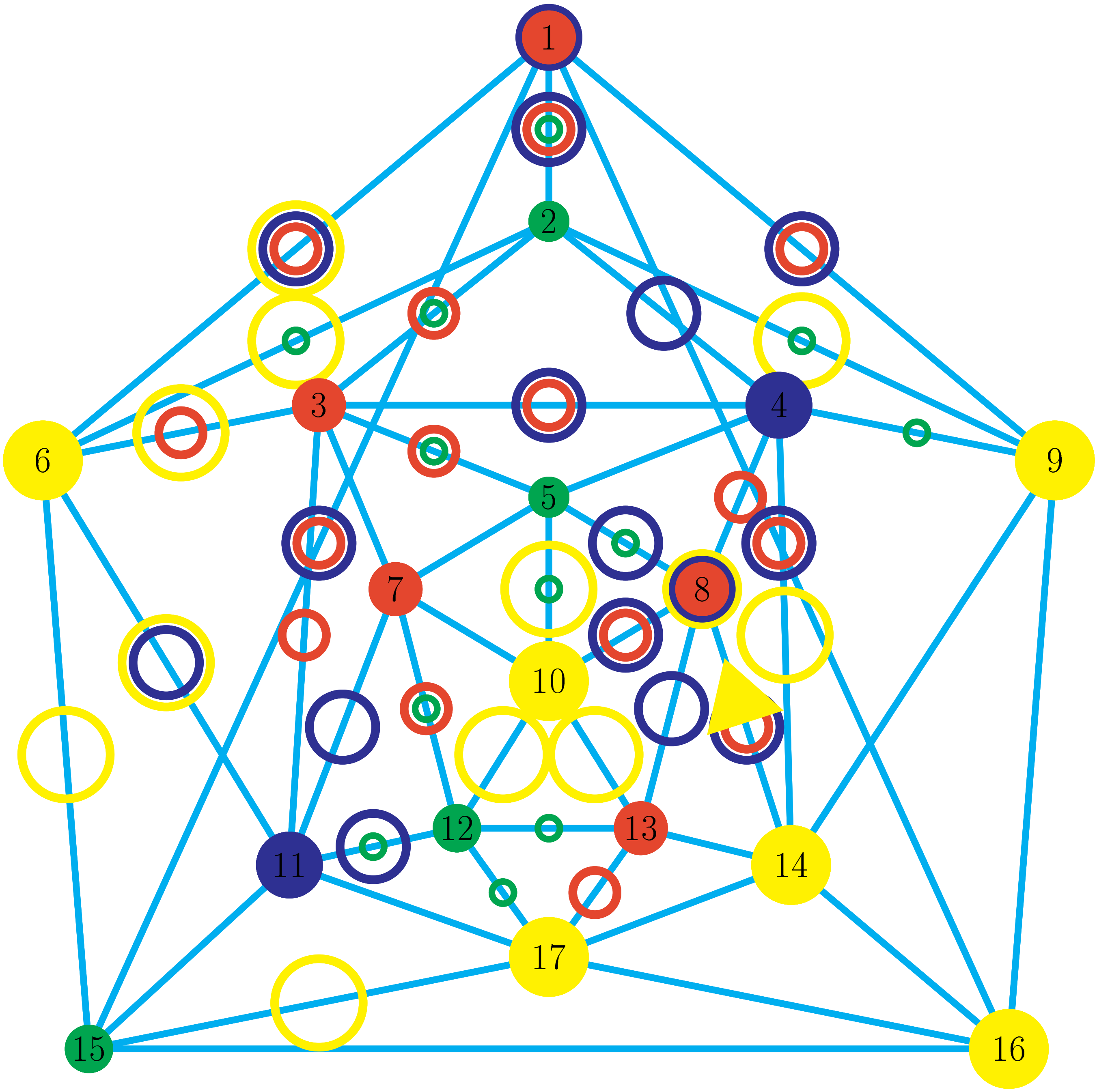


Figure 147.

instruction 308: place edge 14->8 Yellow DeletionArrow

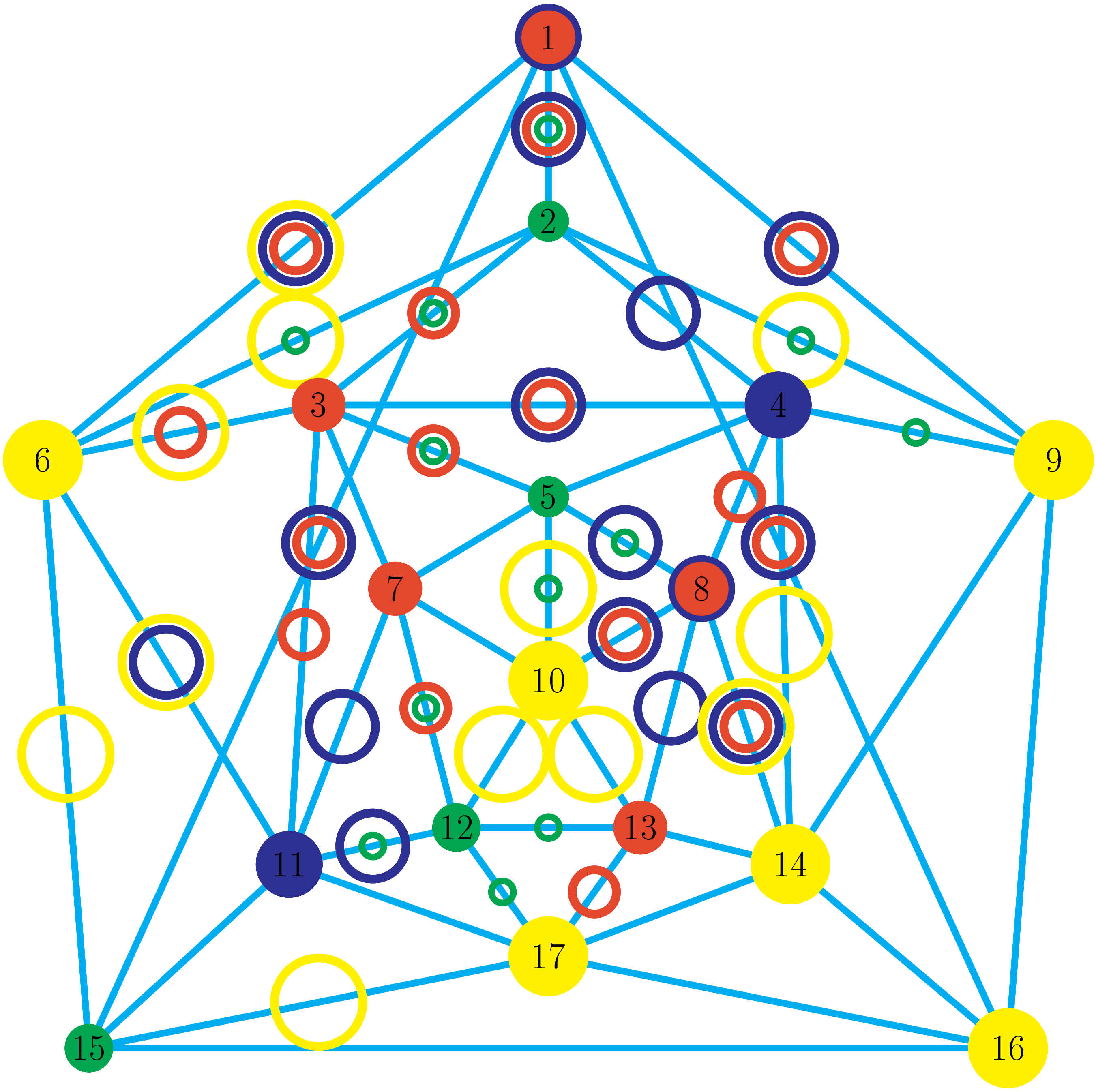


Figure 148.

instruction 309: unplace edge 14->8 Yellow DeletionArrow
instruction 310: place edge 14-8 Yellow Checker
instruction 311: unplace vertex 8 Yellow Checker;

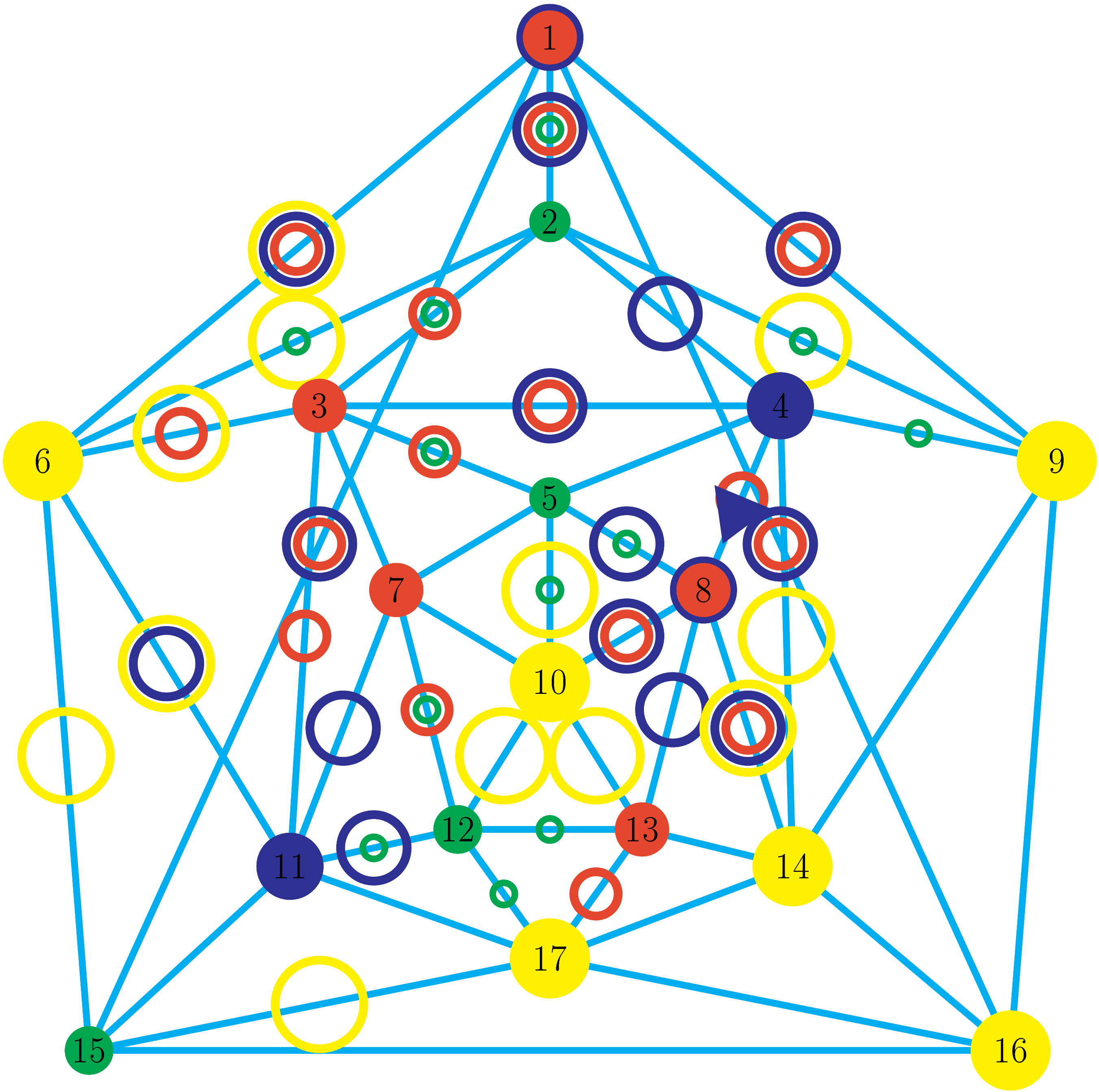


Figure 149.

instruction 312: place edge 4->8 Blue DeletionArrow

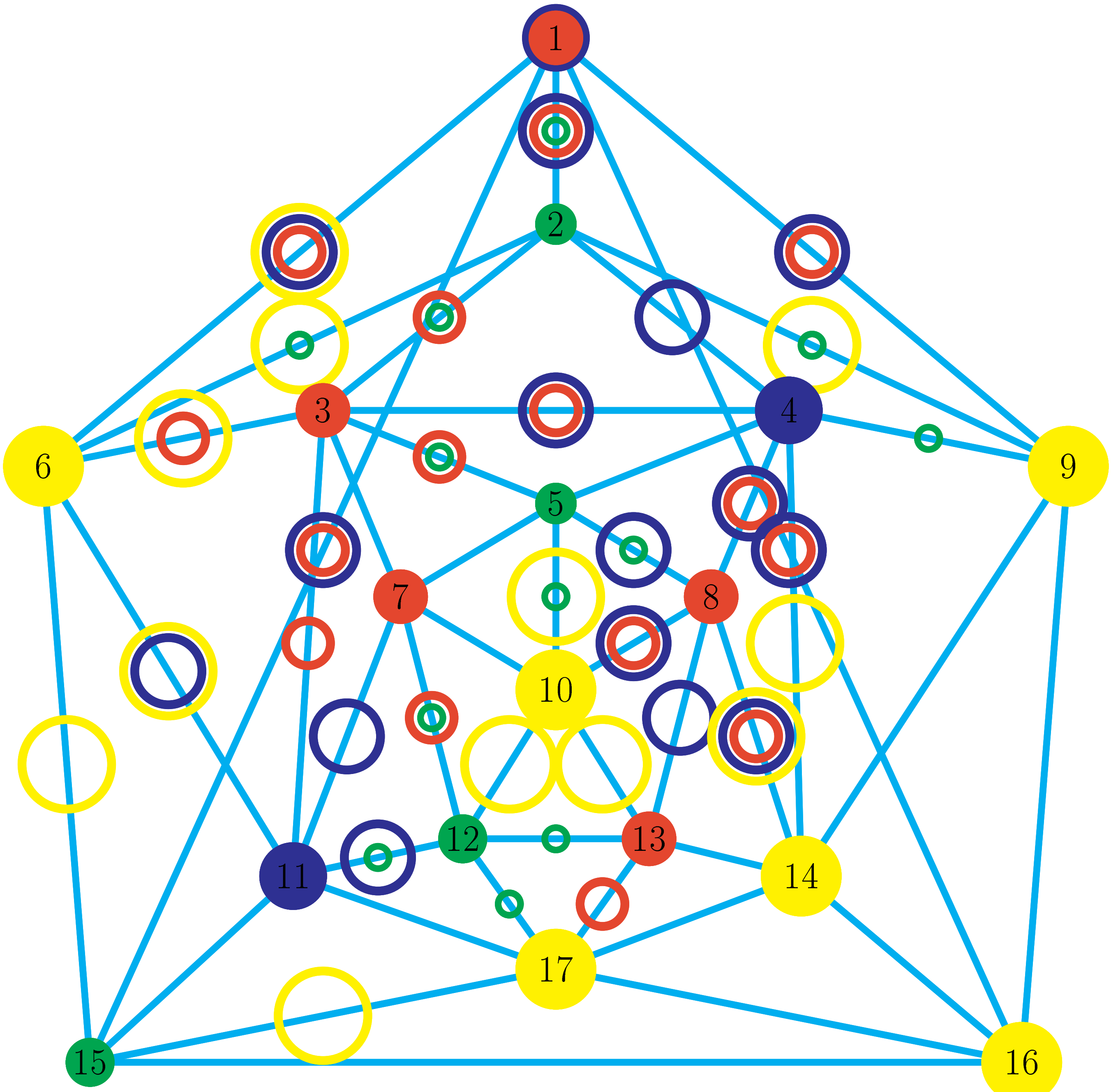


Figure 150.

