ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2022

Problem Set 2

Due: 23:59, Mar. 29, 2021

1. Submit your solutions to Gradescope (www.gradescope.com).
2. In "Account Settings" in Gradescope, set your FULL NAME to your Chinese name.
3. If you want to submit a handwritten version, scan it clearly.
4. When submitting your homework in Gradescope, match each of your solutions to the corresponding problem number.
5. Late homeworks submitted within 24 hours of the due date will be marked down 25%, submitted within 48 hours will be marked down 50%, and will not be accepted past 48 hours.
6. You are required to follow ShanghaiTech's academic honesty policies. You are allowed to discuss problems with other students, but you must write up your solutions by yourselves. You are not allowed to copy materials from other students or from online or published resources. Violating academic honesty can result in serious sanctions.

Problem 1:

Given a list of positive integers, we can choose certain sets of three values to form the sides of a triangle; in particular, the values need to satisfy the triangle inequality. Note that the same value can occur multiple times in the list, and we treat these occurrences as different instances. Design an algorithm to return the number of possible sets, and show the time complexity of your algorithm.

Example:
Input: [3, 3, 4, 5]
Output: 4 sets, $\{3,3,4\}, \{3,4,5\}, \{3,4,5\}, \{3,3,5\}$ (note that $\{3,4,5\}$ is repeated because 3 occurs twice in the input).

Problem 2:

It's a hot day and you want to buy some ice cream. The shop is offering a discount where you can buy two ice creams and get the third one for free. However, the price of the free ice cream cannot be higher than the minimum price of the two you bought. The shop has many types of ice cream. Given a list of their prices, design an algorithm to buy all the ice creams using the minimum cost. Prove your algorithm is optimal and show its time complexity.

Problem 3:

Given a list of letters, you need to merge them into a string. However, there must be at least n characters between two occurrences of the same letter. The letters can be permuted in any order, and you can add '#' characters to the string to satisfy the requirement. Design an algorithm to return the minimum length string, and prove your algorithm is optimal and show its time complexity.

Example:
Input: ['X', 'X', 'X', 'Y', 'Y', 'Y'], $n = 2$
Output: 8. One possible solution is "XY#XY#XY".

Problem 4:

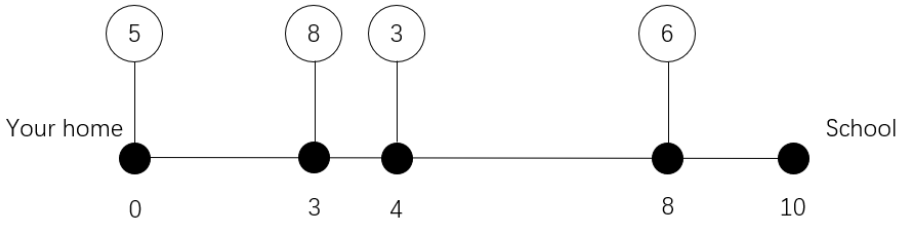
Given a list of non-negative integers, you want to merge all of them together in an arbitrary order into a single number. Design an algorithm which returns

the minimum such number. Prove your algorithm is optimal and show its time complexity.

Example:
Input: [2, 14, 134, 56]
Output: 13414256

Problem 5:

There is a road from your home to school. The length of road is L kilometers, and your home is located at coordinate 0, and the school is located at coordinate L. There are n signs along the road, where the i'th sign has a value a_i, indicating that you must travel at a speed of exactly a_i minutes per kilometer until you reach the next sign. There is also a sign at coordinate 0 which sets the initial speed. We can use the signs to calculate the time to travel from home to school. For example, in Fig. 1, the total time is $3 \cdot 5 + 1 \cdot 8 + 4 \cdot 3 + 2 \cdot 6 = 47$ minutes.



We now want to remove at most k signs, in a way which minimizes the time to travel from home to school. You cannot remove the sign at coordinate 0. Design an efficient algorithm for this problem and give its time complexity.

Problem 6:

You have two strings A and B of equal length. Both strings consist of lower case letters. You now want to change A to B. In each move you can change a segment of characters in A to any other character you want. That is, the new segment consists of only one kind of character. For example, for $A = ddddffee$ and $B = ddbbbcce$, you can first change A to $ddbbbfee$, then to $ddbbbcce$, using two

moves in total. Give an efficient algorithm to calculate the minimum number of moves you need, and show the time complexity of your algorithm.

Problem 7:

You are going to the supermarket to buy some water. There are n types of water, where the i'th type has price p_i and weighs c_i kilograms. There are an infinite number of bottles of each kind of water. You want to know the least amount of money M needed to buy at least X kilograms of water, and the actual weight Y of water you will get. If there are multiple solutions with the same minimum cost, return the one with maximum weight. Design an efficient algorithm for this problem and give the algorithm's complexity.

Problem 8:

The multiplication puzzle is played with a row of cards, each containing a single positive integer. During each move a player takes one card out of the row, and his score increases by the product of the number on the card taken and the numbers on the cards to the left and right of it. The player cannot take the first or last cards in the row. The game ends when there are only two cards left.

For example, if cards are initially [10, 1, 50, 20, 5], then player can can take cards in the order 1, 20, 50, and get a score of $10*1*50+50*20*5+10*50*5 = 500+5000+2500 = 8000$. If he took the cards in the opposite order, i.e. 50, 20, 1, his score would be $1*50*20+1*20*5+10*1*5 = 1000+100+50 = 1150$.

Design an efficient algorithm to calculate the minimum score the player can get, and show the complexity of your algorithm.