PROBLEM 2 *European Summer Vacation* [20 points]

You and a friend are heading to Europe for the summer (hopefully!). You have completed your packing list, and combined, you need to bring n items in total, with the weights of the items given by $W = (w_1, \ldots, w_n)$. Your goal is to divide the items between your suitcases such that the difference in weight between them is at most t. We formally define the problem below:

> SUITCASE: Given a sequence of non-negative weights $W = (w_1, \ldots, w_n)$ and a target weight difference t, can you divide the items among you and your friend such that the weight difference between suitcases is at most t?

Show that the SUITCASE problem defined above is NP-complete (namely, you should show that SUITCASE $\in$ NP and that SUITCASE is NP-hard). For this problem, you may use the fact that the SUBSETSUM problem is NP-complete:

> SUBSETSUM: Given a sequence of non-negative integers $a_1, \ldots, a_n$ and a target value t, does there exist a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} a_i = t$?

Answer

To prove that a Problem is NP – complete, we must recall that the Problem needs to be both NP – hard and NP. So, we need to show that the problem is in NP and in NP – hard.

In our case, to prove that the Suitcase problem is NP – complete, we can use the fact that the Subset Sum problem is NP – complete.

Abridged statement: Given a set of numbers, determine the sets A and B such that `|SUM(A) - SUM(B)| <= m` or the difference between sum of elements in A with those in B is at most m.

1. **Showing that the problem is in NP.**
   - We can set the certificate be (A, B). The certificate can be verified in linear time by summing the elements in the two sets and comparing if their difference is at most t.
2. **Showing that the problem is in NP – hard.**
   - We can show that Suitcase problem is in NP – hard by trying to show that the Subset Sum problem, which is NP – hard can be simplified to the Suitcase problem.
3. **Showing that Subset Sum can be reduced to the Suitcase problem.**
   - Subset sum is defined: given a set of S of integers and a target sum t, find a subset Y of S such that sum(Y) = t. Now, we denote that S' = S U { s – 2t } into the Suitcase Problem. We accept it iff the Suitcase problem accepts. – this reduction works in linear time.

<u>Using reduction from Subset Sum</u>

Considering the Subset Sum problem, we can prove the existence of C in S with a sum c ⇔ S' = S U { s – 2c } can be divided into (A, B) [subsets that Suitcase requires]. Such group S' can be constructed in Linear time, by summing all elements in S then adding s – 2c to S. Hence, we prove the reduction in polynomial time.

Let's suppose that C exists and let S – C = D. Also denoting d as the sum of elements in D, and C' = C U s – 2c and c' the sum of C'.

We have:

d = s – c -> o – c = s – 2c -> c' = c + (s – 2c) -> c' = s – c -> c' = d

So, difference between of sums of C' and sums of D are equal, this proves that

So, we now suppose a sum with at most difference m partitioning (B,B') of S' = S U { s – 2c } exists. Then we might have the following: (s + (s-2c))/2 = s – c. Considering the partition containing the element { s – 2c } to be B, denote A = B – { s – 2c} + m (at most m). We can see that S' – S = { s – 2c}.

So, we reduced the Subset Sum problem to our Suitcase problem, thus Suitcase in NP – complete.