instruction 313: unplace edge 4->8 Blue DeletionArrow
instruction 314: unplace vertex 8 Blue Checker;
instruction 315: place edge 4-8 Blue Checker

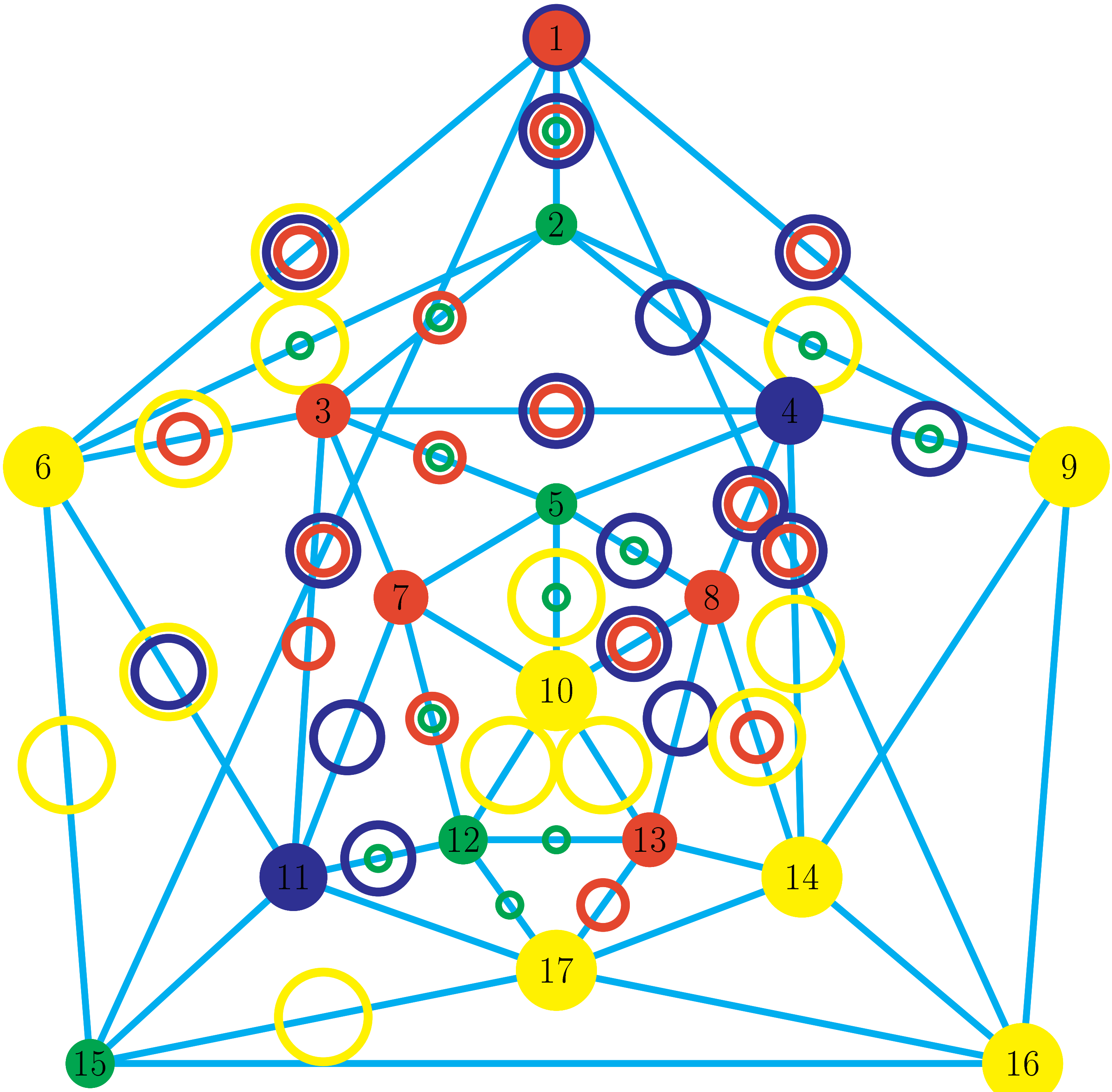


Figure 151.

instruction 316: unplace edge 8-14 Blue Checker

instruction 317: place edge 4-9 Blue Checker

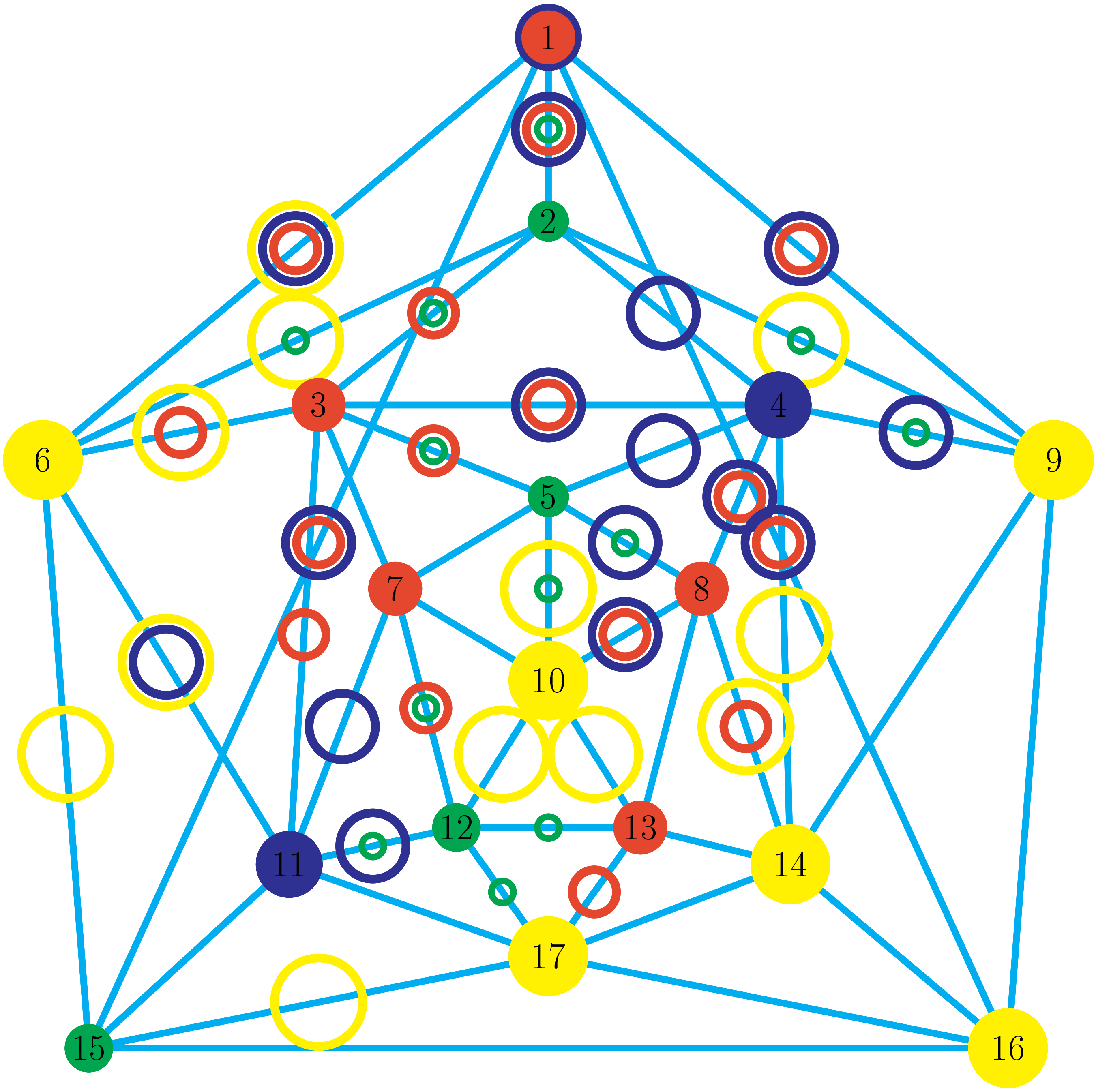


Figure 152.

instruction 318: unplace edge 8-13 Blue Checker

instruction 319: place edge 4-5 Blue Checker

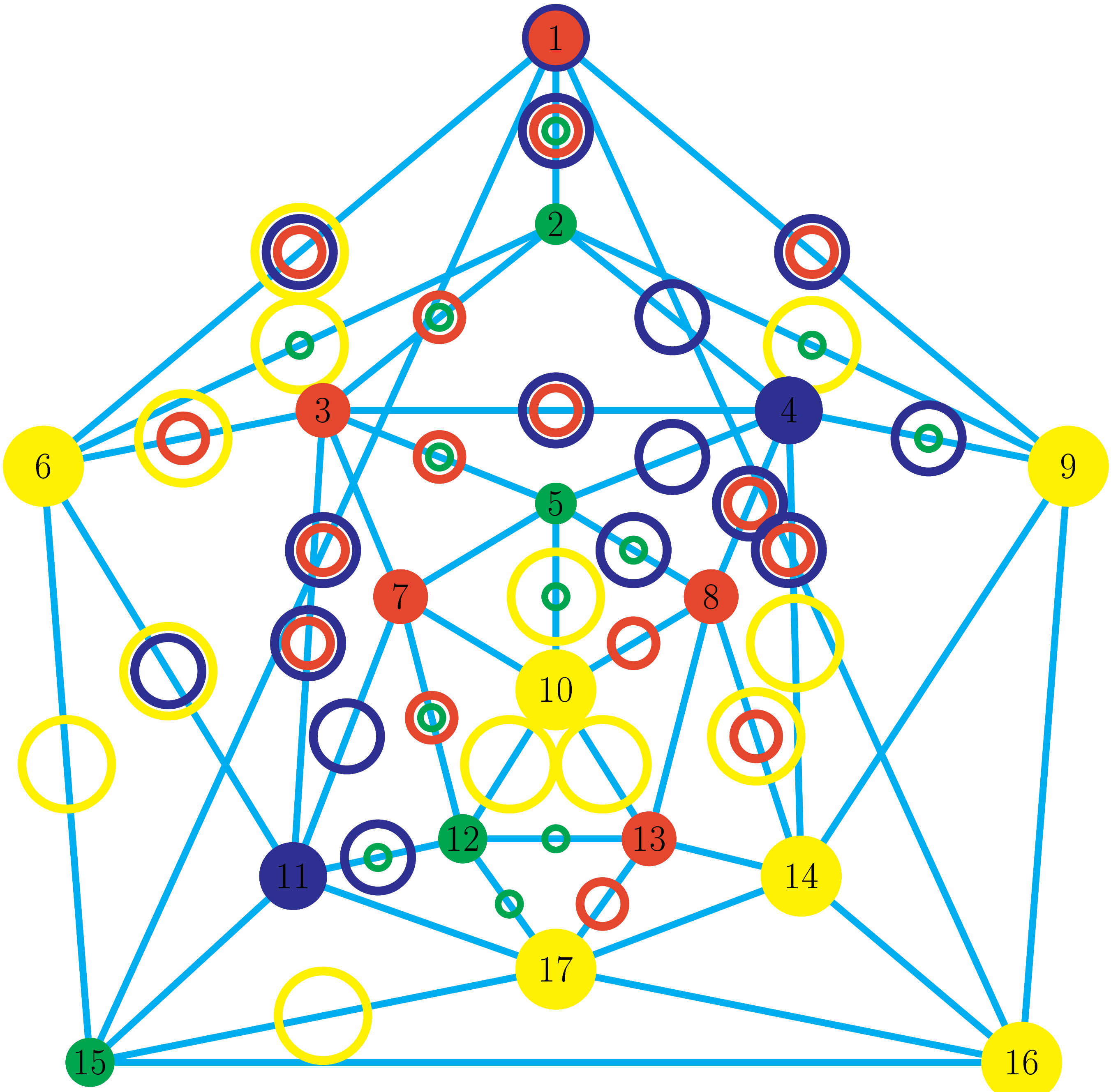


Figure 153.

