How to make a conjecture about the more compact sequence with no 3 terms in arithmetic progression using Mathematica© and www.oeis.org

Edoardo GUEGLIO

egueglio@gmail.com

This is an example of how with a mathematical software you can make a mathematical conjecture and help to prove it.

First of all we need to create a function that check if a sequence has the property that no 3 terms are in arithmetic progression:

In[*]:=
```
checkno3Q[s:{__Integer?Positive}]:=Module[{i,j,l=Length[s]},
    If[l<3,Return[True]];
    For[i=1,i<l-1,i++,
        For[j=i+1,j<l,j++,
            If[MemberQ[s,2s⟦j⟧-s⟦i⟧],Return[False]]
        ]
    ];
    Return[True]
]
checkno3Q[{1,2,3}]
checkno3Q[{1,2,4,5,7}]
checkno3Q[{1,2,4,6,8}]
checkno3Q[{1,2,4,8,11}]
```

Out[*]=
```
False
```

Out[*]=
```
False
```

Out[*]=
```
False
```

Out[*]=
```
True
```

Then we define a greedy algorithm that take a function, a partial sequence and a number of items and try to add one item at a time to the initial sequence starting from the successor of last item in sequence, always satisfying the control function:

In[ ]:=
```
addtoseq[f_,s:{__Integer?Positive},n_Integer?Positive]:=Module[{i,j,l=Length[s],ss=s},
    If[!f[s],Return[{}]];
    For[i=0,i<n,i++,
        For[j=ss[[-1]]+1,True,j++,
            If[f[Append[ss,j]],
                AppendTo[ss,j];
                Break[]
            ]
        ]
    ];
    Return[ss]
]
```

Consider now the sequence with the minimal information given, the subsequence {1}:

In[ ]:=
```
s1=addtoseq[checkno3Q,{1},70]
```

Out[ ]=
```
{1, 2, 4, 5, 10, 11, 13, 14, 28, 29, 31, 32, 37, 38, 40, 41, 82, 83, 85, 86, 91,
 92, 94, 95, 109, 110, 112, 113, 118, 119, 121, 122, 244, 245, 247, 248, 253, 254,
 256, 257, 271, 272, 274, 275, 280, 281, 283, 284, 325, 326, 328, 329, 334, 335, 337,
 338, 352, 353, 355, 356, 361, 362, 364, 365, 730, 731, 733, 734, 739, 740, 742}
```

Now the goal is to find an algorithm to obtain the sequence items by some functions avoiding the greedy algorithm to speed up also the creation of a sequence with many more items. The sequence can be partitioned in couples of consecutive values so for now we focus only on the odd positions.

In[ ]:=
```
s1[[1 ;; ;; 2]]
```

Out[ ]=
```
{1, 4, 10, 13, 28, 31, 37, 40, 82, 85, 91, 94, 109, 112, 118, 121, 244, 247, 253,
 256, 271, 274, 280, 283, 325, 328, 334, 337, 352, 355, 361, 364, 730, 733, 739, 742}
```

If you want to get informations about a non trivial sequence, go to oeis.org. We find a couple of functions: 3n-2 and 3n+1 and if we start with seed sequence {1} and apply the functions to the items in sequence and append if not already present, we quickly rebuild the original sequence as from this code:

In[ ]:=
```
createsequence[start:{__Integer?Positive},fl:{__Function},max_Integer?Positive]:=Module[{i=1,
    While[Length[s]<max,
        ss=MapThread[Apply,{fl,Table[{s[[i]]},l]}];
        s=Union[s,ss];
        i++
    ];
    Return[s]
]
s2=createsequence[{1},{3#-2&,3#+1&},50]
```

Out[ ]=
```
{1, 4, 10, 13, 28, 31, 37, 40, 82, 85, 91, 94, 109, 112, 118, 121, 244, 247,
 253, 256, 271, 274, 280, 283, 325, 328, 334, 337, 352, 355, 361, 364, 730, 733,
 739, 742, 757, 760, 766, 769, 811, 814, 820, 823, 838, 841, 847, 850, 973, 976}
```

At this point we have a quick way to reproduce algorithmically and to test the starting sequence s1 with much more items:

In[ ]:=
```
ss = createsequence[{1, 4}, {3 # - 2 &, 3 # + 1 &}, 2500];
checkno3Q[Union[ss, ss + 1]]
```

Out[ ]=
```
True
```

So at last for the 5000 items sequence the property is verified. But what happened if, always using createsequence, we check all possible values in a range for the first four items of the sequence and keep only the sequences with the property and with the lower last value:

```
In[]:= takefirstlast[ls:{{__Integer?Positive}..}]:=Module[{v=ls[[1,-1]]},Select[ls,#[[-1]]==v&]]
```

