



Skyscraper (Solution)

Subtask 1 [5 points]

Brute force. Try every permutation. $O(N!)$

Subtask 2 [15 points]

Deciding each of the numbers from the leftmost position.

This can be solved by using bit DP. We calculate $dp[S][j][i]$, where S is the set of used number in the array, j is the last chosen number, and i is the current total sum of all differences of two adjacent numbers.

From each DP state, you can choose one of the unused number, so the total time complexity will be $O(2^N \cdot L \cdot N^2)$.

Subtask 3 [80 points]

In this Subtask, you should decide the relative positions of each number from the smallest value.

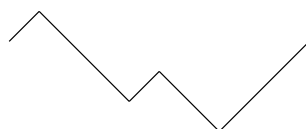


Figure 1

In other words, you should choose positions from below (see Figure 1).

This can be solved by DP. We consider $dp[i][j][k][l]$, where i means we calculate the number of ways to put the i -th smallest element, j is the number of connected components, k is the current total cost, and l is the number of elements already used for the leftmost or the rightmost edges. The treatment of connected components is tricky. When you add the i -th smallest element, either (1) it is disconnected with other connected components, (2) it is connected with one connected component only, or (3) it connects two connected components.

The additional cost on each phase depends on the value $A_i - A_{i-1}$ (after sorting $A_1, A_2, A_3, \dots, A_N$), and the number of connected components.

For further description about this DP, see “AppleTrees” from Member SRM 489¹, “Seatfriends” from SRM 625², or “SumOverPermutations” from SRM 666³.

¹https://community.topcoder.com/stat?c=problem_statement&pm=11213

²https://community.topcoder.com/stat?c=problem_statement&pm=12909

³https://community.topcoder.com/stat?c=problem_statement&pm=13764