instruction 320: unplace edge 8-10 Blue Checker

instruction 321: place edge 3-11 Blue Checker

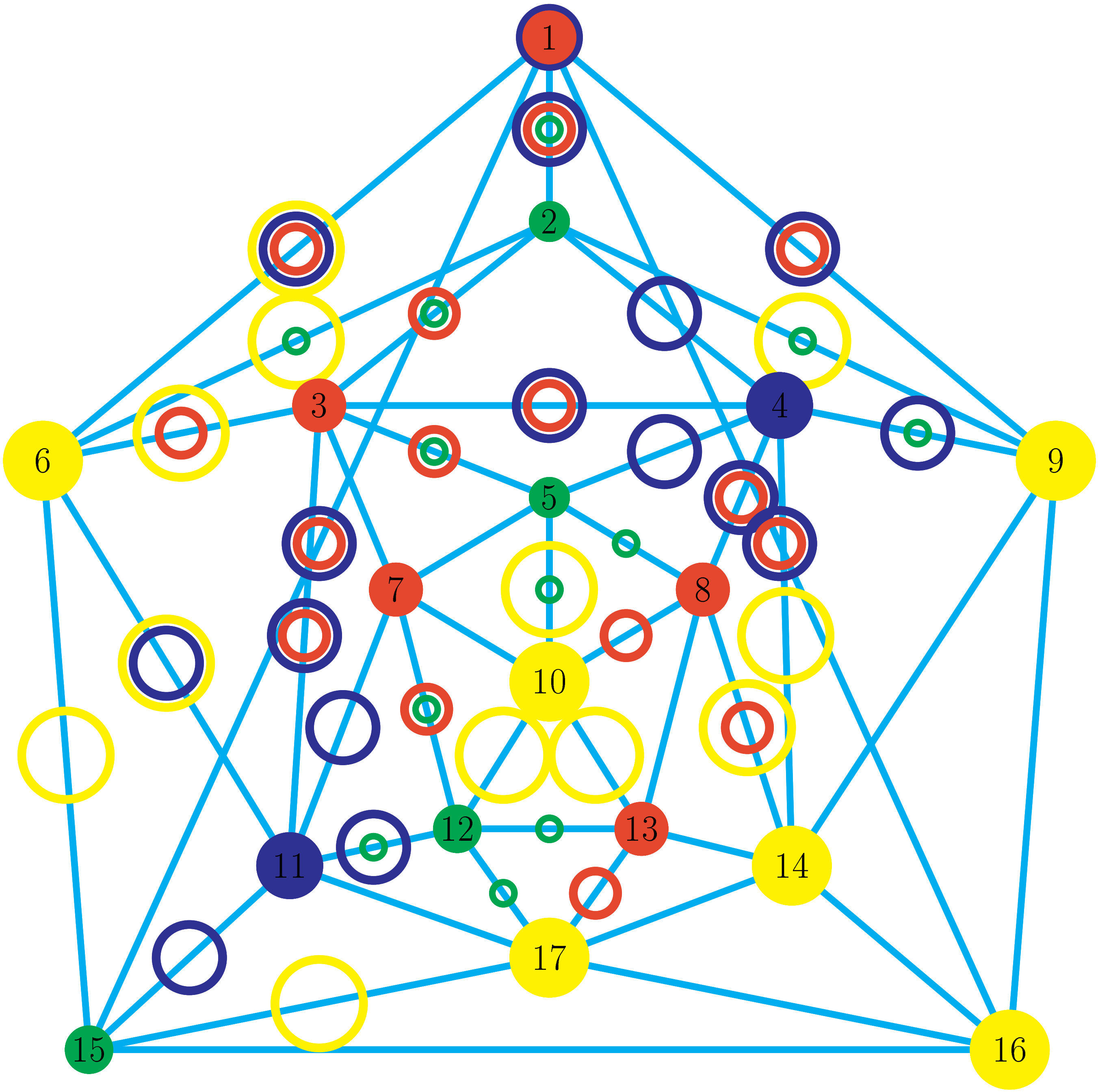


Figure 154.

instruction 322: unplace edge 8-5 Blue Checker
instruction 323: place edge 15-11 Blue Checker

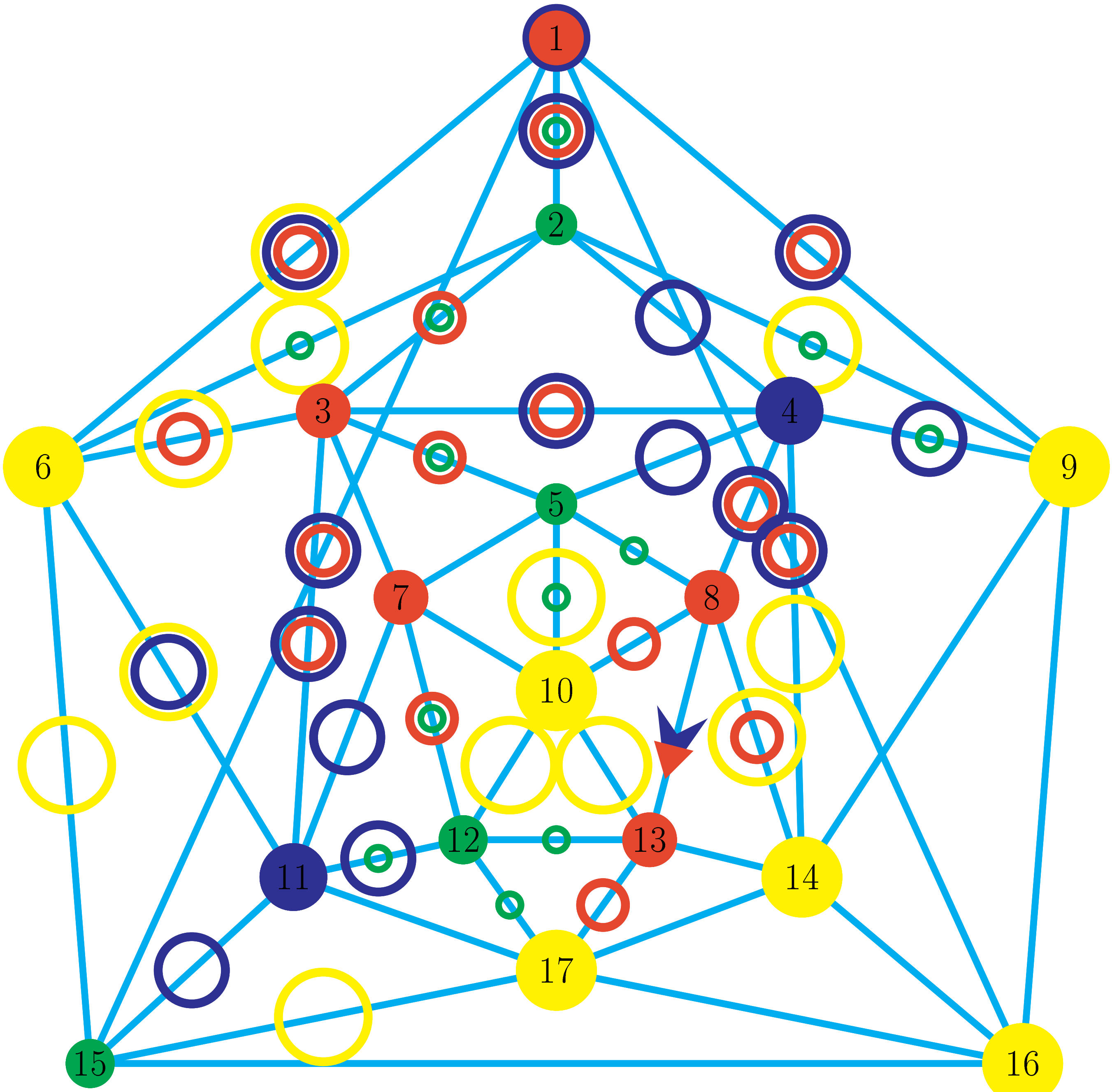


Figure 155.

instruction 324: place edge 8->13 Red DeletionArrow
instruction 325: place edge 8->13 Blue InsertionArrow

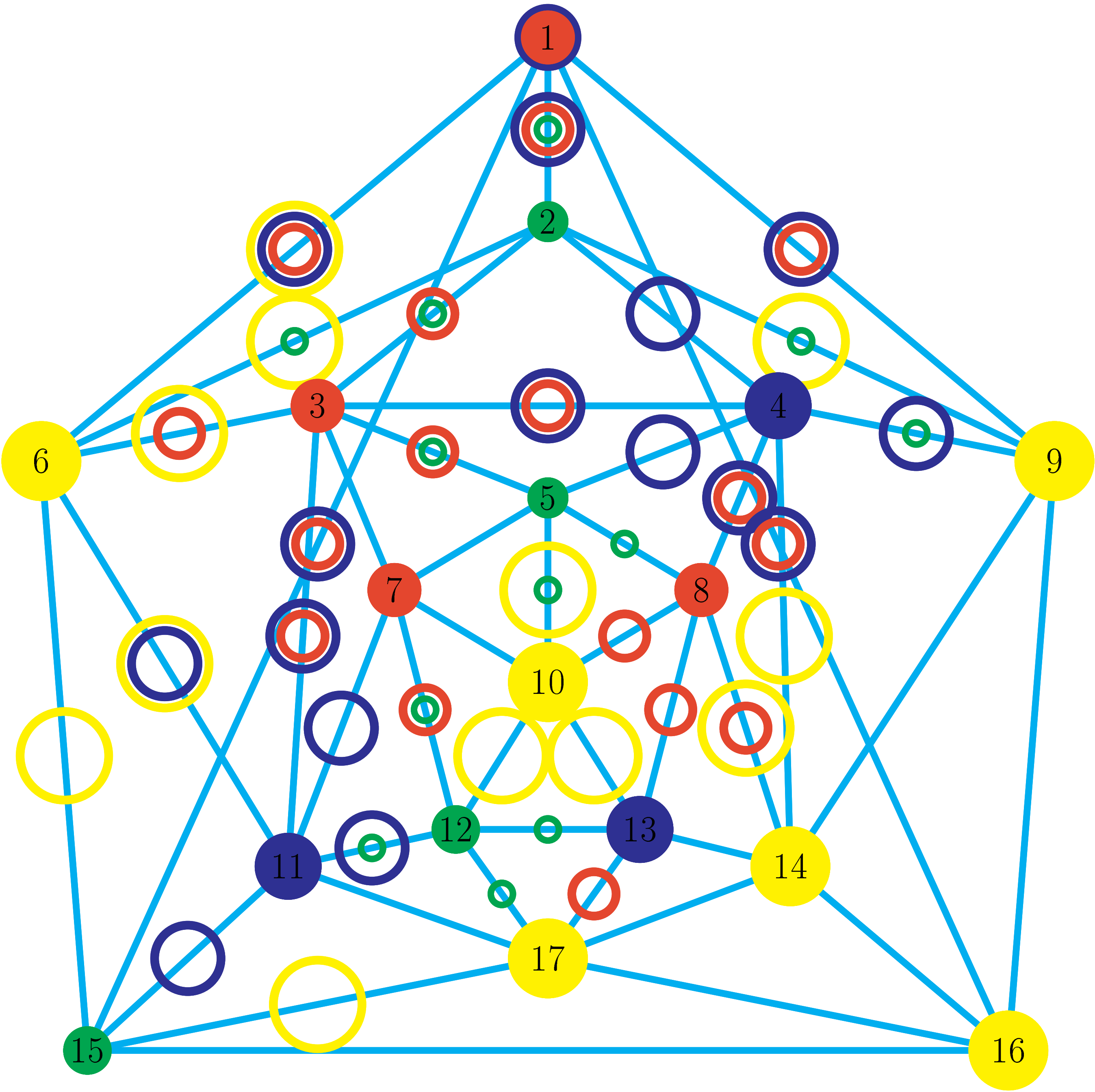


Figure 156.

instruction 326: unplace edge 8->13 Red DeletionArrow
instruction 327: unplace edge 8->13 Blue InsertionArrow
instruction 328: unplace vertex 13 Red Checker;
instruction 329: place vertex 13 Blue Checker;
instruction 330: place edge 13-8 Red Checker

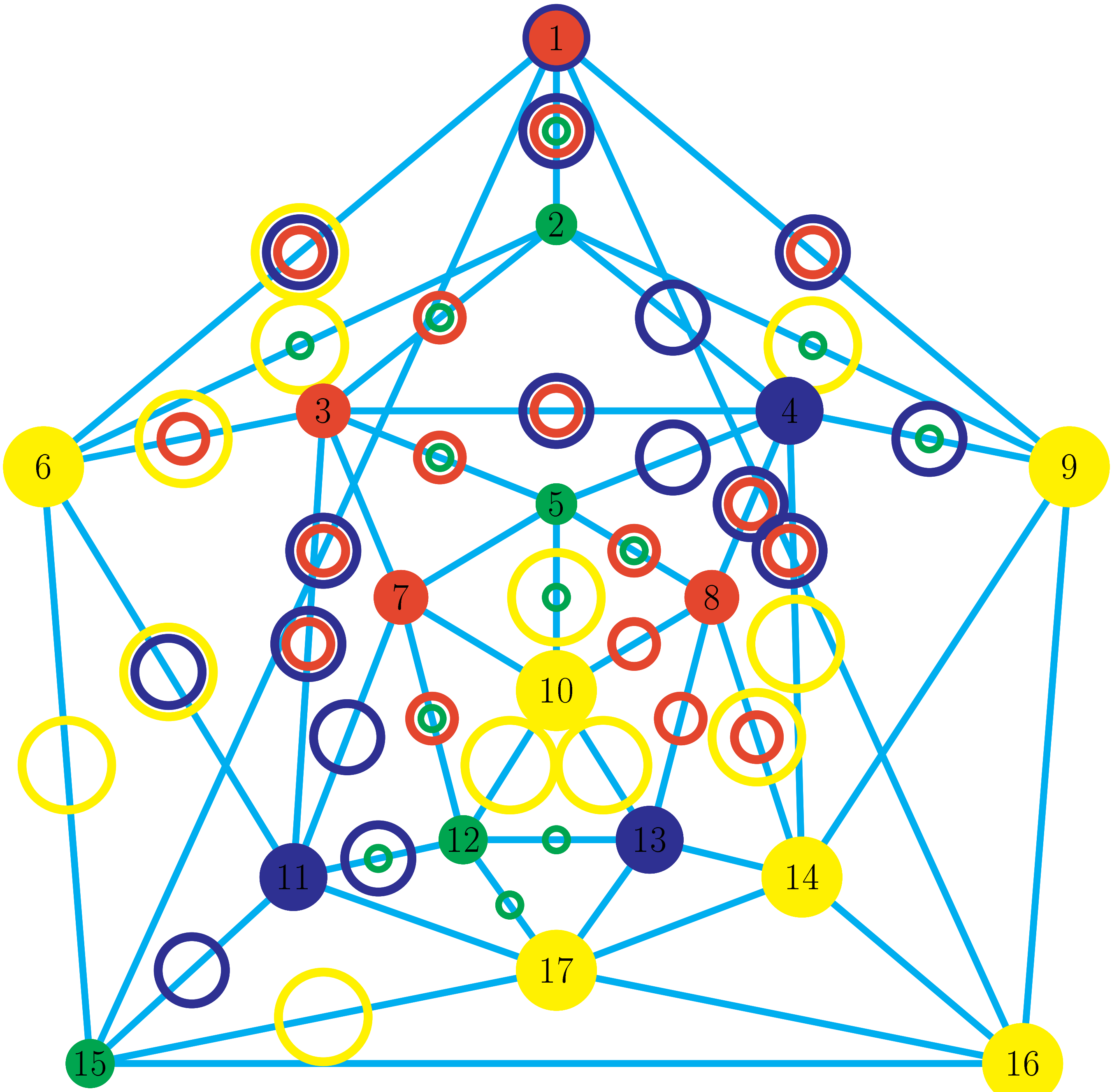


Figure 157.

