Eli — PS 12 — 2025-03-21

EIWL3 Sections 31 and 32

Parts of Lists

In[1]:= **(*31.1*)Take[IntegerDigits[2^1000], -5]**

Out[1]= {6, 9, 3, 7, 6}

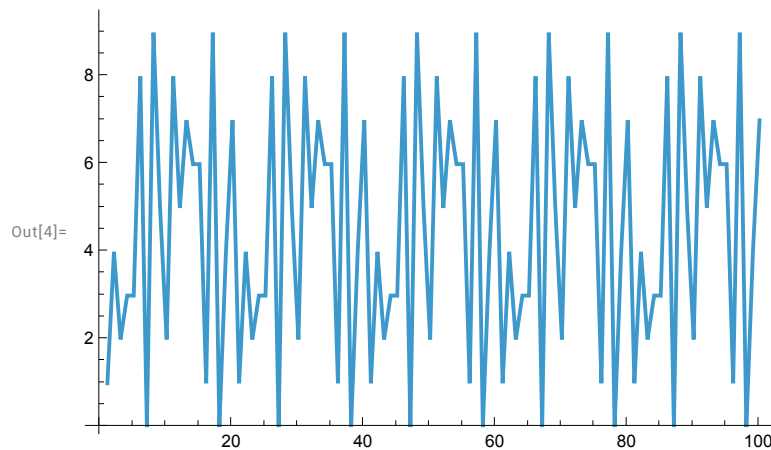In[2]:= **(*31.2*)Alphabet[]〚10 ;; 20〛**

Out[2]= {j, k, l, m, n, o, p, q, r, s, t}

In[3]:= **(*31.3*)GatherBy[Alphabet[], EvenQ[LetterNumber[#]] &]〚2〛**

Out[3]= {b, d, f, h, j, l, n, p, r, t, v, x, z}

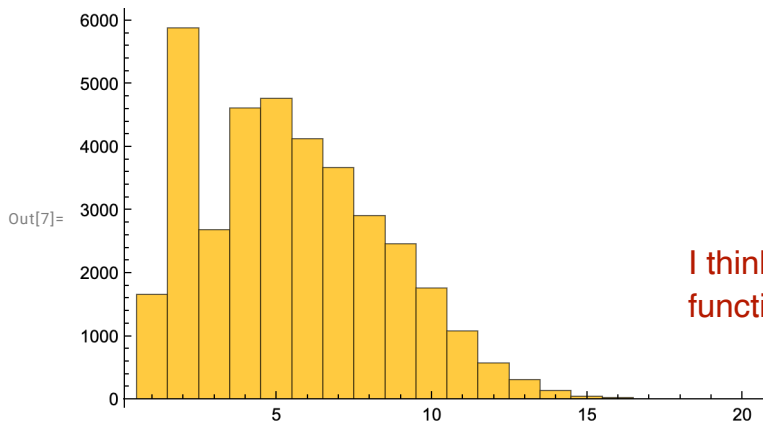In[4]:= **(*31.4*)ListLinePlot[Table[IntegerDigits[12^n], {n, 100}]〚All, -2〛]**

Out[4]=

In[5]:= **(*31.5*)TakeSmallest[Flatten[Table[{n^2, n^3}, {n, 20}]], 10]**

Out[5]= {1, 1, 4, 8, 9, 16, 25, 27, 36, 49}

In[6]:= **(*31.6*)Position[TextWords[WikipediaData["computers"]], "software"]**

Out[6]= {{62}, {6110}, {6204}, {6226}, {6966}, {6988},
  {6991}, {6995}, {7009}, {8212}, {8313}, {8320}, {8328}, {8350}}

In[7]:= `(*31.7*)Histogram[Position[Characters[WordList[]], "e"][[All, 2]] ]`

Out[7]=

I think it doesn't work because your anonymous function isn't Listable.

Testing IntegerQ==True is redundant (but harmless).

In[8]:= `(*31.8*)If[IntegerQ[Sqrt[Position[#]]] == True &, Red, Table[x^3, {x, 100}]]`
`(*Can't figure out why this one isn't working*)`

Out[8]= $\text{If}\left[\text{IntegerQ}\left[\sqrt{\text{Position}[\#1]}\ \right]\ \text{== True \&, Red, Table}\left[x^3, \{x, 100\}\right]\right]$

Looks like you got an alternative working though.