When we consider sequences of length 20 we have:

```
In[]:= For[i=1;r={},i<50,i++,
    For[j=i+1,j<50,j++,
        For[k=j+1,k<50,k++,
            For[l=k+1,l<50,l++,
                z=createsequence[{i,j,k,l},{3#-2&,3#+1&},20];
                If[checkno3Q[z],If[!MemberQ[r,z],AppendTo[r,z]]]
            ]
        ]
    ]
]
takefirstlast@Sort[r,#1[[-1]]<#2[[-1]]||(#1[[-1]]==#2[[-1]]&&#1[[1]]<#2[[1]])&]
```

```
Out[]=
{{15, 18, 24, 27, 43, 46, 52, 55, 70, 73, 79, 82, 127, 130, 136, 139, 154, 157, 163, 166}}
```

The “best” sequence seems to have nothing in common with our s1 but add just five items and s1 pop up:

```
In[]:= For[i=1;r={},i<50,i++,
    For[j=i+1,j<50,j++,
        For[k=j+1,k<50,k++,
            For[l=k+1,l<50,l++,
                z=createsequence[{i,j,k,l},{3#-2&,3#+1&},25];
                If[checkno3Q[z],If[!MemberQ[r,z],AppendTo[r,z]]]
            ]
        ]
    ]
]
takefirstlast@Sort[r,#1[[-1]]<#2[[-1]]||(#1[[-1]]==#2[[-1]]&&#1[[1]]<#2[[1]])&]
```

```
Out[]=
{{1, 4, 10, 13, 28, 31, 37, 40, 82, 85, 91, 94, 109, 112, 118, 121, 244,
  247, 253, 256, 271, 274, 280, 283, 325, 328}, {4, 10, 13, 28, 31, 37, 40, 82, 85,
  91, 94, 109, 112, 118, 121, 244, 247, 253, 256, 271, 274, 280, 283, 325, 328}}
```

Maybe it’s a casualty so proceed with 50 items:

```
In[]:= For[i=1;r={},i<50,i++,
    For[j=i+1,j<50,j++,
        For[k=j+1,k<50,k++,
            For[l=k+1,l<50,l++,
                z=createsequence[{i,j,k,l},{3#-2&,3#+1&},50];
                If[checkno3Q[z],If[!MemberQ[r,z],AppendTo[r,z]]]
            ]
        ]
    ]
]
takefirstlast@Sort[r,#1[[-1]]<#2[[-1]]||(#1[[-1]]==#2[[-1]]&&#1[[1]]<#2[[1]])&]
```

```
Out[]=
{{1, 4, 10, 13, 28, 31, 37, 40, 82, 85, 91, 94, 109, 112, 118, 121, 244, 247,
  253, 256, 271, 274, 280, 283, 325, 328, 334, 337, 352, 355, 361, 364, 730, 733,
  739, 742, 757, 760, 766, 769, 811, 814, 820, 823, 838, 841, 847, 850, 973, 976}}
```

And finally when we consider sequences of length 100 we have:

In[∘]:=
```
For[i=1;r={},i<50,i++,
    For[j=i+1,j<50,j++,
        For[k=j+1,k<50,k++,
            For[l=k+1,l<50,l++,
                z=createsequence[{i,j,k,l},{3#-2&,3#+1&},100];
                If[checkno3Q[z],If[!MemberQ[r,z],AppendTo[r,z]]]
            ]
        ]
    ]
]
takefirstlast@Sort[r,#1[[-1]]<#2[[-1]]||(#1[[-1]]==#2[[-1]]&&#1[[1]]<#2[[1]])&]
```

Out[∘]=
```
{{1, 4, 10, 13, 28, 31, 37, 40, 82, 85, 91, 94, 109, 112, 118, 121, 244, 247, 253, 256, 271, 274,
  280, 283, 325, 328, 334, 337, 352, 355, 361, 364, 730, 733, 739, 742, 757, 760, 766, 769,
  811, 814, 820, 823, 838, 841, 847, 850, 973, 976, 982, 985, 1000, 1003, 1009, 1012, 1054,
  1057, 1063, 1066, 1081, 1084, 1090, 1093, 2188, 2191, 2197, 2200, 2215, 2218, 2224, 2227,
  2269, 2272, 2278, 2281, 2296, 2299, 2305, 2308, 2431, 2434, 2440, 2443, 2458, 2461,
  2467, 2470, 2512, 2515, 2521, 2524, 2539, 2542, 2548, 2551, 2917, 2920, 2926, 2929}}
```

At this point we proceed with the increment sequence to try to get new informations:

In[∘]:=
```
s3=Differences[s2]
```

Out[∘]=
```
{3, 6, 3, 15, 3, 6, 3, 42, 3, 6, 3, 15, 3, 6, 3, 123, 3, 6, 3, 15, 3, 6, 3, 42,
 3, 6, 3, 15, 3, 6, 3, 366, 3, 6, 3, 15, 3, 6, 3, 42, 3, 6, 3, 15, 3, 6, 3, 123, 3}
```

First thing to note is that all items are multiple of 3 hence we can derive the base sequence:

In[∘]:=
```
s4=s3/3
```

Out[∘]=
```
{1, 2, 1, 5, 1, 2, 1, 14, 1, 2, 1, 5, 1, 2, 1, 41, 1, 2, 1, 5, 1, 2, 1,
 14, 1, 2, 1, 5, 1, 2, 1, 122, 1, 2, 1, 5, 1, 2, 1, 14, 1, 2, 1, 5, 1, 2, 1, 41, 1}
```