instruction 331: unplace edge 13-17 Red Checker
instruction 332: place edge 5-8 Red Checker

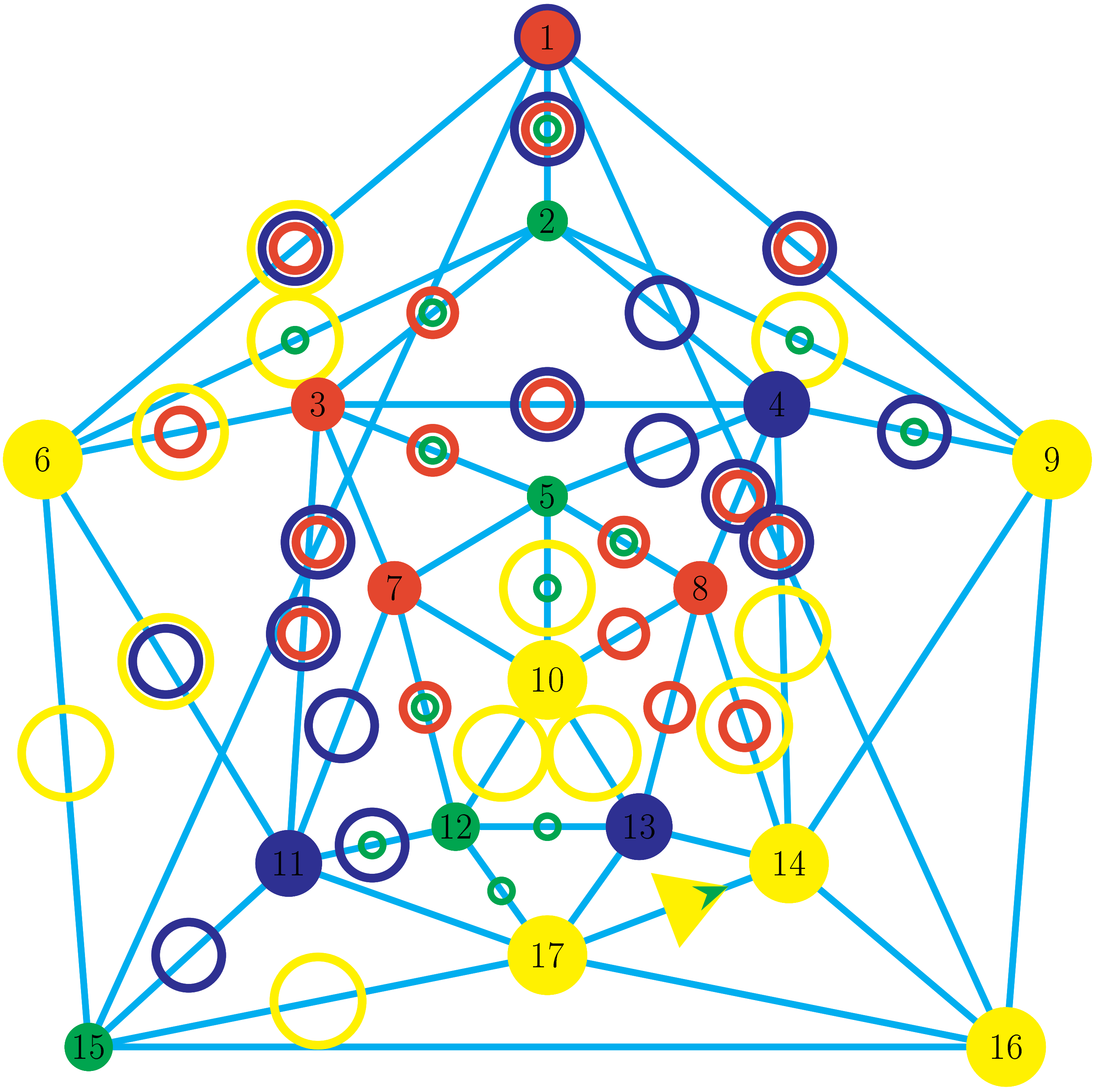


Figure 158.

instruction 333: place edge 17->14 Yellow DeletionArrow
instruction 334: place edge 17->14 Green InsertionArrow

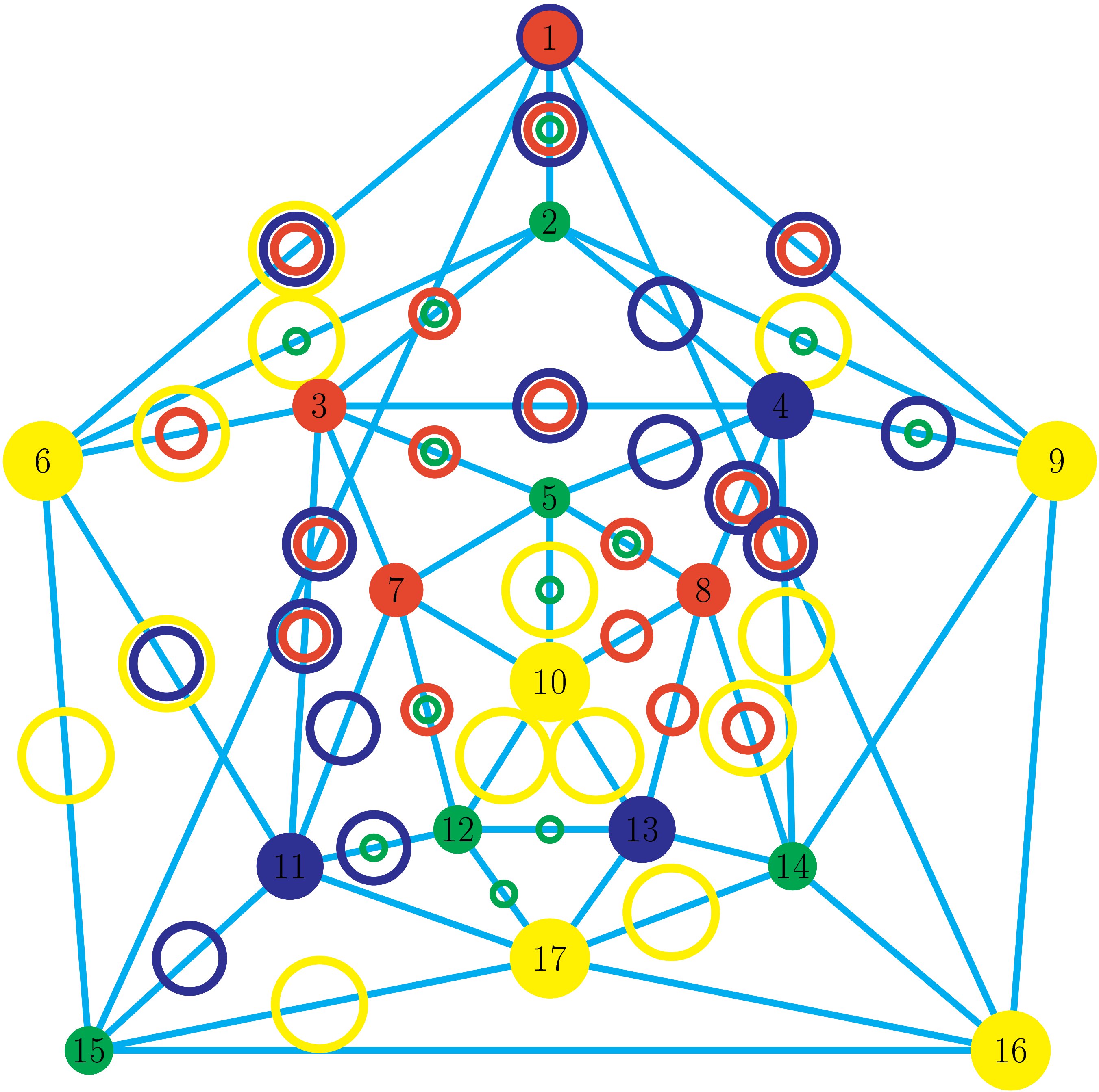


Figure 159.

instruction 335: unplace edge 17->14 Yellow DeletionArrow
instruction 336: unplace edge 17->14 Green InsertionArrow
instruction 337: place edge 17-14 Yellow Checker
instruction 338: unplace vertex 14 Yellow Checker;
instruction 339: place vertex 14 Green Checker;

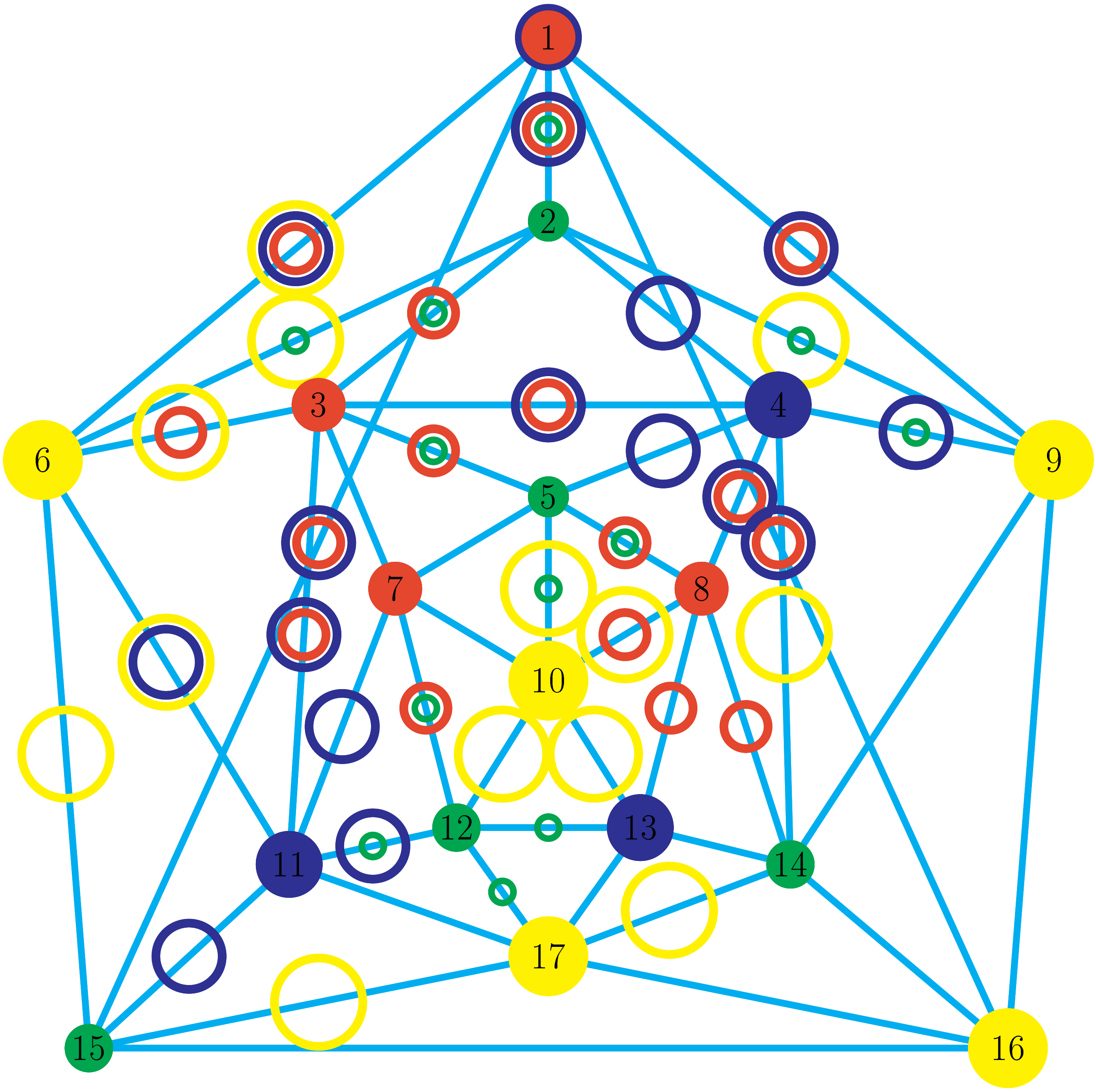


Figure 160.

