081203M04001H - Algorithm Design and Analysis

Assignment 3

October 31, 2021

Notice:

1. Please submit your answer in hard copy AND submit a digital version to UCAS website http://sep.ucas.ac.cn.
2. Hard copy should be submitted before 9.am. November 12 and digital version should be submitted before 11.30pm. November 12.
3. Please choose **three** from problems 1-6, and you should do at least the following things:
   (a) Describe the basic idea of your algorithm in natural language AND pseudo-code;
   (b) Describe the greedy-choice property and optimal substructure;
   (c) Prove the correctness of your algorithm;
   (d) Analyse the complexity of your algorithm.

1 Monkeys and Bananas

There are N Monkeys and N bananas are placed in a straight line.

Each monkey want to have a banana, if two monkeys want to own the same banana, there will be a fight! A monkey can stay at his position, move one step right from x to $x+1$, or move one step left from x to $x-1$. Any of these moves consumes 1 second. Assign monkeys to banana so that no monkey fight each other and the time when the last monkey gets a banana is minimized.

2 Job Schedule

There are n distinct jobs, labeled $J_1,\ J_2,\ \ldots, J_n$, which can be performed completely independently of one another. Each jop consists of two stages: first it needs to be preprocessed on the supercomputer, and then it needs to be finished on one of the PCs.

Let's say that job J_i needs p_i seconds of time on the supercomputer, followed by f_i seconds of time on a PC.

Since there are at least n PCs available on the premises, the finishing of the jobs can be performed on PCs at the same time. However, the supercomputer can only work on a single job a time without any interruption. For every job, as soon as the preprocessing is done on the supercomputer, it can be handed off to a PC for finishing.

Let's say that a schedule is an ordering of the jobs for the supercomputer, and the completion time of the schedule is the earlist time at which all jobs have finished processing on the PCs. Give a polynomial-time algorithm that finds a schedule with as small a completion time as possible.

3 Cross the River

Some people want to cross a river by boat. Each person has a weight, and each boat can carry an equal maximum weight limit. Each boat carries at most 2 people at the same time, provided the sum of the weight of those people is at most boat's weight limit. Return the minimum number of boats to carry every given person.

Note that it is guaranteed each person can be carried by a boat.

4 Permutation Partition

Bob is given a permutation $p_1,\ p_2, \ldots,\ p_n$ of integers from 1 to n and an integer k, such that $1 \leq k \leq n$. A permutation means that every number from 1 to n is contained exactly once.

Consider all partitions of this permutation into k disjoint segments. Formally, a partition is a set of segments $\{[s_0,\ s_1],\ [s_1+1,\ s_2], \ldots,\ [s_{k-1}+1,\ s_k]\}$, such that: $1 = s_0 < s_1 < s_2 < \ldots < s_{k-1} < s_k = n$. Two partitions are different if there exists a segment that lies in one partition but not the other.

Bob wants to calculate the partition value, defined as $\sum_{i=1}^{k} max_{s_{i-1} \leq j \leq s_i} p_j$ for all possible partitions of the permutation into k disjoint segments. Please help him find the maximum possible partition value over all such partitions, and the number of ways to make partition with this value.

5 Toy Buildings

Bob has n toy buildings in a line, the i-th from left of which has weight a_i. He wants to make these building nondecreasing in height from left to right. In one operation, he can take any contiguous subsegment of them and add 1 to each of their heights.

Help Bob find the minimum number of operations he needs to perform to make his toy buildings nondecreasing.

6 Maximum Number of Coins You Can Get

There are $3n$ piles of coins of different size, you and your friends will take piles of coins as follows: In each step, you will choose any 3 piles of coins (not necessarily consecutive). Your friend Alice will pick the pile with the maximum number of coins. You will pick the pile with submaximal number of coins. Your friend Bob will pick the last pile. Repeat until there are no more piles of coins. Given an array of integers piles where $piles[i]$ is the number of coins in the i-th pile. Return the maximum number of coins which you can have.