



Master algorithms together on [Binary Search](#)! Create a room, invite your friends, and race to finish the problems.

Daily Coding Problem

[Blog](#)

Daily Coding Problem #186

Problem

This problem was asked by Microsoft.

Given an array of positive integers, divide the array into two subsets such that the difference between the sum of the subsets is as small as possible.

For example, given `[5, 10, 15, 20, 25]`, return the sets `{10, 25}` and `{5, 15, 20}`, which has a difference of 5, which is the smallest possible difference.

Solution

One way to solve this problem would to be enumerate over all possible pairs of subsets and find the smallest difference between each pair:

```
from math import inf
from itertools import combinations

def two_subsets(nums):
    smallest_diff = inf
```

```

result = None
for subset1, subset2 in subset_pairs(nums):
    diff = abs(sum(subset1) - sum(subset2))
    if diff < smallest_diff:
        smallest_diff = diff
        result = (subset1, subset2)
return result

def subset_pairs(nums):
    n = len(nums)
    for r in range(n + 1):
        for indices in combinations(range(n), r):
            subset1 = [nums[i] for i in indices]
            subset2 = [nums[i] for i in set(range(n)) - set(indices)]
            yield subset1, subset2

```

This takes $O(2^n)$ time, since there are 2^n subsets in a set of size n .

© Daily Coding Problem 2019

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)