AIR6003 OJ Report1

Yonglin DENG

Date: October 27, 2023

1 Re-sorting the Array with Scores

1.1 Ideas of solutions

First calculate the score for each integer, then sort in ascending order and select the integer corresponding to the i-th score.

1.2 Time and space complexity

- time complexity: $O(N \lg N)$, the time complexity of calculating the score for each integer is $O(N \lg N)$, and the complexity of sorting is also the same
- space complexity: $O(N)$, a one-dimensional array records integers and corresponding scores

2 Gas station planning

2.1 Ideas of solutions

This question is to minimize the sum of distances between every village and its neareast gas station. Becasue we can obtain the minimum value of the original problem through the minimum value of the sub-problem and the problem has overlapping subproblems, we use dynamic programming to solve this problem.

There are two points that must be pay attention to:

- If there are more gas stations than villages, the sum of distances equals to 0.
- If there is only one gas station, the gas station should be placed in the middle so that the sum is smallest.

The above two points also correspond to the base cases of dynamic programming.

The transfer equation of dynamic programming is

$$dp(n,k) = \min_{i=1,\ldots,n} \left(\sum_{i':i<i' \leq n} d\left(\text{city}_{i'}, \text{pump}_k\right) + dp(i, k-1) \right) \tag{1}$$

2.2 Time and space complexity

- time complexity: $O(N^3)$, we use three layers of circulation
- space complexity: $O(N^2)$, we use two-dimensional arrays to record the distance sum between different villages in the case of a gas station as well as the dp array,

3 Number of Rigorous Subsequences

3.1 Ideas of solutions

The question is to find subsequences that meet the conditions. We use dynamic programming to solve the problem. During the process of traversing the sequence, use a one-dimensional array(dp[3]) to record the number of qualified subsequences ending with ”2”,”1”,”0”.

The state transition equation is:

$$\begin{cases} dp[2] = 2 \times dp[2] + 1 \\ dp[1] = 2 \times dp[1] + dp[2] \\ dp[0] = 2 \times dp[0] + dp[1] \end{cases} \tag{2}$$

The explanation of the above formula is: when traversing the sequence to element 2, the subsequence that meets the conditions has two options. One is to be connected to the previous subsequence ending with 2, and the other is to serve as a new subsequence by itself. Element 1 and element 0 can be obtained in the same way.

3.2 Time and space complexity

- time complexity: $O(N)$, we only needs to traverse the sequence once
- space complexity: $O(N)$, we only needs to traverse the sequence once

4 Save Cost for Cargo Port

4.1 Ideas of solutions

To get the minimum number of days to open the port, we need to make sure that as many ships as possible can be working simultaneously in a day. Adopting greedy algorithm , sort the ships in ascending order according to the end date. For a ship, choosing the date of work starting from the later days, which can maximize the number of ships working at the same time.

4.2 Time and space complexity

- time complexity: $O(N \lg N)$,the time complexity of sorting the vector is $O(N \lg N)$. Although in the process of arranging days, a two-level loop is used, but the time complexity is $O(N)$.
- space complexity: $O(N^2)$, we use a two-dimensional array to record the tasks of each ship