instruction 340: unplace edge 8-14 Yellow Checker

instruction 341: place edge 8-10 Yellow Checker

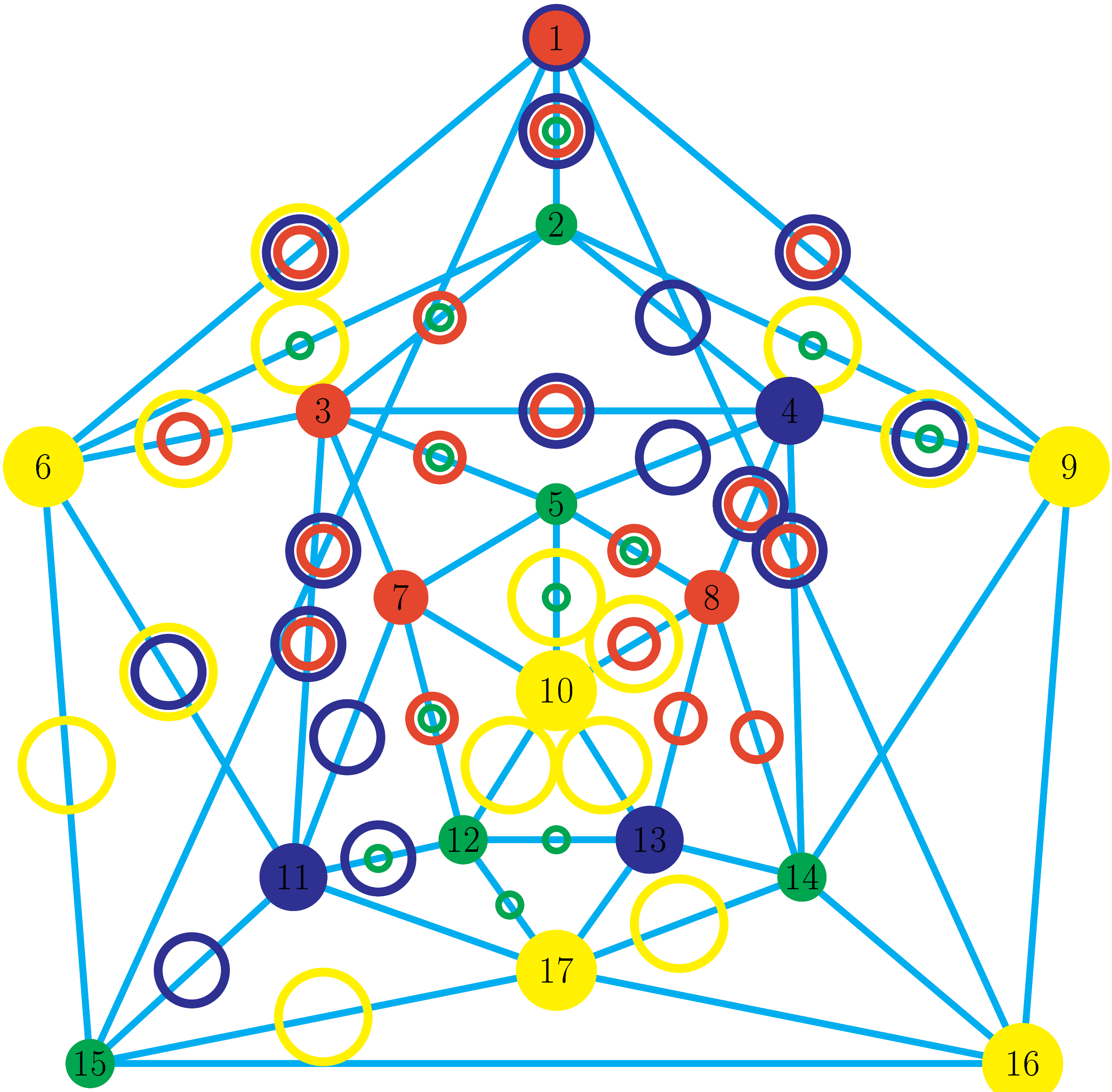


Figure 161.

instruction 342: unplace edge 4-14 Yellow Checker

instruction 343: place edge 4-9 Yellow Checker

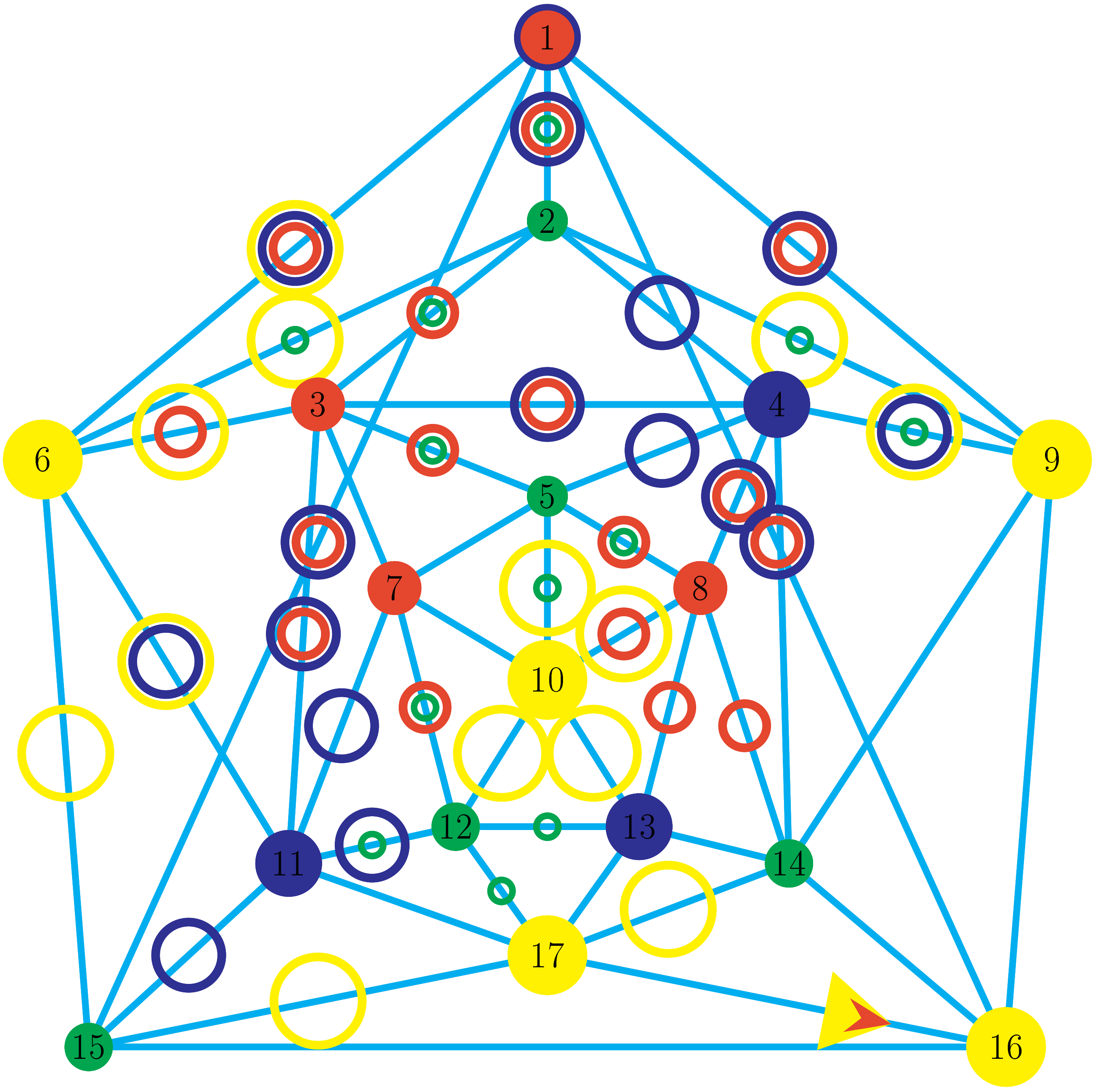


Figure 162.

instruction 344: place edge 17->16 Yellow DeletionArrow
instruction 345: place edge 17->16 Red InsertionArrow

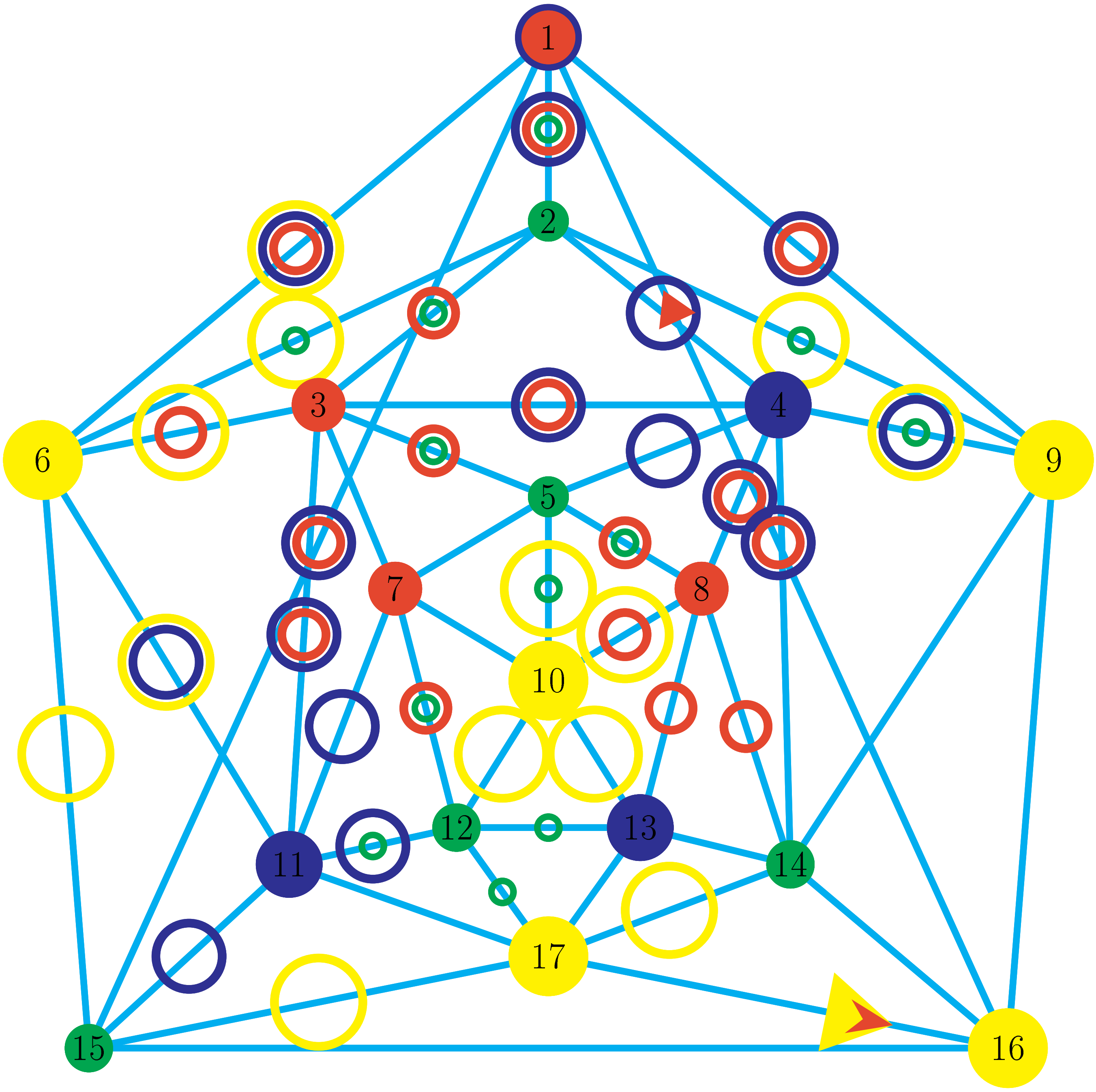


Figure 163.

instruction 346: place edge 16->1 Red DeletionArrow

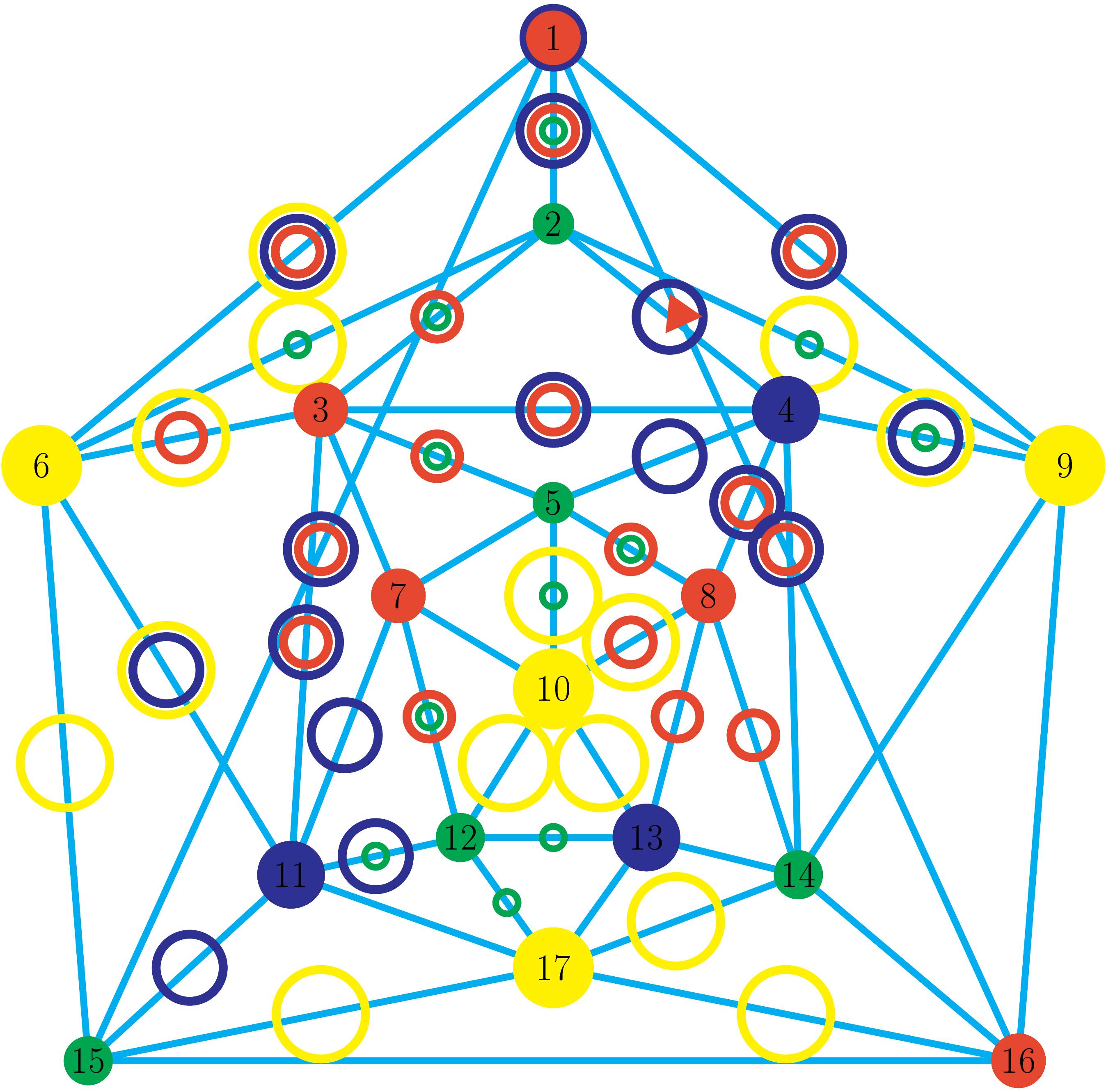


Figure 164.

