day10

December 12, 2020

1 Day 10 Tree or List

Ok, so my instinct with this one was to create a tree structure.

If we had a tree object whereby every possible adaptor was represented, we can apply lots of neat graph algorithms across the tree to find a given goal, find the distance between nodes, walk the tree and so on.

But the rule that you want to use the most number of adapters makes me think that the easiest way to approach this is simply to put them into a list, and step over the list counting the number of +1 and +3 differences.

That feels way too easy, and I strongly suspect that the follow up question will be to search to find the shortest chain of adapters that gets to the goal, which will be much easier if we build a tree structure, so we might come back to that idea.

(I'll admit that building simple data structures in python is something that I've tried and failed at before. Because I first learn to program in TurboPascal and C/C++, pointers and references are somewhat native in my experience of building data structures. Python doesn't really have that, so if we have to build a tree out of dictionaries or named-tuples or something, I'll have to look up the best way to do that)

[1]:

```
import ipytest
ipytest.autoconfig()
```

Ok, lets get started, I think list form is easiest. Taking the worked example, we have `16,10,15,5,1,11,7,19,6,12,4` which if we sort it would be `1,4,5,6,7,10,11,12,15,16,19`. If we add the wall and our adapter (always 0 and n+3 respectively) we end up with: `0,1,4,5,6,7,10,11,12,15,16,19,22`.

So all we then do is iterate the list and count from previous to now, and if it's +1 or +3, we add to the count and then multiply them.

[7]:

```
test_data = [16,10,15,5,1,11,7,19,6,12,4]

def count_differences(data):
    prev = 0
    count1 = 0
    count3 = 0
    for item in sorted(data):
        if item - prev == 1:
            count1 += 1
        if item - prev == 3:
            count3 += 1
        prev = item
    count3 += 1
    return count1,count3

assert count_differences(test_data) == (7,5)
```

Ok, that works, lets try on the bigger dataset

[9]:

```
test_data = [int(i.strip()) for i in """28
33
18
42
31
14
46
20
48
47
24
23
49
45
19
38
39
11
1
32
25
35
8
17
7
9
4
2
34
10
3""".split("\n")]

assert count_differences(test_data) == (22,10)
```

Yup, that works, real data

[12]:

```
data = [int(i.strip()) for i in open("day10.txt").readlines()]
c1,c3 = count_differences(data)
print(f"{c1}*{c3}={c1*c3}")
```

```
76*32=2432
```