Algorithm Design - Exam

Academic year 2018/2019

Instructor: Prof. Stefano Leonardi, Prof. Chris Schwiegelshohn

February 18, 2019

Exercise 1. We are given two strings A and B. A subsequence is of A is any string S that can be obtained by deleting characters of A. For example, the string *abaacba* is a substring of both *abcaacabba* and *aabaabcba*. The longest common subsequence problem (LCS) asks for a string S of maximum length such that S is a subsequence of both A and B

1. Give the recursive form used in the dynamic programming formulation of LCS. Specifically consider how the length of the optimal substring $OPT[i, j]$ up to position i in string A and position j in string B can be expressed as a function of $OPT[i-1, j]$, $OPT[i, j-1]$ and $OPT[i-1, j-1]$.

   Hint: The running time should be $|A| \cdot |B|$.

2. Assume that we want to find the LCS of k strings A_1, A_2, ..., A_k. Extend the dynamic program to this case.

   Hint: The running time should be $\prod_{i=1}^{k} |A_i|$.

3. If k is part of the input, the problem is NP hard. If k is constant, the problem is in polynomial time. Explain how this statement is reflected in the running time of the dynamic program.

Exercise 2. We are give n points A in Euclidean space. We would like to find a ball $B(c, r)$ centered at some point c with radius r such that all points in A are contained in r. Our objective is to minimize r. Design a 2-approximation, i.e. give an algorithm that determines a center c' such that $B(c', 2r)$ contains all points of A.

Exercise 3. Suppose we are given a set of n distinct numbers. We want to sort them with a randomized version of Quicksort. The code is given as follows:

Algorithm 1 Quicksort

Input: Array $A[i, j]$
0: If $i = j$, Return A.
1: Pick $A[k]$, $i \leq k \leq j$ uniformly at random
2: Put all elements of A that are at most $A[k]$ into an array B
3: Put all elements of A that are greater than $A[k]$ into an array C
4: Return Quicksort(B),$A[k]$,Quicksort(C)

1. What is the expected value of $A[k]$?

2. What is the expected size of B and C?

3. Suppose the randomized Quicksort algorithm has an expected running time of $10n \log n$. Give an upper bound on the probability that the algorithm requires more than $100n \log n$ units of time.

Exercise 4. You are given a graph $G(V, E)$. We would like to see if we can partition V into two sets V_1 and V_2, such that there exist a Hamiltonian cycles for V_1 and V_2. Show that this problem is NP-complete.

1. First, argue why the problem is in NP.

2. Second, give a reduction from any NP-hard problem of your choice.