day1

December 6, 2020

Day 1, can we find pairs of numbers that add up to a total?

[1]:
```
import pytest
import ipytest
ipytest.autoconfig()
```

[2]:
```
import itertools

def find_pair(l, target):
    for (a,b) in itertools.combinations(l,2):
        if a+b == target:
            return (a,b)
    return (0,0)
```

Let's test some of that

[3]:
```
assert (1,6) == find_pair([1,2,3,4,5,6], 7)
assert (1721,299) == find_pair([1721,979,366,299,675,1456], 2020)
```

Ok, so we can find pairs, let's find the pair from the actual input. This requires reading all the numbers from the data

[4]:
```
data = [int(l) for l in open("day1.txt").readlines()]
```

[5]:
```
find_pair(data, 2020)
```

[5]:
```
(1073, 947)
```

0.1 Day 2

Great, so lets move onto day 2, now we need to find triples, rather than pairs. Should be easy, just change find_pair into find_triple and everything else is the same

[6]:
```
def find_triple(l, target):
    for (a,b,c) in itertools.combinations(l,3):
        if a+b+c == target:
            return (a,b,c)
    return (0,0,0)
```

[7]:
```
assert (1,2,6) ==  find_triple([1,2,3,4,5,6], 9)
assert (979, 366, 675) == find_triple([1721,979,366,299,675,1456], 2020)
```

Great, that works, so lets now run it against the real data

[8]:
```
find_triple(data, 2020)
```

[8]:
```
(911, 618, 491)
```