In[9]:= `If[IntegerQ[Sqrt[#]] == True, Red, #] & /@ Table[x^3, {x, 100}]`

Out[9]= {■, 8, 27, ■, 125, 216, 343, 512, ■, 1000, 1331, 1728, 2197, 2744, 3375, ■, 4913, 5832, 6859, 8000, 9261, 10 648, 12 167, 13 824, ■, 17 576, 19 683, 21 952, 24 389, 27 000, 29 791, 32 768, 35 937, 39 304, 42 875, ■, 50 653, 54 872, 59 319, 64 000, 68 921, 74 088, 79 507, 85 184, 91 125, 97 336, 103 823, 110 592, ■, 125 000, 132 651, 140 608, 148 877, 157 464, 166 375, 175 616, 185 193, 195 112, 205 379, 216 000, 226 981, 238 328, 250 047, ■, 274 625, 287 496, 300 763, 314 432, 328 509, 343 000, 357 911, 373 248, 389 017, 405 224, 421 875, 438 976, 456 533, 474 552, 493 039, 512 000, ■, 551 368, 571 787, 592 704, 614 125, 636 056, 658 503, 681 472, 704 969, 729 000, 753 571, 778 688, 804 357, 830 584, 857 375, 884 736, 912 673, 941 192, 970 299, ■}

In[10]:= `(*31.9*)If[IntegerDigits[#][[1]] > 5, Nothing, Prime[#]] & /@ Prime[Range[100]]`

Out[10]=

{3, 5, 11, 31, 41, 59, 67, 83, 109, 127, 157, 179, 191, 211, 241, 277, 547, 563, 587, 599, 617, 709, 739, 773, 797, 859, 877, 919, 967, 991, 1031, 1063, 1087, 1153, 1171, 1201, 1217, 1297, 1409, 1433, 1447, 1471, 1499, 1523, 1597, 1621, 1669, 1723, 1741, 1787, 1823, 1847, 1913, 2027, 2063, 2081, 2099, 2221, 2269, 2341, 2351, 2381, 2417, 2477, 2549, 2609, 2647, 2683, 2719, 2749, 2803, 2897, 2909, 3001, 3019, 3067, 3109, 3169, 3229, 3259, 3299, 3319, 3407, 3469, 3517, 3559, 3593, 3637, 3733, 3761, 3911}

In[11]:= `(*31.10*)`
`Grid[NestList[ReplacePart[#, RandomInteger[Length[#]] → Nothing] &, Range[10], 9]]`

Out[11]=
```
1 2 3 4 5 6 7 8  9 10
1 2 3 4 5 6 7 8 10
1 3 4 5 6 7 8 10
1 3 4 5 6 7 8
1 4 5 6 7 8
1 4 6 7 8
1 4 7 8
1 4 8
```

In[12]:= `(*31.11*)TakeLargestBy[WordList[], StringLength[#] &, 10]`

Out[12]=
```
{electroencephalographic, electroencephalograph,
 buckminsterfullerene, compartmentalization,
 counterrevolutionary, electroencephalogram, internationalization,
 magnetohydrodynamics, uncharacteristically, counterintelligence}
```

In[13]:= `(*31.12*)TakeLargestBy[IntegerName[Range[100]], StringLength[#] &, 5]`

Out[13]=
```
{seventy-three, seventy-seven, seventy-eight, twenty-three, twenty-seven}
```

In[14]:= `(*31.13*)TakeLargestBy[IntegerName[Range[100]],`
` Length[Position[Characters[#], "e"]] &, 5]`

Out[14]=
```
{seventeen, seventy-three, seventy-seven, eleven, eighteen}
```

Patterns

In[15]:= `(*32.1*)Cases[IntegerDigits[Range[1000]], {1, __, 9}]`

Out[15]=
```
{{1, 0, 9}, {1, 1, 9}, {1, 2, 9}, {1, 3, 9},
 {1, 4, 9}, {1, 5, 9}, {1, 6, 9}, {1, 7, 9}, {1, 8, 9}, {1, 9, 9}}
```