instruction 347: unplace edge 17->16 Yellow DeletionArrow
instruction 348: unplace edge 17->16 Red InsertionArrow
instruction 349: place edge 17-16 Yellow Checker
instruction 350: unplace vertex 16 Yellow Checker;
instruction 351: place vertex 16 Red Checker;

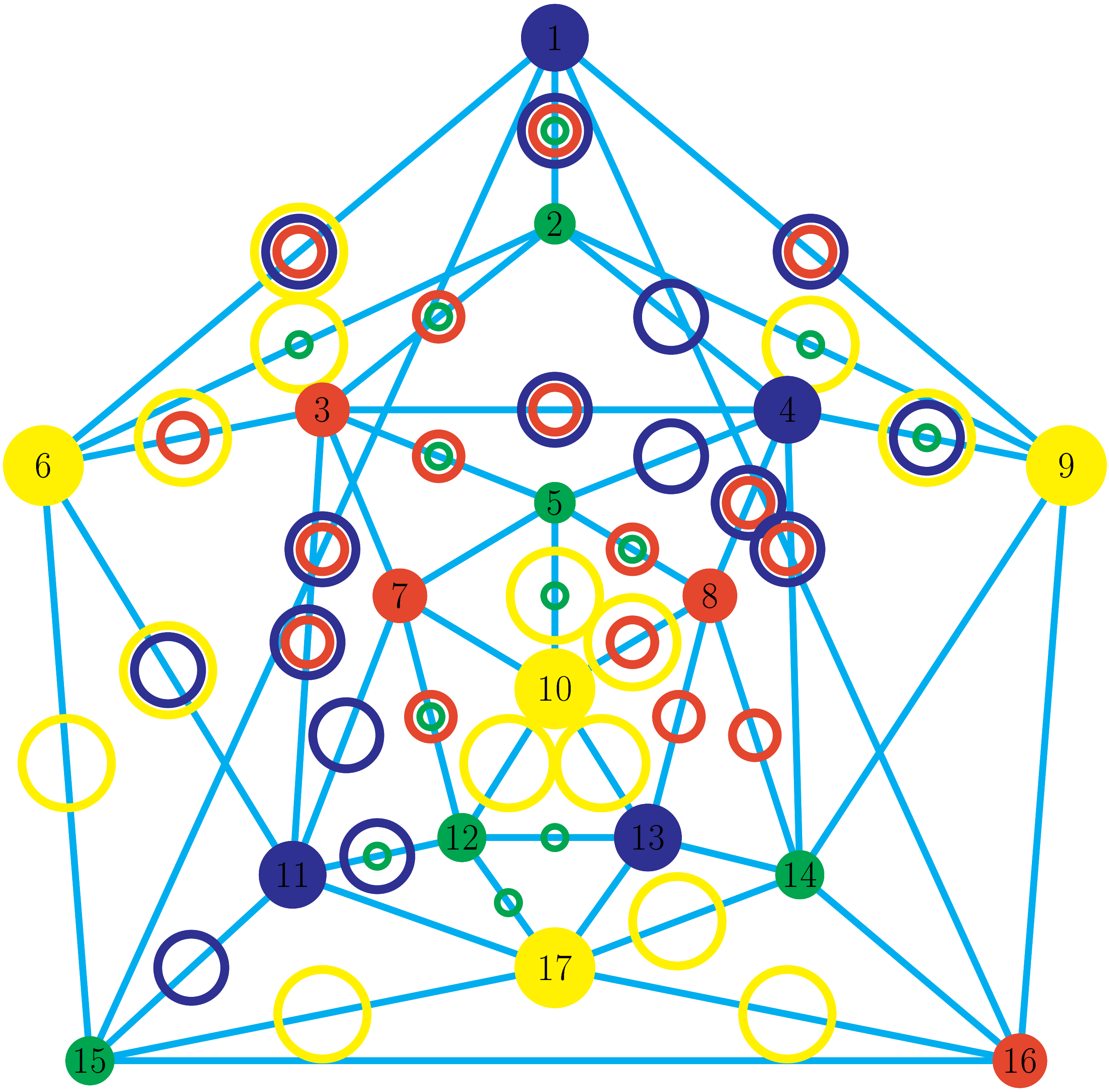


Figure 165.

instruction 352: unplace edge 16->1 Red DeletionArrow
instruction 353: unplace vertex 1 Red Checker;

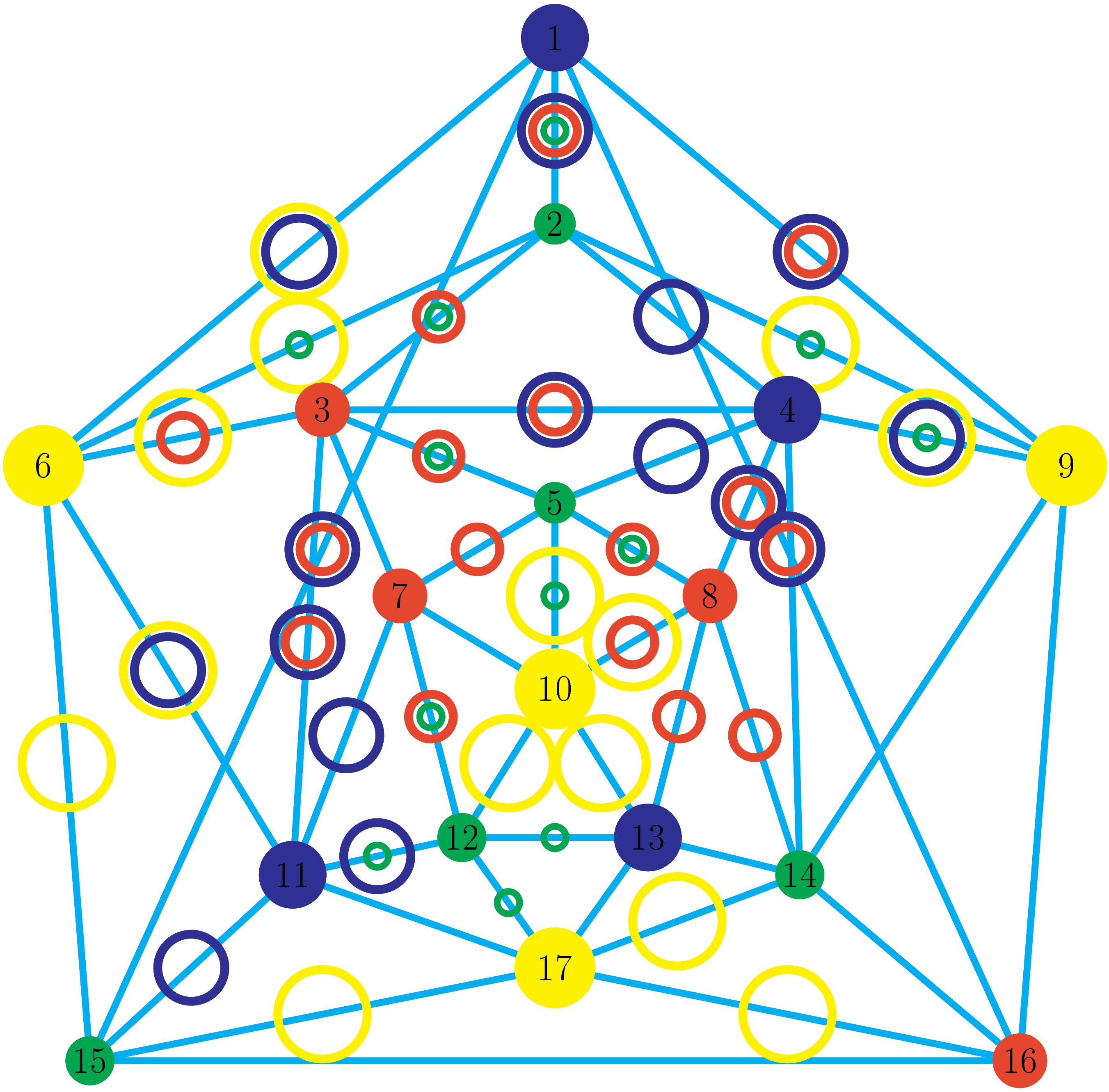


Figure 166.

instruction 354: unplace edge 6-1 Red Checker

instruction 355: place edge 7-5 Red Checker

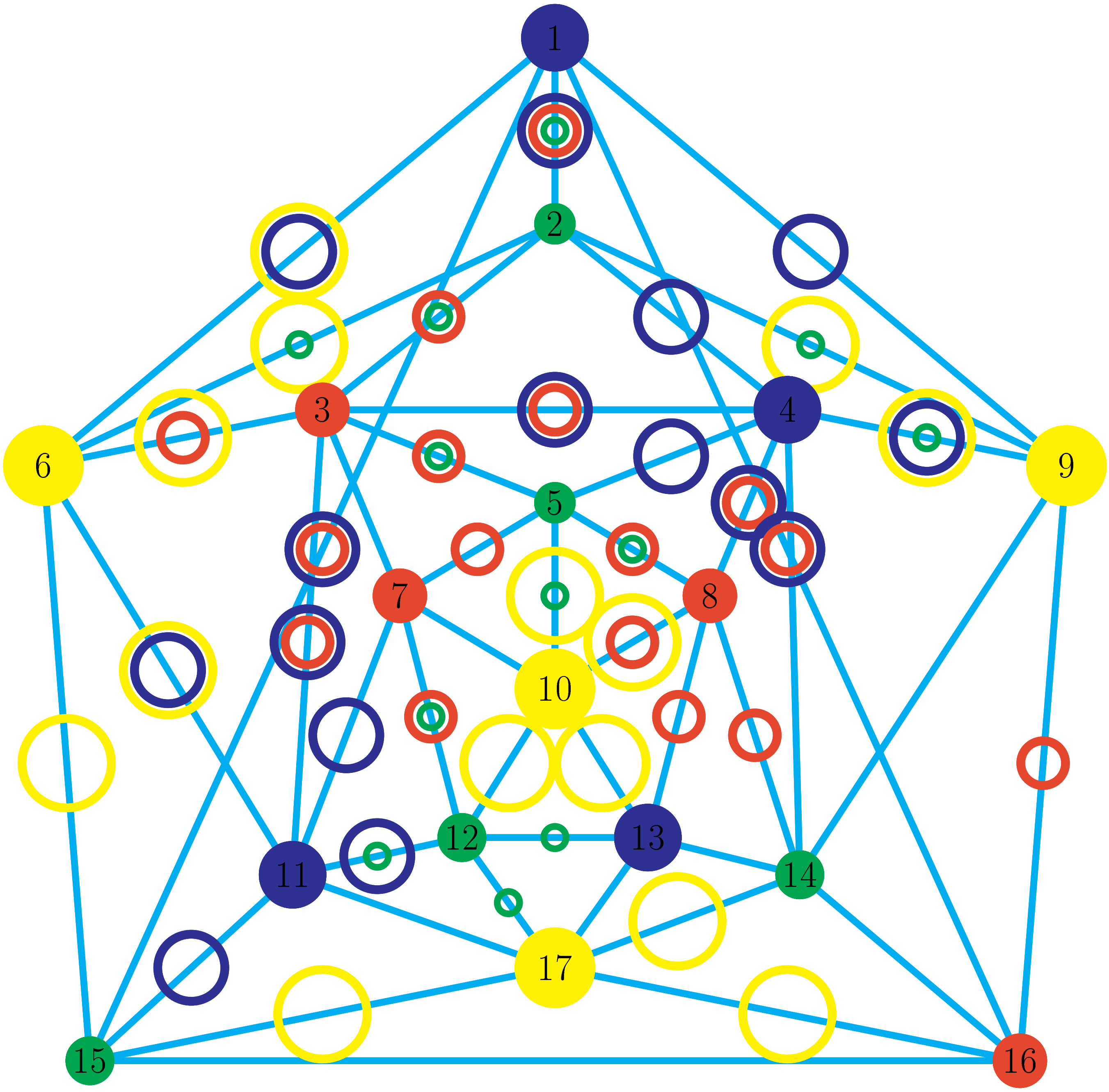


Figure 167.

