

## Brute Force Enumeration

The idea of brute force enumeration is to simply enumerate over every possible subset, and check if the sum is equal to the target sum.

TARGET = 40541043 mod 100000000

NUM\_POSSIBLE\_SUBSETS =  $\binom{24}{12} = 2704156$

If we precompute all the sums in this fashion:

```
THE_DICTIONARY = {
4285467: [(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13)],
26343272: [(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 16)],
31536986: [(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 19)],
33268004: [(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15)],
34176022: [(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)],
40469194: [(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 18)],
41519580: [(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 17)],
60375227: [(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14)],
70644896: [(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20)],
72288846: [(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12)],
... (2651034 entries omitted) ...
23259935: [(11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23)],
24903885: [(11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23)],
35173554: [(11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23)],
54029201: [(11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23)],
55079587: [(11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23)],
61372759: [(12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23)],
62280777: [(11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23)],
64011795: [(11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 23)],
69205509: [(11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23)],
91263314: [(11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23)]
}
```

then we can simply look up the answer based on dictionary index

SOLUTIONS = [(0, 1, 5, 6, 9, 10, 11, 13, 14, 15, 16, 22)]

### Notes:

- early exit (rather than full enumeration) would have entailed a little less computation, but we should not expect to find the answer if we enumerate a significantly smaller set.
- the solutions are a list, because in theory there could be more than one answer, and this becomes more likely as the modulus is decreased