In[16]:= `(*32.2*)Cases[IntegerDigits[Range[1000]], {x_, x_, x_}]`

Out[16]=
```
{{1, 1, 1}, {2, 2, 2}, {3, 3, 3}, {4, 4, 4},
 {5, 5, 5}, {6, 6, 6}, {7, 7, 7}, {8, 8, 8}, {9, 9, 9}}
```

In[17]:= `(*32.3*)Cases[Table[IntegerDigits[x^2], {x, 1000}], {9, __, 0 | 1}]`

Out[17]=

```
{{9, 0, 0}, {9, 6, 1}, {9, 8, 0, 1}, {9, 0, 0, 0, 0},
 {9, 0, 6, 0, 1}, {9, 5, 4, 8, 1}, {9, 6, 1, 0, 0}, {9, 6, 7, 2, 1},
 {9, 0, 0, 6, 0, 1}, {9, 0, 2, 5, 0, 0}, {9, 0, 4, 4, 0, 1}, {9, 1, 9, 6, 8, 1},
 {9, 2, 1, 6, 0, 0}, {9, 2, 3, 5, 2, 1}, {9, 3, 8, 9, 6, 1}, {9, 4, 0, 9, 0, 0},
 {9, 4, 2, 8, 4, 1}, {9, 5, 8, 4, 4, 1}, {9, 6, 0, 4, 0, 0}, {9, 6, 2, 3, 6, 1},
 {9, 7, 8, 1, 2, 1}, {9, 8, 0, 1, 0, 0}, {9, 8, 2, 0, 8, 1}, {9, 9, 8, 0, 0, 1}}
```

In[18]:= `(*32.4*)IntegerDigits[Range[100]] /. {9 → Orange, 0 → Gray}`

Out[18]=

```
{{1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {■}, {1, ■}, {1, 1}, {1, 2}, {1, 3},
 {1, 4}, {1, 5}, {1, 6}, {1, 7}, {1, 8}, {1, ■}, {2, ■}, {2, 1}, {2, 2},
 {2, 3}, {2, 4}, {2, 5}, {2, 6}, {2, 7}, {2, 8}, {2, ■}, {3, ■}, {3, 1}, {3, 2},
 {3, 3}, {3, 4}, {3, 5}, {3, 6}, {3, 7}, {3, 8}, {3, ■}, {4, ■}, {4, 1}, {4, 2},
 {4, 3}, {4, 4}, {4, 5}, {4, 6}, {4, 7}, {4, 8}, {4, ■}, {5, ■}, {5, 1}, {5, 2},
 {5, 3}, {5, 4}, {5, 5}, {5, 6}, {5, 7}, {5, 8}, {5, ■}, {6, ■}, {6, 1}, {6, 2},
 {6, 3}, {6, 4}, {6, 5}, {6, 6}, {6, 7}, {6, 8}, {6, ■}, {7, ■}, {7, 1}, {7, 2},
 {7, 3}, {7, 4}, {7, 5}, {7, 6}, {7, 7}, {7, 8}, {7, ■}, {8, ■}, {8, 1}, {8, 2},
 {8, 3}, {8, 4}, {8, 5}, {8, 6}, {8, 7}, {8, 8}, {8, ■}, {■, ■}, {■, 1},
 {■, 2}, {■, 3}, {■, 4}, {■, 5}, {■, 6}, {■, 7}, {■, 8}, {■, ■}, {1, ■, ■}}
```

