中国科学院大学 2016 年卜东坡算法分析与设计考试

091M4041H-2016: Final Examination

Notice:

1. Please write your name along with student ID.
2. There are 8 sections in the sheet, and for sections 1-6, you can arbitrarily choose one problem. If you answer two problems, say 1.1 and 1.2, the higher mark will be chosen.
3. When you are asked to give an algorithm, you should describe your algorithm in natural language or pseudo-codes, prove the correctness, and analyze time complexity.
4. You can write answers in either Chinese or English.

1 Divide and Conquer (12 marks)

1.1 Special Inversion Counting

Recall the problem of finding the number of inversions. Given a sequence of n distinct numbers $a_1, \cdots, a_n$. Then an inversion can be defined as a pair (a_i, a_j), where $a_i > a_j$ and $i < j$.

Inversion number can be considered as a good measure of the difference between two orderings of sequence. In certain conditions, the measure is too sensitive. Here we propose a *special inversion* if a pair satisfies $i < j$ and $a_i > 4a_j$.

Please give an $O(n \log n)$ algorithm to count the number of the special inversions between two orderings, prove the correctness and analyze the complexity.

1.2 Local Maximal in Array

Given an input array $num[0, \ldots, n-1]$ where $num[i] \neq num[i+1]$ for all $i = 0, \ldots, n-2$. A local maximal element is defined as the one which is greater than its neighbors. Suppose that $num[-1] = num[n] = -\infty$. The problem is to find one local maximal element and report its index. The array may contain several local maximals, in which case you can report any one of them.

For example, in array [1, 2, 3, 1], 3 is a local maximal element and your objective is to report the index number 2.

Please give an algorithm with $O(\log n)$ complexity, prove the correctness and analyze the complexity.

2 Dynamic Programming (12 marks)

2.1 Synonymous Subset

Now you are given a set S of distinct sentences and a judge function $J(s_1, s_2)$ judging whether exactly two sentences have exactly the same meaning. Please design an algorithm to determine the max size of S', the subset of S, in which every two sentences have exact the same meaning. Since judging every two sentences is a tough job, you should call this function as few times as possible.

Please give your algorithm with DP recurrence relations, prove the correctness and analyze the complexity of your algorithm.

2.2 Palindrome Partition

A palindrome is defined as a string that reads the same backward and forward. For example, the word "refer", is a palindrome. A palindrome partition is, given a string s, cutting s into several substrings, such that every substring of the partition is a palindrome. The Palindrome Partition problem is to report the with minimum numbers of cuts needed for a palindrome partition.

For example, given s = "*aabaacaa*", you should report 2, because the palindrome partitioning ["*aabaa*","*c*", "*aa*"] can be produced using only 2 cuts, which is the minimum.

Please give your algorithm with DP recurrence relations, prove the correctness and analyze the complexity of your algorithm.

$f(i) = f(i-1)$

3 Greedy Algorithm (12 marks)

3.1 Assignments and Deadline

Mike has just come back to school from the 30th Algorithm competition. Now he has a lot of assignments to do. Every teacher gives him a deadline of handing in the assignments. If Mike hands in the assignments after the deadline, the teacher will reduce his score of the final test. And now we assume that doing every assignment always takes one day. So how should Mike arrange the order of doing homework to minimize the reduced score.

For example, there are 4 assignments

$$\{(D_i, S_i)|i = 1, \cdots, 4\} = \{(2, 100), (3, 80), (1, 50), (2, 70)\},$$

where D_i is the deadline and S_i is the reduced score. Finally, Mike can finish the first, second, fourth homework, and the reduced score is 50.

N=7

Please give your algorithm to get the minimum reduced score (6 marks), prove the correctness (4 marks) and analyze the complexity of your algorithm (2 marks).

