Minimum subset sum difference

Problem link - https://practice.geeksforgeeks.org/problems/minimum-sum-partition3317/1

Flow

1. **Problem statement**
2. **Similarity**
3. **Solve using previous concept**

Problem Statement - Given an array arr of size n containing non-negative integers, the task is to divide it into two sets S1 and S2 such that the absolute difference between their sums is minimum and find the minimum difference

Example 1:

```
Input: N = 4, arr[] = {1, 6, 11, 5}
Output: 1
Explanation:
Subset1 = {1, 5, 6}, sum of Subset 1 = 12
Subset2 = {11}, sum of Subset2 = 11
```

Example 2:

```
Input: N = 2, arr[] = {1, 4}
Output: 3
Explanation:
Subset1 = {1}, sum of Subset1 = 1
Subset2 = {4}, sum of Subset2 = 4
```

Array - { } iska do paration kar deta hai partition 1 and partition 2 and let say partition 1 ka sum s1 hai and partition 2 ka sum s2.

To , we have to find the minimum difference between both the sum s1 and s2 , That is (s1 - s2 = min).

Similarity

This problem is similar to the equal sum partition problem. Where we find
The s1-s2 = 0 ,or s1 == s2 and one more thing this equal sum partition question based on the subset sum problem. And here in this question we have to find the s1-s2 == min

How to approach this question

Here, we don’t know the exact value of s1 and s2 , but we know that s1 and s2 lie in some range that is s1 , s2 (0 , range) where range = sum of all the numbers.

Let’s suppose array - {1 ,2 , 7};
Now we have a range from 0 to 10 , can all the numbers between 0 to 10 be candidates for s1 and s2? Let's find out 0 ,1 ,2 ,3 ,4, 5,6 ,7,8 ,9 , 10.
Can 0 is candidate Yes , because empty subset is { }can be value of s1
Can 1 is a candidate , Yes because {1} one of the subsets from the given array is 1.
Can 4 is a candidate , No because no subset can form with sum 4.
Can 9 is a candidate , yes because {2 , 7} can form with sum 9.

So we filter some values from the range. Now we got 1 , 2 , 3 , 7 ,8 ,9 , 10. These numbers are candidates for s1 and s2.

Ans one more thing here we consider s1 as min and s2 , so the range for s1 is half the range we got **1 ,2 ,3 |** 7, 8, 9 10

Now we know that s1 + s2 = Range
If we find the value of s1 , s2 = range - s1. And in the question asking for s1-s2=min , so , putting the value of s2 in this (range - s1 - s1) that becomes min = range - 2s1.

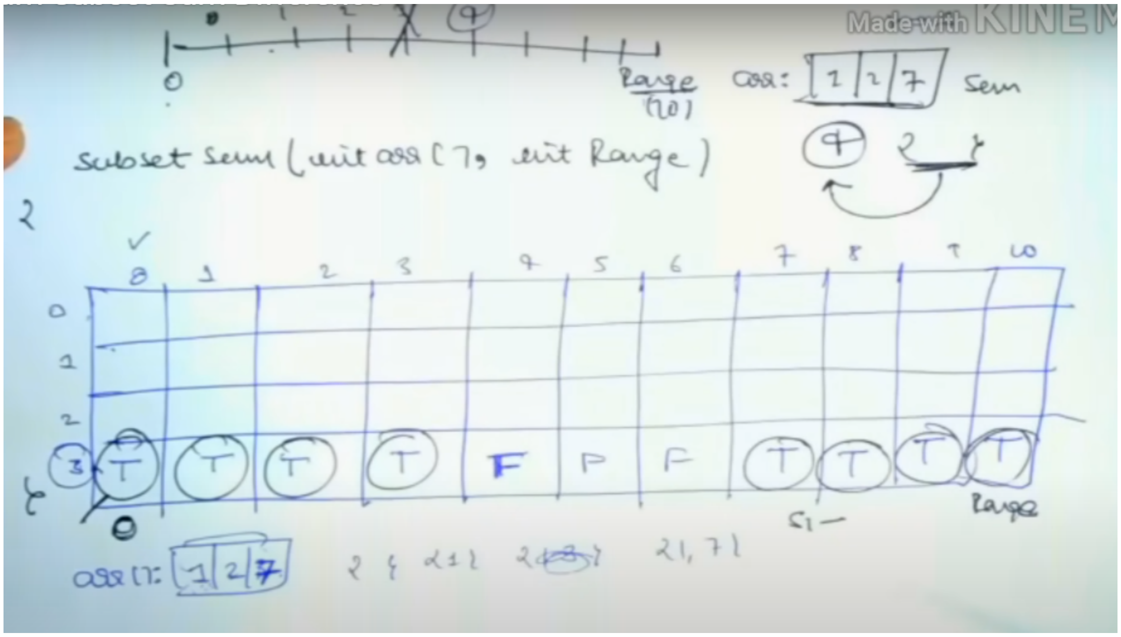
Now the value of s1 = 1 , 2, 3 and min = range - 2s1 now if you put s1 = 1 we get min = 10-1 = 9 , s1 = 2 we get 10 - 4 = 6 , s1 = 3 we get 10 - 6 = 4.
So the minimum value of s1-s2 = 4.

