Partition NP Complete

Luke Erlewein
CIS 263 Summer 2025

What is the Partition Problem?

The Partition problem asks, for a list of only positive integers, does there exist two sub lists that the sum of equals the sum of the original list.

Basically can you split the original list into two smaller lists that equal each other.

Example:
If you have a list of {3, 2, 4, 3, 1, 1} which has a sum of 14. You can split it into two lists {3, 4} and {1, 1, 2, 3} which both equal 7. This would be an example of a successful partition.

Reducing the Problem

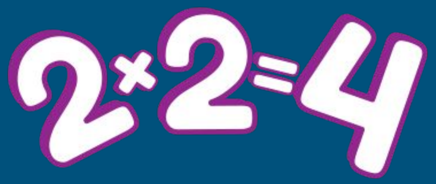
To prove that the Partition problem is a problem of complexity NP, we need to reduce from a known NP hard problem to the Set Partition Problem.

The problem it will be reduced from is the Subset Sum problem.

Reducing the Problem

The first step is to understand what the Subset Sum Problem is.

If you have a set of numbers S, you want to reach a target number T. Are you able to use numbers from the set of numbers S to create a sum that equals T?



Reducing the Problem

The subset problem provides an input as a set of numbers S, and a target sum of t. The problem wants to find a solution of a subset T belonging to S with the same sum as t. s will be the sum of all the numbers in S.

So we can put S’ = S U {s - 2t} into the Set Partition problem

Reducing the Problem

In order to prove that the Partition problem is a true derivation of the Subset sum problem, we can follow two things.

The first is that we can partition the original set of numbers into two subsets that are equal to each other.

With a set of numbers, T, and its total sum t. Then the remaining elements in the set S can be described as having the sum o = s-t. This means that the original set is equal to T' = T U (s - 2t) which has a sum t'.

Reducing the Problem

This means that the following is true.

o = s-t

o - t = s - 2t (This is the difference between O and T)

t’ = t + (s - 2t)

t’ = s - t

t’ = o (The sum of T’ and O are equal to each other.)

Reducing the Problem

With the previous observations, it can be concluded that the original set S can be divided into two subsets of the sum which is (s-t) for each of them. This means they are equal and the Set Partition problem has been solved.

Reducing the Problem

Now we can consider the opposite scenario where we have two partitions that have an equal sum. (A, A’) these partitions are of the bigger set S’ = S U {s-2t}

The sum of each partition is given by the following equation.

$$a = \frac{s+(s-2*t)}{2} = (s-t)$$

Reducing the Problem

Now let's say the partition containing the element {s - 2t} is A' so that A = A' - {s - 2t}.
This means the sum of all all elements in A is the following

A = s - t - {s- 2t}

A = t

So S' - S = {s - 2t}. This means that A is a subset of S with sum equal to t.
Therefore, the Subset Sum Problem is confirmed using the same stuff.

Reducing the Problem

With the Subset Sum problem being able to be derived from the Set Partition problem we can confirm that the Set partition problem has the same complexity as the Subset Sum Problem which is NP.

Sources

Partition Problem - https://en.wikipedia.org/wiki/Partition_problem

Reducing the Problem - https://www.geeksforgeeks.org/dsa/set-partition-is-np-complete/