instruction 356: unplace edge 9-1 Red Checker

instruction 357: place edge 9-16 Red Checker

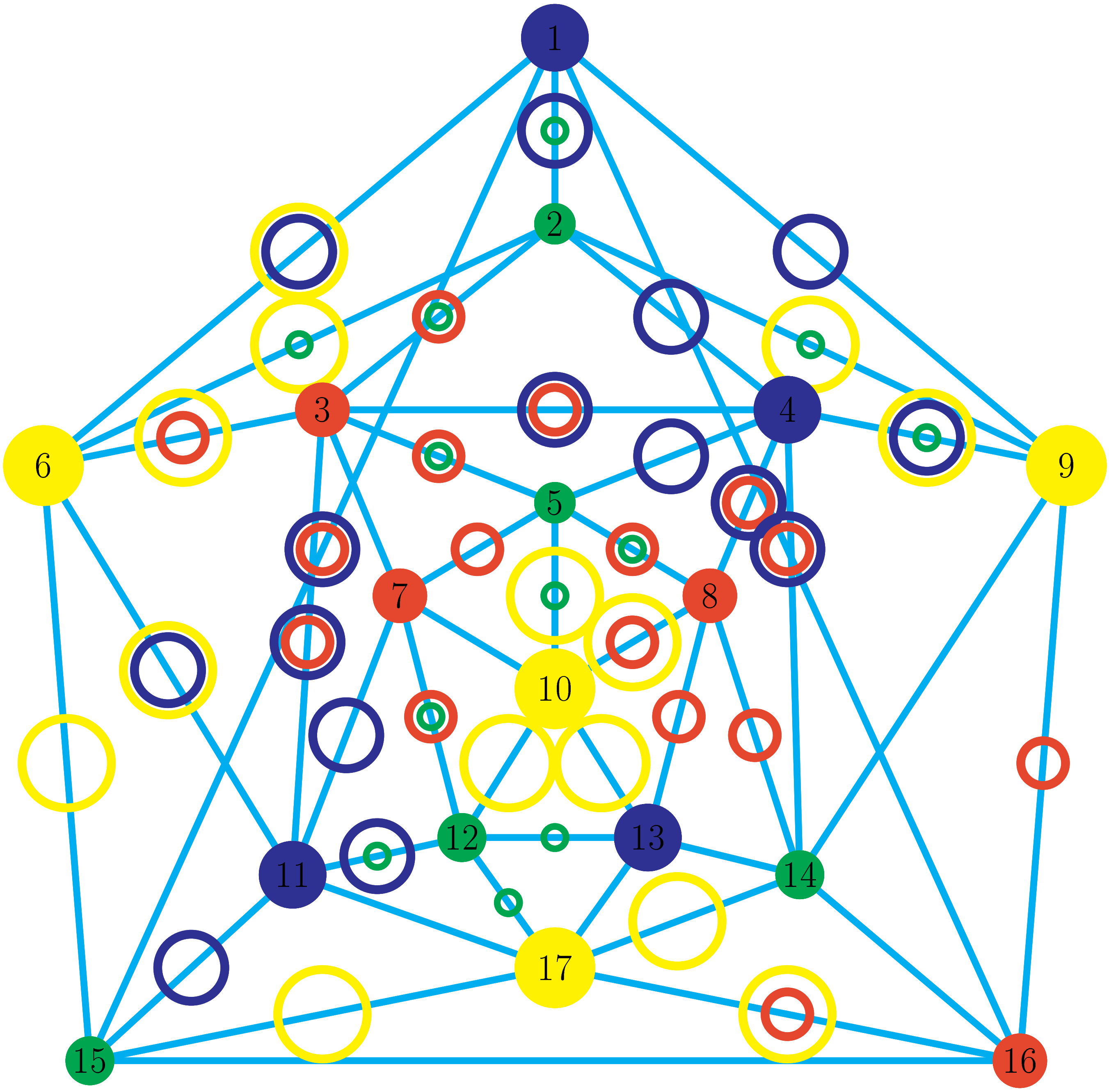


Figure 168.

instruction 358: unplace edge 2-1 Red Checker
instruction 359: place edge 17-16 Red Checker

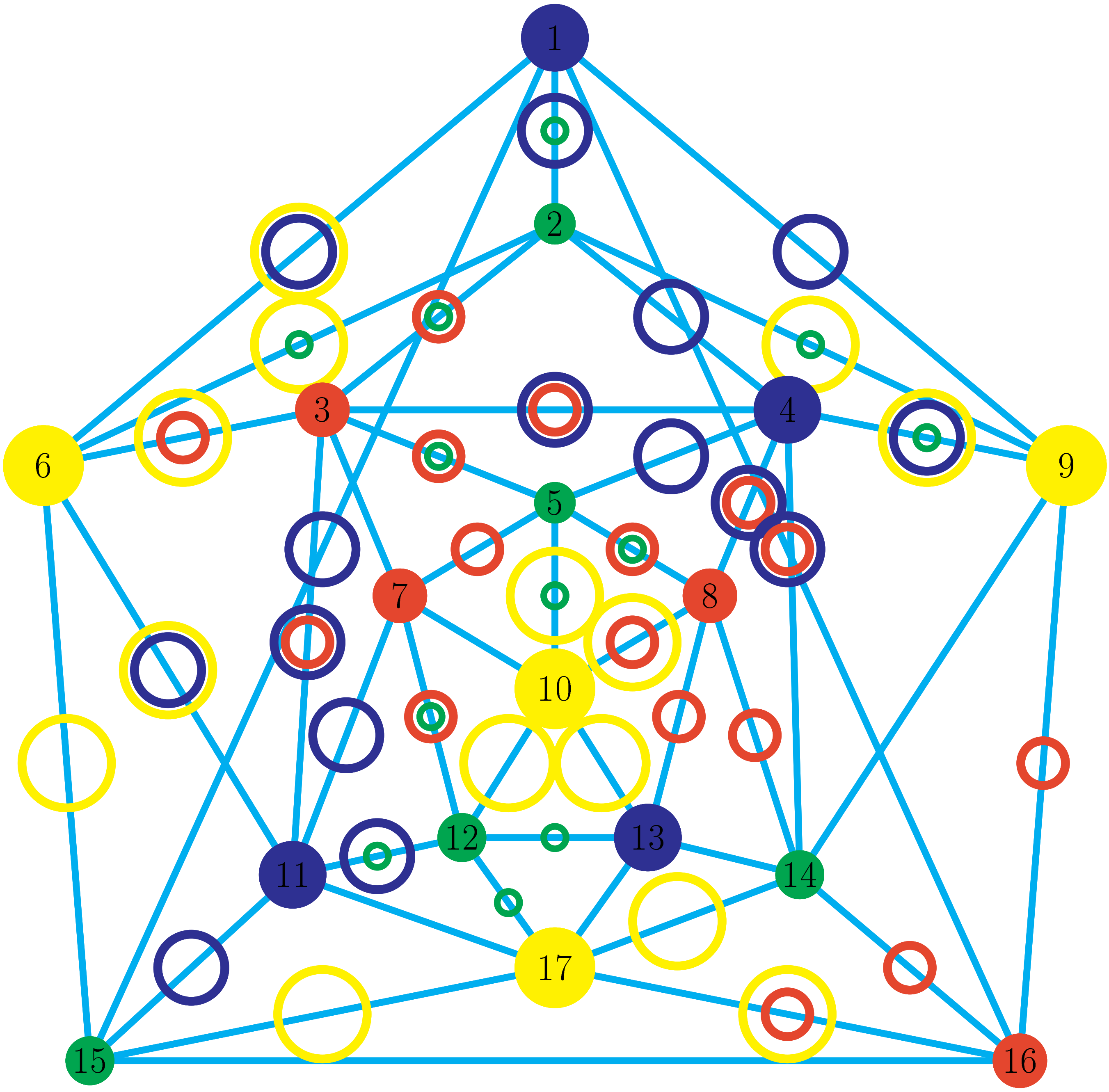


Figure 169.

instruction 360: unplace edge 15-1 Red Checker

instruction 361: place edge 14-16 Red Checker

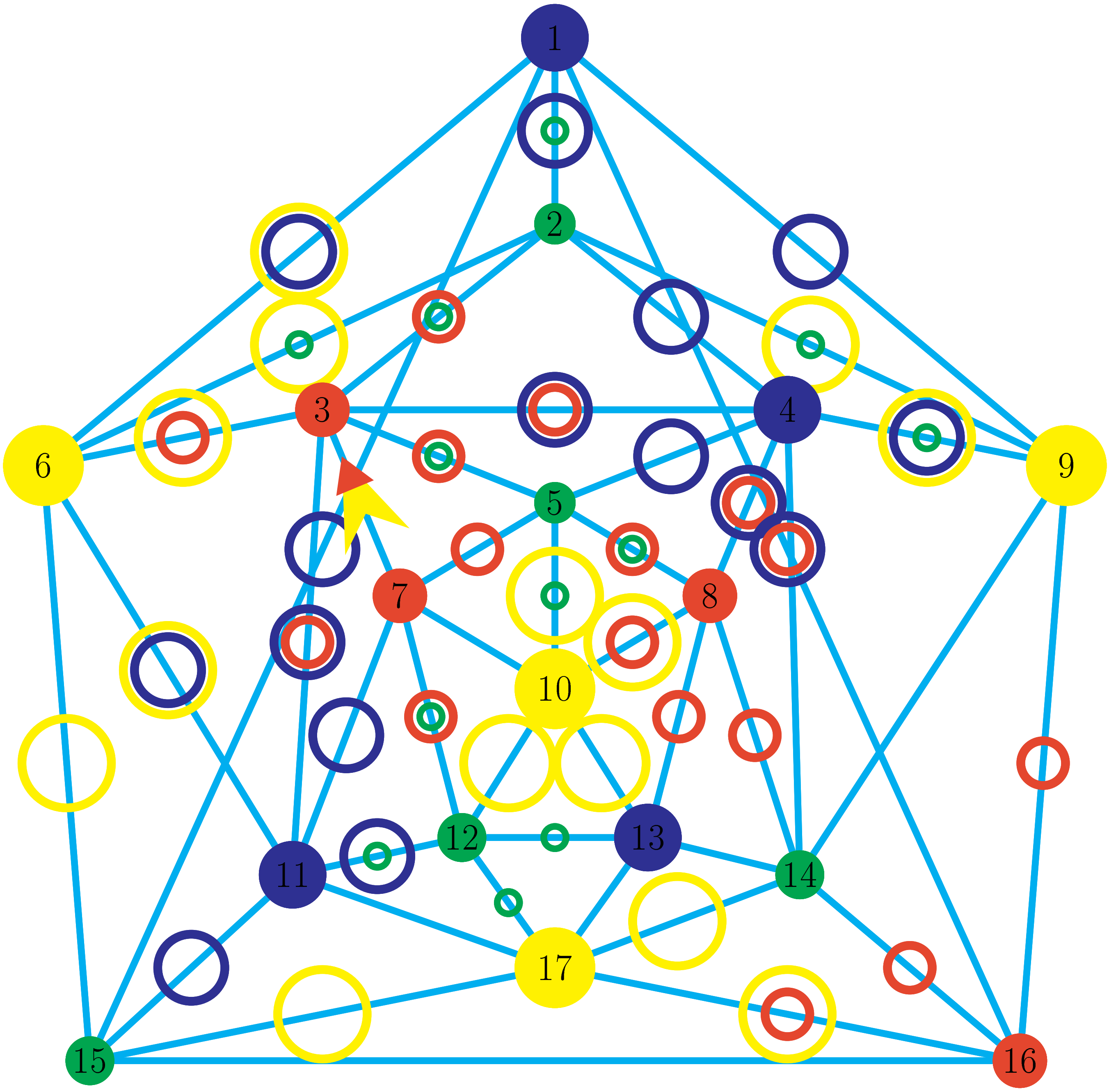


Figure 170.

instruction 362: place edge 7->3 Red DeletionArrow
instruction 363: place edge 7->3 Yellow InsertionArrow

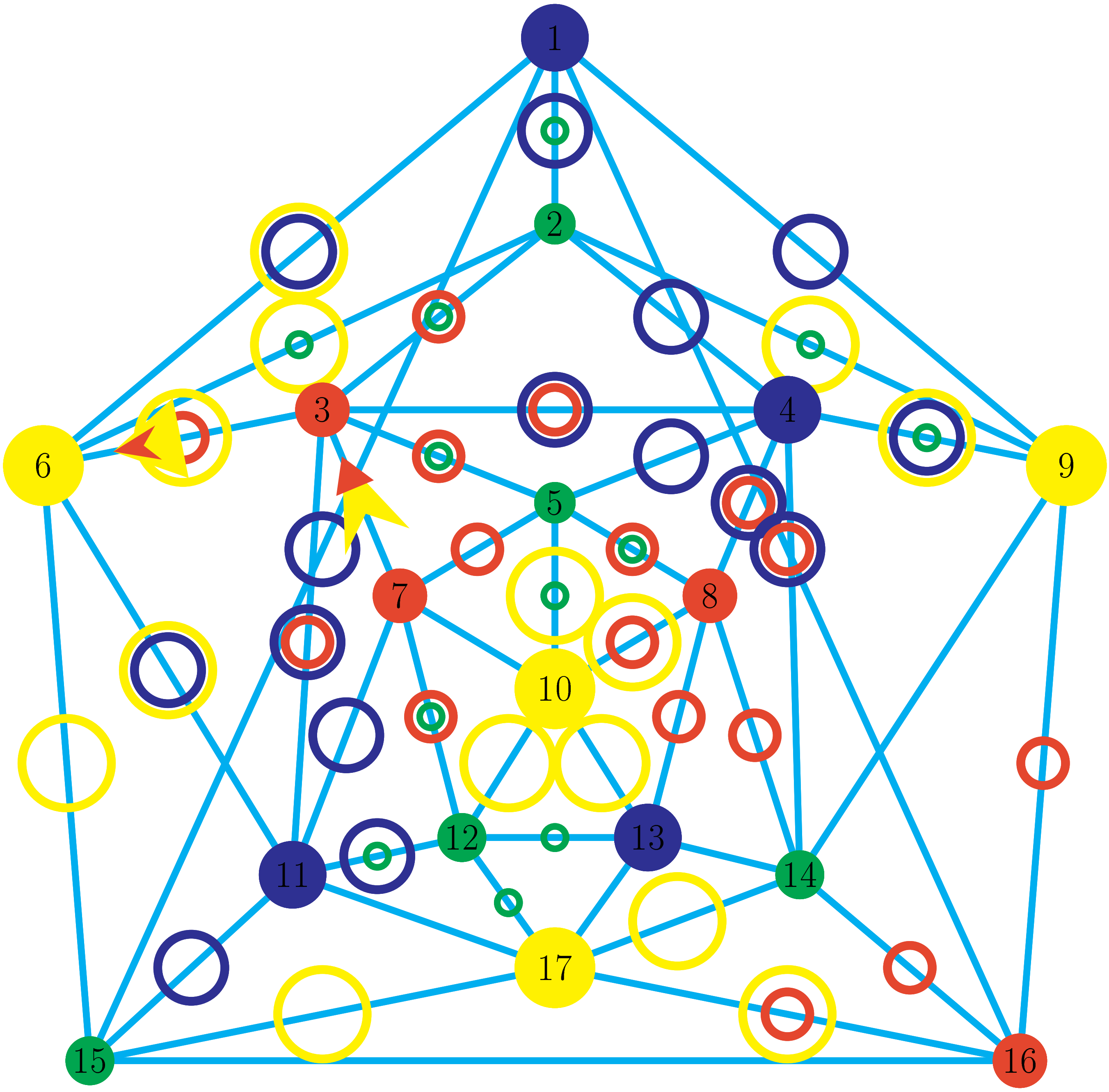


Figure 171.

