P7. Underwater Cable

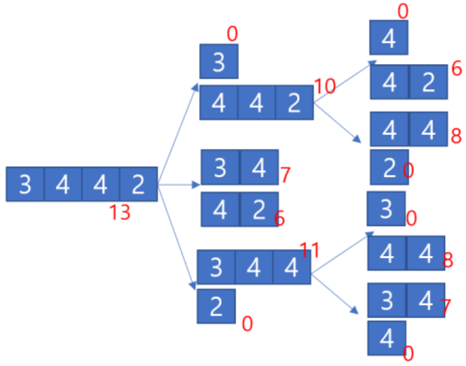
[[홍원표 21701065]]

문제 분석

There are N pieces, where 3 <= N <= 500, each of their length less than 10000. The goal is to add all pieces into one, while adding is only allowed between two pieces. The returned value must be the minimum total sum of additions that had taken place to create a single piece.

문제 풀이

The simplest way of solving this is by brute force: calculate all possible splits of the input array by recursions. For example, like the picture on the right side, if the initial value is {3, 4, 4, 2} of size n=4, there will be n-1=3 different ways to split the array. Recursion of these splits can occur until the size of the subarrays are 1 or 2. If 1, return 0 because there is no ‘merging’, and if 2, return the sum of two elements because they must merge with themselves. Adding all recursions back to top will result 13+10+6, 13+10+8, 13+7+6, 13+11+8, 13+11+7 → 29, 31, 26, 32, 31. Therefore, the minimum of these 5 possibilities is 26, which is the answer.
However, reduction of steps is possible if we ‘remember’ the sum of the same subarrays.
In this example, {4, 4} occurs twice. If we memorize 6 for {4, 4} using Dynamic Programming, we don’t have to do the same process twice.



문제 풀이 분석

The total DP memorization will become [0,0], [0,1], …, [0,n-1], [1,1], [1,2], … , [1, n-1], … … [n-1, n-1]. This is $O(n^2)$.
Also, there are n-1 possibility of splits of size n. Therefore, **the time complexity is $O(n^3)$.**
The space complexity is $O(n^2)$ because it requires 2 dimensional array for DP.

Discussion

My first approach was to find the minimum difference of left-sum and right-sum and split according to that minimum, and then recursion on the splitted arrays. This way actually works with the two examples given by the professor. Then, why is this a wrong answer?