(3, 20) (4, 60) (2, 70) (1, 80)

3.2 Greedy Hero

DotA is a popular computer game played between two teams, each of which consists of N players. Each player independently controls a powerful character, known as a "hero". When playing DotA with ~~god-like rivals and pig-like~~ team members, you have to face an embarrassing situation: all your teammates are killed, and you have to fight with all other N rivals alone.

Let's model this problem. There are two key attributes for the heroes in the game, health point (HP) and damage per shot(DPS). Assume your hero has infinite HP and unit DPS (DPS=1), but others' heroes have limited HP ($1 \leq$ HP ≤ 1000) and fixed DPS ($1 \leq$ DPS ≤ 1000). In order to further simplify the problem, we assume the game is turn-based, but not real-time. In each round, you can attack one rival, so his HP will decrease by your unit DPS. At the same time, your HP will decrease by the summation of DPS attacked every lived rival. If one hero's HP fall equal to (or below) zero, he will die forever.

For example, there are 2 rivals. Rival A has 1 HP and 100 DPS, rival B has 100 HP and 1 DPS. You attack rival A in the first round, then you will get a 101 HP loss (attacked by both A and B). As A died in the first round, you just need to fight with B in the following rounds. After another 100 rounds, B die and you win with 100 HP loss. So the summation of HP loss is 201 under this strategy.

Although your hero is undefeated (with infinite HP), please give your algorithm to get the best strategy to kill all the enemy heroes with minimum HP loss (6 marks), prove the correctness (4 marks) and analyze the complexity of your algorithm (2 marks).

4 Linear Programming Formulation (12 marks)

Notice: Absolute value ($|\cdot|$) shall never appear in a Linear Programming (LP) formulation, since it is not a linear operation. Neither shall other non-linear operations, such as $\max\{\cdot\}$, etc.

4.1 Transportation

There are N cities, and M directed roads connecting them. Now you want to transport K units of goods from city 1 to city N. There are many robbers on the road, so you must be very careful. The more goods you carry, the more dangerous it is. To be more specific, for each road i, there is a coefficient a_i, b_j. If you want to carry x units of goods along this road, you should pay $a_i x + b_i$ dollars to hire guards to protect your goods. And what's worse, for each road i, there is an upper bound C_i, which means that you cannot transport more than C_i units of goods along this road. Please note that you can only carry integral unit of goods along each road.

You should find out the minimum cost to transport all the goods safely. Please formulate this problem as an ILP (9 marks) and explain the meaning of every constrain (3 marks).

4.2 Airplane Landing Problem and Duality

With human lives at stake, an air traffic controller has to schedule the airplanes that are landing at an airport in order to avoid airplane collision. Each airplane i has a time window $[s_i, t_i]$ during which it can safely land. You must compute the exact time of landing for each airplane that respects these time windows. Furthermore, the airplane landings should be stretched out as much as possible so that the minimum time gap between successive landings is as large as possible.

For example, if the time window of landing three airplanes are [10:00-11:00], [11:20-11:40], [12:00-12:20], and they land at 10:00, 11:20, 12:20 respectively, then the smallest gap is 60 minutes, which occurs between the last two airplanes.

Given n time windows, denoted as $[s_1, t_1], [s_2, t_2], \cdots, [s_n, t_n]$ satisfying $s_1 < t_1 < s_2 < t_2 < \cdots < s_n < t_n$, you are required to give the exact landing time of each airplane, in which the smallest gap between successive landings is maximized.

1. Formulate this problem as a LP (3 marks) and explain the meaning of every constrain (3 marks).
2. Give the dual problem of this LP. (6 marks)

5 Network Flow Formulation (12 marks)

5.1 Shooting

Given a matrix with the size of $m \times n$. For each grid, there will be an object or not. With a gun in your hand, you can shoot a row horizontally or a column vertically. After shooting, the objects on the row or the column will be destroyed at once. Your job is to use the minimum number of bullets to destory all the objects in the matrix.