instruction 364: place edge 3->6 Yellow DeletionArrow
instruction 365: place edge 3->6 Red InsertionArrow

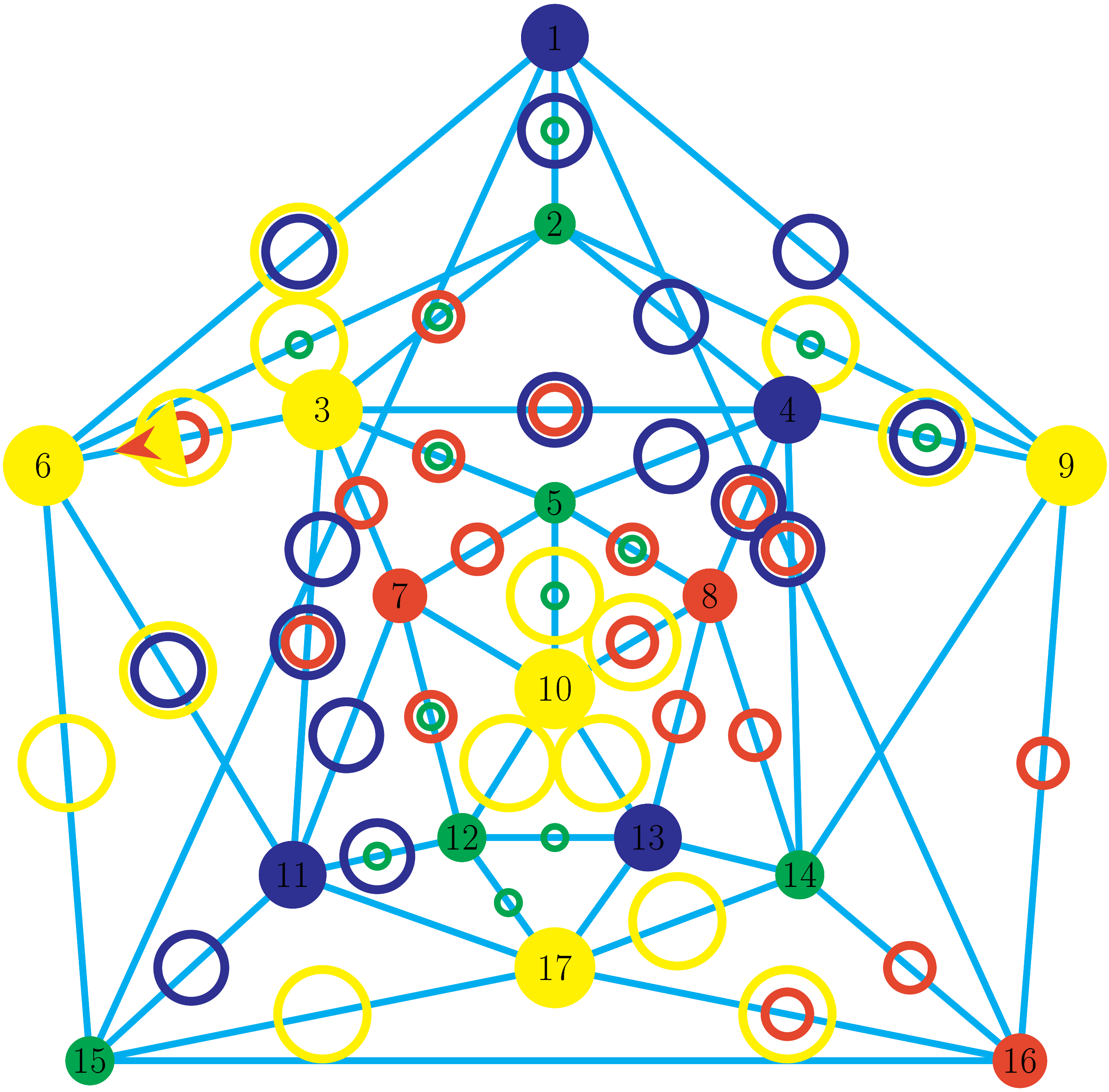


Figure 172.

instruction 366: unplace edge 7->3 Red DeletionArrow
instruction 367: unplace edge 7->3 Yellow InsertionArrow
instruction 368: place edge 7-3 Red Checker
instruction 369: unplace vertex 3 Red Checker;
instruction 370: place vertex 3 Yellow Checker;

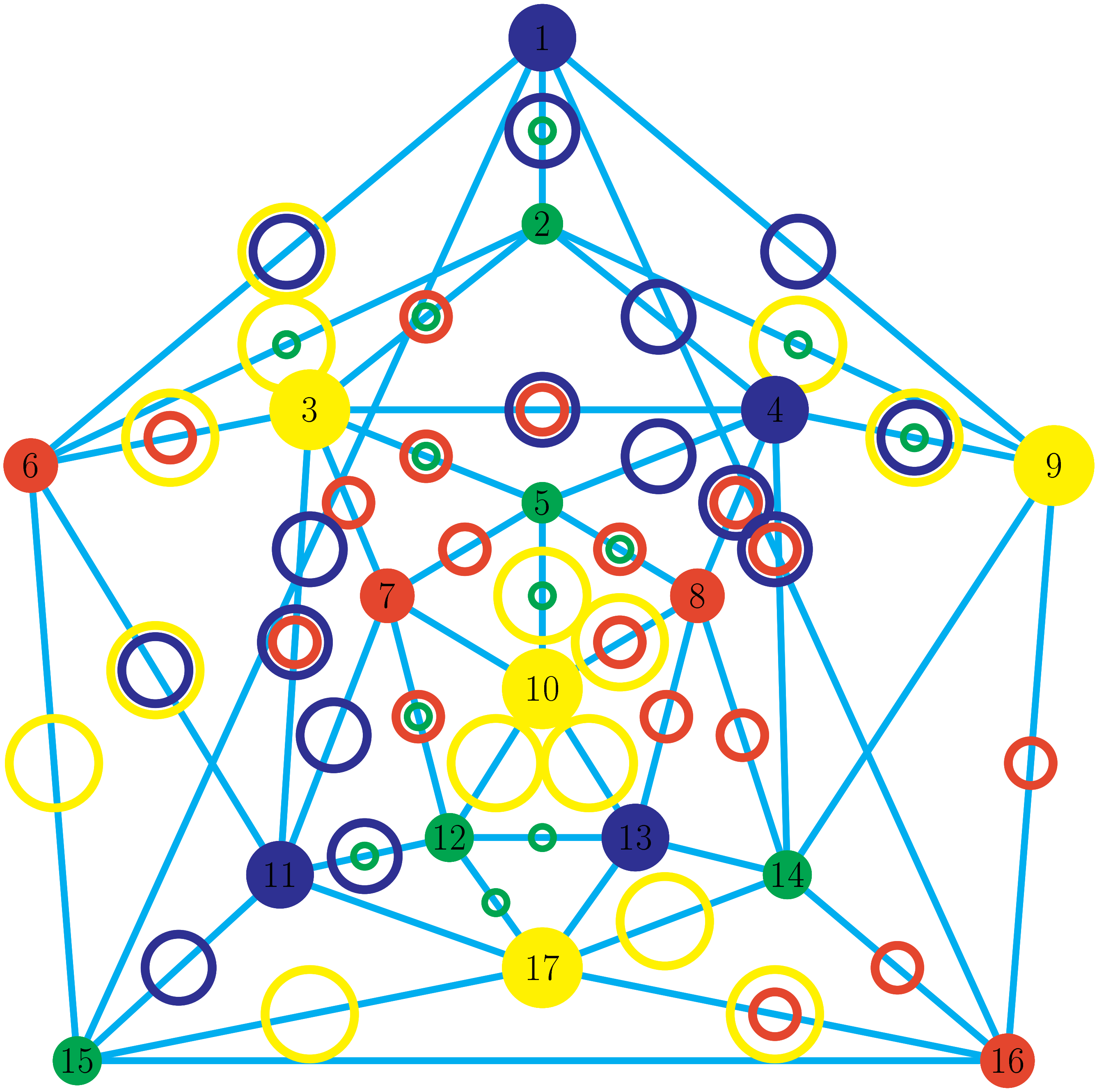
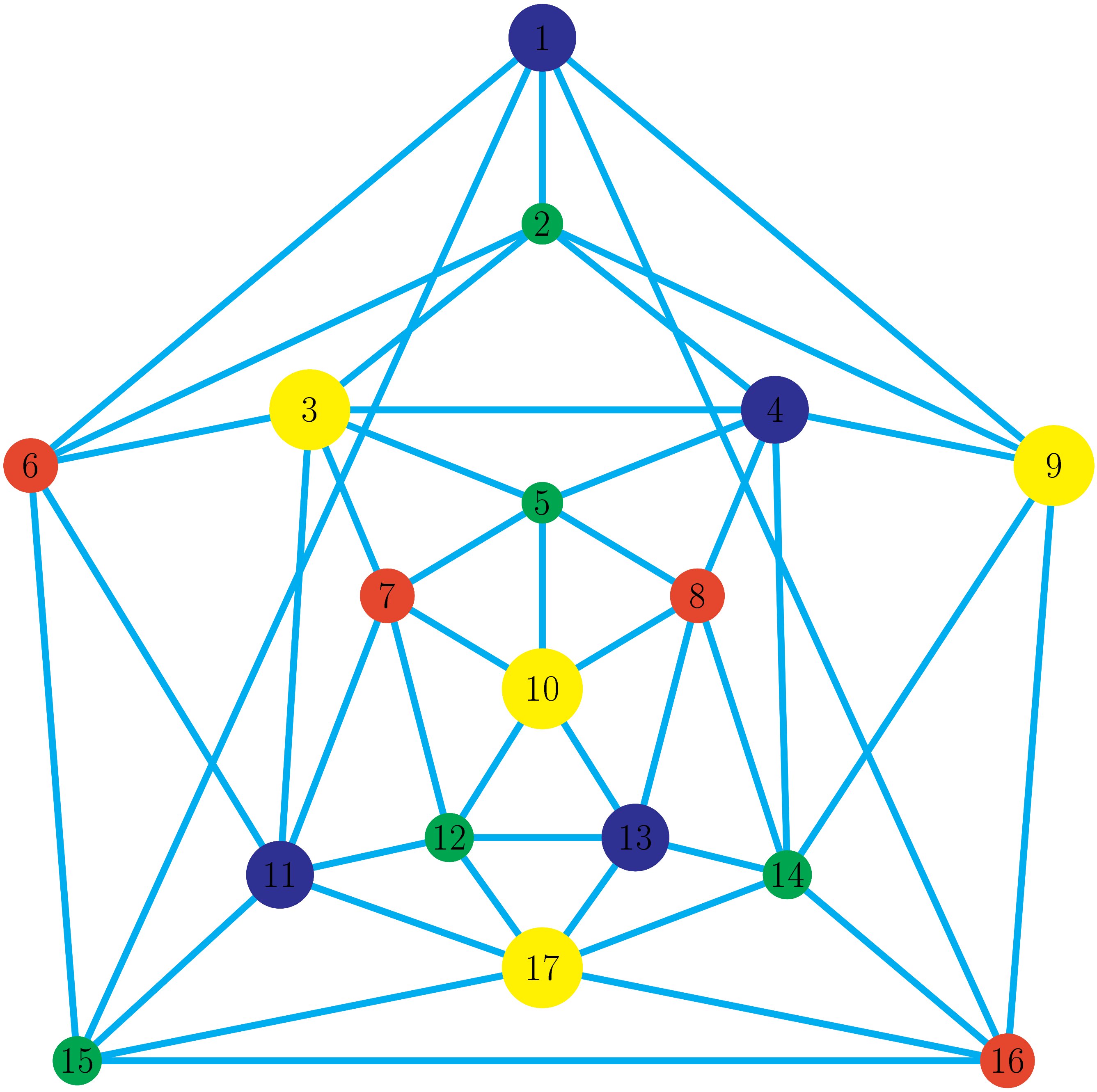


Figure 173.

instruction 371: unplace edge 3->6 Yellow DeletionArrow
instruction 372: unplace edge 3->6 Red InsertionArrow
instruction 373: unplace vertex 6 Yellow Checker;
instruction 374: place vertex 6 Red Checker;

THE GRAPH HAS BEEN PROPERLY FOUR COLORED



THE END

Figure 174.

THE END

Section 3. Observations

It is property B that makes it easy to find the next alternating chain because it eliminates the problem of self conflicting chains. At certain points it looks as though you could grab extra territory by placing an extra circle on an edge without violating property A, or B. But this would be a mistake. By adding an extra circle gratuitously you would have spent a cartridge with nothing to show for it in the form of work done, a gross inefficiency. The algorithm is designed to break the problem into components and to solve one component at a time. Since it is easy to find the right alternating chain by sight for a human, it should be "easy" algorithmicaly.

References

Clairmont, John. “TOWARD A NEW PROOF OF THE FOUR COLOR THEOREM.” *viXra.org*, 2024.