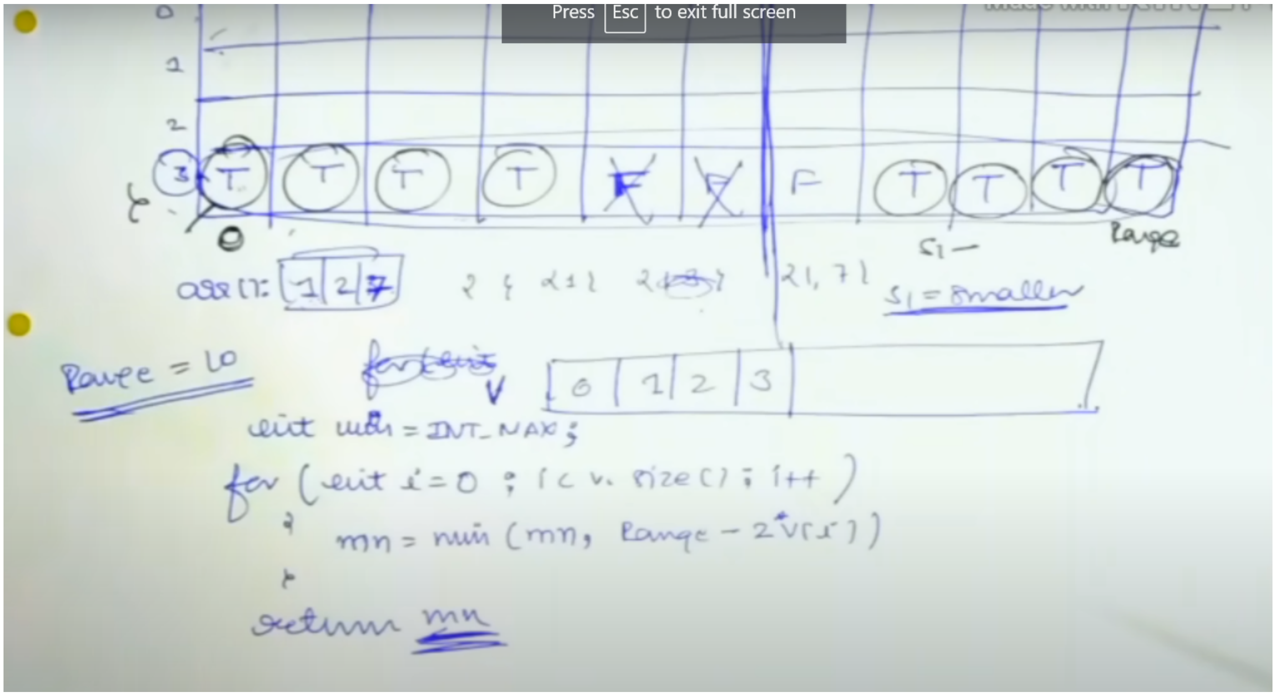
// Code explanation

If you remember that the subset sum the last row in 2d, the boolean vector denotes that if the subset exists or not with the sum if true then exists else false.