Now if you use the inverse operation of the increments sequence, the accumulation that start from 1:

In[∘]:=
```
s5=Accumulate@Prepend[s4,1]
s1[[;;50]]==s5[[;;50]]
```

Out[∘]=
```
{1, 2, 4, 5, 10, 11, 13, 14, 28, 29, 31, 32, 37, 38, 40, 41, 82, 83,
 85, 86, 91, 92, 94, 95, 109, 110, 112, 113, 118, 119, 121, 122, 244, 245, 247,
 248, 253, 254, 256, 257, 271, 272, 274, 275, 280, 281, 283, 284, 325, 326}
```

Out[∘]=
```
True
```

And we are back to the sequence generated with the greedy algorithm. Now we can focus on the increments sequence s2. The construction mechanism of this sequence is the following:

s(0)={1}

s(1)=s(0)(Sum(S(0))+1)s(0)={1}{2}{1}

s(2)=s(1)(Sum(S(1))+1)s(1)={1,2,1}{5}{1,2,1}

s(3)=s(2)(Sum(S(2))+1)s(2)={1,2,1,5,1,2,1}{14}{1,2,1,5,1,2,1}

s(4)=s(3)(Sum(S(3))+1)s(3)={1,2,1,5,1,2,1,14,1,2,1,5,1,2,1}{41}{1,2,1,5,1,2,1,14,1,2,1,5,1,2,1}

...

The “peak” numbers {1,2,5,14,41,122,...} follows the rule:

In[∘]:=
```
FindSequenceFunction[{1,2,5,14,41,122}]
```

Out[∘]=

$$\frac{1}{6}\left(3 + 3^{\#1}\right) \&$$

From the point of view of the increments sequence the constraint of no 3 terms in arithmetic progression can be viewed in term of increments sequence as there are no two subsequences that are ADJACENTS and WITH EQUAL SUM. It is possible to demonstrate that by induction.

For s(1)={1,2,1} it is verified that the “peak” or item greater than all contiguous items belongs to first item or to second item. Suppose that the condition is verified for s(n) now look at s(n+1)=s(n){(3^(n+1)+3)/6}s(n), these two contiguous subsequences are both in a s(n) and the condition is already verified or the “peak” is in one of the subsequences guaranteeing that the sums are different. Now this increments sequence is the bottom solution because the peaks considered are the MINIMUM values required and raising one or more peaks produces another sequence with the required property but same number of items occupy a broader range so those examined is the MOST COMPACT. The recursive method of creation of the increments sequence with its property guarantee that lower increments cannot be found without breaking the rule of the different sum for adjacent subsequences.

Increments sequence s4 can also be found on oeis.org and you can get a couple of functions to calculate all the values and derive every values of s1:

In[∘]:=
```
s[n_Integer?OddQ] := s[n] = 1
s[n_Integer] := s[n] = 3 s[n/2] - 1
valueofs1[n_] := Total[s /@ Range[n - 1]] + 1
s /@ Range[100]
valueofs1 /@ Range[100]
```

Out[∘]=
```
{1, 2, 1, 5, 1, 2, 1, 14, 1, 2, 1, 5, 1, 2, 1, 41, 1, 2, 1, 5, 1, 2, 1,
 14, 1, 2, 1, 5, 1, 2, 1, 122, 1, 2, 1, 5, 1, 2, 1, 14, 1, 2, 1, 5, 1, 2, 1, 41,
 1, 2, 1, 5, 1, 2, 1, 14, 1, 2, 1, 5, 1, 2, 1, 365, 1, 2, 1, 5, 1, 2, 1, 14, 1, 2,
 1, 5, 1, 2, 1, 41, 1, 2, 1, 5, 1, 2, 1, 14, 1, 2, 1, 5, 1, 2, 1, 122, 1, 2, 1, 5}
```

Out[∘]=
```
{1, 2, 4, 5, 10, 11, 13, 14, 28, 29, 31, 32, 37, 38, 40, 41, 82, 83, 85, 86, 91, 92, 94, 95, 109, 110,
 112, 113, 118, 119, 121, 122, 244, 245, 247, 248, 253, 254, 256, 257, 271, 272, 274, 275, 280,
 281, 283, 284, 325, 326, 328, 329, 334, 335, 337, 338, 352, 353, 355, 356, 361, 362, 364, 365,
 730, 731, 733, 734, 739, 740, 742, 743, 757, 758, 760, 761, 766, 767, 769, 770, 811, 812,
 814, 815, 820, 821, 823, 824, 838, 839, 841, 842, 847, 848, 850, 851, 973, 974, 976, 977}
```

Then you can proceed plotting the ratios between valuesofs1[n]/n that exposes fractal properties:

```
ListPlot[{#, N[valueofs1[#] / #]} & /@ Range[100, 100000, 50], PlotRange → All]
```

Out[∘]=