

Computative programming

Assignment 3-5

Hallticket no:2303A510A6

Batch no:02

Assignment -1 Subset Sum Using Meet-in-the-Middle

Problem Statement:

You are given an array of N integers and a target sum S.

Using the Meet-in-the-Middle technique, determine whether there exists a subset whose sum is exactly equal to S.

Input Format:

- The first line contains an integer T, the number of test cases.

For each test case:

- The first line contains two integers N and S.
- The second line contains N integers.

Output Format:

For each test case, print YES if such a subset exists, otherwise print NO.

Constraints:

- $1 \leq T \leq 20$
- $1 \leq N \leq 40$
- $-10^9 \leq A[i] \leq 10^9$

Sample input

2

4 9

3 1 5 7

5 10

2 4 6 8 1

Sample output

YES

NO

CODE (java):

```
import java.util.*;
public class Ass {
    static void subsetSums(int[] arr, int index, long sum, ArrayList<Long> list) {
        if (index == arr.length) {
            list.add(sum);
            return;
        }
        subsetSums(arr, index + 1, sum, list);      // exclude
        subsetSums(arr, index + 1, sum + arr[index], list); // include
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = sc.nextInt();
        while (T-- > 0) {
            int N = sc.nextInt();
            long S = sc.nextLong();
            int[] arr = new int[N];
            for (int i = 0; i < N; i++) {
                arr[i] = sc.nextInt();
            }
            int mid = N / 2;
            int[] left = Arrays.copyOfRange(arr, 0, mid);
            int[] right = Arrays.copyOfRange(arr, mid, N);
            ArrayList<Long> leftSums = new ArrayList<>();
            ArrayList<Long> rightSums = new ArrayList<>();
            subsetSums(left, 0, 0, leftSums);
            subsetSums(right, 0, 0, rightSums);
            Collections.sort(rightSums);
            boolean found = false;
            for (long x : leftSums) {
                if (Collections.binarySearch(rightSums, S - x) >= 0) {
                    found = true;
                    break;
                }
            }
            if (found)
                System.out.println("YES");
            else
                System.out.println("NO");
        }
        sc.close();
    }
}
```

The screenshot shows an IDE interface with a project tree on the left and a code editor on the right. The project tree includes a 'day 3 java' folder containing 'src' and 'day 3.java.iml'. Under 'src', there is a 'java codes' folder containing several Java files: Animal, Arr.java, Array, Arraye, Ass, Ass.java, Bike, Bill, book.class, Car, child, creditImpl, Deq, Deq.java, Dog, EmailImpl, Execp, Harshadnumber, Harshadnumber.java, and Lorry. The code editor displays Java code for a 'subsetSums' method and a 'main' method. The 'subsetSums' method takes an int[] arr, an int index, a long sum, and an ArrayList<Long> list. It adds the current sum to the list if the index is at the end of the array. It then calls itself recursively with the next index and sum, excluding the current element, and including the current element. The 'main' method reads input from System.in, initializes variables, and calls the 'subsetSums' method to find if a target sum exists in the subset.

```

static void subsetSums(int[] arr, int index, long sum, ArrayList<Long> list) { no usages
    if (index == arr.length) {
        list.add(sum);
        return;
    }
    subsetSums(arr, index + 1, sum, list); // exclude
    subsetSums(arr, index + 1, sum + arr[index], list); // include
}
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int T = sc.nextInt();
    while (T-- > 0) {
        int N = sc.nextInt();
        long S = sc.nextLong();
        int[] arr = new int[N];
        for (int i = 0; i < N; i++) {
            arr[i] = sc.nextInt();
        }
        int mid = N / 2;
        int[] left = Arrays.copyOfRange(arr, 0, mid);
        int[] right = Arrays.copyOfRange(arr, mid, N);
        ArrayList<Long> leftSums = new ArrayList<>();
        ArrayList<Long> rightSums = new ArrayList<>();
        subsetSums(left, 0, 0, leftSums);
        subsetSums(right, 0, 0, rightSums);
        Collections.sort(rightSums);
        boolean found = false;
        for (long x : leftSums) {
            if (Collections.binarySearch(rightSums, S - x) >= 0) {
                found = true;
                break;
            }
        }
        if (long x : leftSums) {
            if (Collections.binarySearch(rightSums, S - x) >= 0) {
                found = true;
                break;
            }
        }
        if (found)
            System.out.println("YES");
        else
            System.out.println("NO");
        sc.close();
    }
}

```

This screenshot shows the same IDE environment as above, but with additional terminal output at the bottom. The terminal window shows the following session:

```

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\shyam\OneDrive\Desktop\java codes> & javac ass.java
ass.java:3: error: class Main is public, should be declared in a file named Main.java
public class Main {
^
1 error
PS C:\Users\shyam\OneDrive\Desktop\java codes> & javac Ass.java
PS C:\Users\shyam\OneDrive\Desktop\java codes> java Ass
2
4 9
3 1 5 7
YES
5 10
2 4 6 8 1
YES
PS C:\Users\shyam\OneDrive\Desktop\java codes>

```

Code (python):

```

def subset_sums(arr):

    sums = [0]

    for x in arr:

```

```

new_sums = []
for s in sums:
    new_sums.append(s + x)
sums = sums + new_sums
return sums

T = int(input())
for _ in range(T):
    N, S = map(int, input().split())
    arr = list(map(int, input().split()))
    mid = N // 2
    left = arr[:mid]
    right = arr[mid:]
    left_sums = subset_sums(left)
    right_sums = subset_sums(right)
    right_set = set(right_sums)
    found = False
    for x in left_sums:
        if S - x in right_set:
            found = True
            break
    if found:
        print("YES")
    else:
        print("NO")

```

The screenshot shows a web-based IDE interface for Python. The main area displays a Python script named 'main.py' with the following code:

```
1 def subset_sums(arr):
2     sums = [0]
3     for x in arr:
4         new_sums = []
5         for s in sums:
6             new_sums.append(s + x)
7             sums = sums + new_sums
8     return sums
9
10 T = int(input())
11 for _ in range(T):
12     N, S = map(int, input().split())
13     arr = list(map(int, input().split()))
14
15     mid = N // 2
16     left = arr[:mid]
17     right = arr[mid:]
18
19     left_sums = subset_sums(left)
20     right_sums = subset_sums(right)
21
22     right_set = set(right_sums)
23
24     found = False
25     for x in left_sums:
26         if S - x in right_set:
27             found = True
28             break
29
30     if found:
31         print("YES")
32     else:
33         print("NO")
```

Below the code, there is an 'input' field containing the following test cases:

```
0 1 5 7
YES
5 10
2 4 6 8 1
YES
```

The status bar at the bottom of the browser window shows the URL `onlinedb.com/online.python.compiler`.