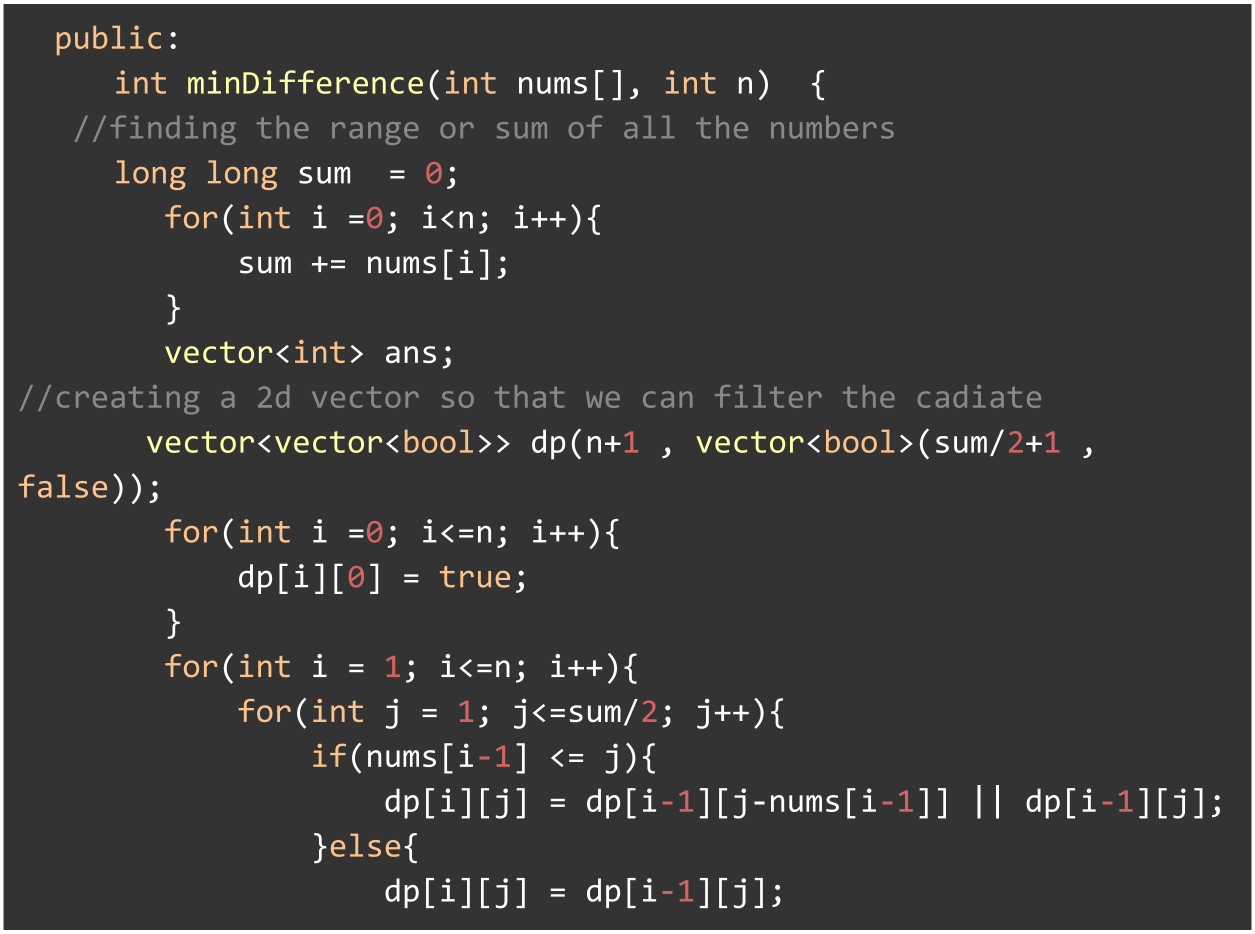
So we check the last row of that matrix and check if true then put in a vector of int.

And for calculating minimum we already derive the formula that is range-2s1.





```
int mn = INT_MAX;
for (int i=0 ; i< v.size() ; i++ )
{
    mn = min (mn, Range - 2*v[i])
}
return mn
```



```
  public:
    int minDifference(int nums[], int n)  {
   //finding the range or sum of all the numbers
    long long sum  = 0;
       for(int i =0; i<n; i++){
           sum += nums[i];
       }
       vector<int> ans;
//creating a 2d vector so that we can filter the cadiate
       vector<vector<bool>> dp(n+1 , vector<bool>(sum/2+1 ,
false));
       for(int i =0; i<=n; i++){
           dp[i][0] = true;
       }
       for(int i = 1; i<=n; i++){
           for(int j = 1; j<=sum/2; j++){
               if(nums[i-1] <= j){
                   dp[i][j] = dp[i-1][j-nums[i-1]] || dp[i-1][j];
               }else{
                   dp[i][j] = dp[i-1][j];
```

```
                }
            }
        }
//here filtering the candidate for s1
        for(int i =0; i<=sum/2; i++){
            if(dp[n][i] == true){
                ans.push_back(i);
            }
        }
//here finding the min value
         int mn = INT_MAX;
        for(int i =0; i<ans.size(); i++){
            int a =  sum - (2 * ans[i]);
           mn = min(mn , a);
        }
        return mn;
      }
};
```

If you still have donuts we can check out the videos -

10 Minimum Subset Sum Difference