1. Formulate this problem as a NF, prove the correctness, and analyze time complexity. (9 marks)
2. Explain which rows and columns you choose according to your NF formulation. (3 marks)

5.2 Football league

N football teams are playing a football league. There are only m games remained currently and you know the schedule(the two race teams for each game) of remained games and the result of each completed game. Assume each game always have one winner and one loser, no tie.

Now your team have already finished all its games and you want to know whether it is possible for your team to be the champion. The team that wins the most is the champion, if there are more than one, any of them is the champion.

Please formulate this problem as a NF, prove the correctness, and analyze time complexity. (12 marks) (Hint: This problem is modified from load balance problem in assignment.)

6 NP-completeness Reduction (12 marks)

Notice: Only the following problems which have been proved in NP-complete in slides of this course can be used as for proof in this section, in order to avoid circular reasoning:

SAT, 3SAT, CLIQUE, INDEPENDENT-SET, VERTEX-COVER, SET-COVER, SUBSET-SUM, 3-COLORING, HAMILTONIAN-CYCLE

6.1 QUARTER-3SAT

In the QUARTER-3SAT problem, given a 3SAT formula ϕ with n variables and m clauses, where m is a multiple of 4. Determine whether there exists an assignment to the variables of ϕ such that exactly a quarter of the clauses evaluate to true and three quarters of the clauses evaluate to false or not.

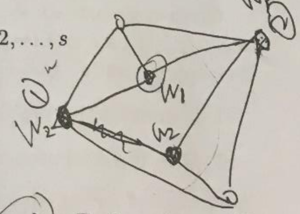
Prove that QUARTER-3SAT problem is in NP-complete.

6.2 Set Intersection Problem (SIP)

The SIP is defined as follows: Given finite sets $A_1, A_2, \ldots, A_r$ and $B_1, B_2, \ldots, B_s$, is there a set T such that

$$|T \cap A_i| \geq 1, \quad i = 1, 2, \ldots, r \quad \text{and} \quad |T \cap B_j| \leq 1, \quad j = 1, 2, \ldots, s$$

Prove that the SIP is NP-complete.



7 Independent Set on Tree (14 marks)

Given an undirected tree $T = (V, E)$, which is a connected graph without cycles. Each node $v \in V$ has positive weight w_v.

The goal is to find the MAXIMUM-WEIGHT INDEPENDENT SET, namely a set S of nodes such that:

- S is independent, which means that for any edge (u, v), at most one of u or v belongs to S.

- S has maximum total weight $\sum_{v \in S} w_v$.

A trivial greedy algorithm picks a heaviest node v^*, removes v^* and its neighbors and repeats.

1. Give an example to show that the given algorithm cannot guarantee an optimal solution. (4 marks)
2. Give a polynomial-time algorithm that finds an optimal solution to the problem. (6 marks)
3. Give a linear time algorithm for the uniform case where all nodes have unit weight. (4 marks)

Note that you should describe your algorithm in natural language or pseudo-codes, prove the correctness, and analyze time complexity.

8 Selection in Linear Time (14 marks)

We can formally specify the SELECTION problem as follows:

Input: A set A of n distinct numbers and an integer i, with $1 \leq i \leq n$.
Output: The element $x \in A$ that is larger than exactly $i - 1$ other elements of A.

We can solve the selection problem in $O(n \log n)$ time, since we can sort the numbers using *heapsort* or *mergesort*, and then simply index the ith element in the output array. But we need some faster algorithms here.

1. Give an algorithm which can find both the minimum and maximum simultaneously using at most $3\lfloor n/2 \rfloor$ comparisons. Note that you must explain why it is at most $3\lfloor n/2 \rfloor$. (4 marks)
2. Give an expected linear time algorithm that finds the ith smallest element of the array A. Note that you must analyze the expected running time. (6 marks) (Hint: The algorithm can be modeled after the *quicksort* algorithm.)
3. Give a worst-case linear time algorithm that finds the ith smallest element of the array A. Note that you must analyze the worst-case running time. (4 marks)

Note that you should give two different algorithms for problem 2 and 3.