In[19]:= `(*32.5*)IntegerDigits[ 2^1000] /. 0 → Red`

Out[19]=

```
{1, ■, 7, 1, 5, ■, 8, 6, ■, 7, 1, 8, 6, 2, 6, 7, 3, 2, ■, 9, 4, 8, 4, 2, 5, ■, 4, 9,
 ■, 6, ■, ■, ■, 1, 8, 1, ■, 5, 6, 1, 4, ■, 4, 8, 1, 1, 7, ■, 5, 5, 3, 3, 6, ■,
 7, 4, 4, 3, 7, 5, ■, 3, 8, 8, 3, 7, ■, 3, 5, 1, ■, 5, 1, 1, 2, 4, 9, 3, 6, 1, 2,
 2, 4, 9, 3, 1, 9, 8, 3, 7, 8, 8, 1, 5, 6, 9, 5, 8, 5, 8, 1, 2, 7, 5, 9, 4, 6, 7, 2,
 9, 1, 7, 5, 5, 3, 1, 4, 6, 8, 2, 5, 1, 8, 7, 1, 4, 5, 2, 8, 5, 6, 9, 2, 3, 1, 4, ■,
 4, 3, 5, 9, 8, 4, 5, 7, 7, 5, 7, 4, 6, 9, 8, 5, 7, 4, 8, ■, 3, 9, 3, 4, 5, 6, 7, 7,
 7, 4, 8, 2, 4, 2, 3, ■, 9, 8, 5, 4, 2, 1, ■, 7, 4, 6, ■, 5, ■, 6, 2, 3, 7, 1, 1,
 4, 1, 8, 7, 7, 9, 5, 4, 1, 8, 2, 1, 5, 3, ■, 4, 6, 4, 7, 4, 9, 8, 3, 5, 8, 1, 9, 4,
 1, 2, 6, 7, 3, 9, 8, 7, 6, 7, 5, 5, 9, 1, 6, 5, 5, 4, 3, 9, 4, 6, ■, 7, 7, ■, 6,
 2, 9, 1, 4, 5, 7, 1, 1, 9, 6, 4, 7, 7, 6, 8, 6, 5, 4, 2, 1, 6, 7, 6, 6, ■, 4, 2, 9,
 8, 3, 1, 6, 5, 2, 6, 2, 4, 3, 8, 6, 8, 3, 7, 2, ■, 5, 6, 6, 8, ■, 6, 9, 3, 7, 6}
```

In[20]:= `(*32.6*)Characters["The Wolfram Language"] /. "a" | "e" | "i" | "o" | "u" → Nothing`

Out[20]=

```
{T, h, , W, l, f, r, m, , L, n, g, g}
```

In[21]:= `(*32.7*)Cases[IntegerDigits[2^1000], 0 | 1]`

Out[21]=

```
{1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0}
```

In[22]:= `(*32.8*)Cases[IntegerDigits[Range[100, 999]], {x_, __, x_}]`

Out[22]=

```
{{1, 0, 1}, {1, 1, 1}, {1, 2, 1}, {1, 3, 1}, {1, 4, 1}, {1, 5, 1}, {1, 6, 1}, {1, 7, 1},
 {1, 8, 1}, {1, 9, 1}, {2, 0, 2}, {2, 1, 2}, {2, 2, 2}, {2, 3, 2}, {2, 4, 2}, {2, 5, 2},
 {2, 6, 2}, {2, 7, 2}, {2, 8, 2}, {2, 9, 2}, {3, 0, 3}, {3, 1, 3}, {3, 2, 3}, {3, 3, 3},
 {3, 4, 3}, {3, 5, 3}, {3, 6, 3}, {3, 7, 3}, {3, 8, 3}, {3, 9, 3}, {4, 0, 4}, {4, 1, 4},
 {4, 2, 4}, {4, 3, 4}, {4, 4, 4}, {4, 5, 4}, {4, 6, 4}, {4, 7, 4}, {4, 8, 4}, {4, 9, 4},
 {5, 0, 5}, {5, 1, 5}, {5, 2, 5}, {5, 3, 5}, {5, 4, 5}, {5, 5, 5}, {5, 6, 5}, {5, 7, 5},
 {5, 8, 5}, {5, 9, 5}, {6, 0, 6}, {6, 1, 6}, {6, 2, 6}, {6, 3, 6}, {6, 4, 6},
 {6, 5, 6}, {6, 6, 6}, {6, 7, 6}, {6, 8, 6}, {6, 9, 6}, {7, 0, 7}, {7, 1, 7},
 {7, 2, 7}, {7, 3, 7}, {7, 4, 7}, {7, 5, 7}, {7, 6, 7}, {7, 7, 7}, {7, 8, 7},
 {7, 9, 7}, {8, 0, 8}, {8, 1, 8}, {8, 2, 8}, {8, 3, 8}, {8, 4, 8}, {8, 5, 8},
 {8, 6, 8}, {8, 7, 8}, {8, 8, 8}, {8, 9, 8}, {9, 0, 9}, {9, 1, 9}, {9, 2, 9},
 {9, 3, 9}, {9, 4, 9}, {9, 5, 9}, {9, 6, 9}, {9, 7, 9}, {9, 8, 9}, {9, 9, 9}}
```