PROTOTYPE MIDTERM, WINTER 2020 — COMPUTER SCIENCE 251

Examiner: Luc Devroye
Instructions: Calculators and computers are not allowed.
Weight: All questions carry equal weight.

Question 1. Write upper bound recurrences (not solutions) for T_n, the worst-case the complexity for the following problems, where n is the input size. So, $T_n \leq$ (your answer).

(i) The divide-and-conquer method for finding a majority element in a set $\{x_1, \ldots, x_n\}$ known to have a majority element. You can only use the identity oracle (i.e., the oracle that answers the question, "is $x = y$?").

(ii) The median-of-5 algorithm for finding the k-th smallest of n numbers. (Binary comparison oracle.)

(iii) Binary search in a sorted set of size n using a ternary comparison oracle.

Question 2. A polynomial of order n, $a_0 + a_1 x + \cdots + a_n x^n$, is stored as an array of coefficients, $(a_0, a_1, \ldots, a_n)$. Suppose we wish to multiply the polynomial $(a_0, a_1, \ldots, a_n)$ with the polynomial $(b_0, b_1, \ldots, b_n)$ under a model in which every multiplication and addition have a unit cost. As an example, the polynomial $(2, 5)$ multiplied with the polynomial $(1, 3)$ is $(2, 11, 15)$, because $(2 + 5x) \times (1 + 3x) = 2 + 11x + 15x^2$.

(i) How fast can you do that in $O(\cdot)$ notation?

(ii) Describe your strategy using one paragraph.

Question 3. Give explicit decision tree lower bounds in the binary comparison oracle model for these problems:

(i) Finding the second smallest of four numbers.

(ii) Sorting six numbers.

(iii) Determining if a given number is present in a sorted array of five numbers.

Question 4. We consider the longest common subsequence problem given two arrays of symbols, $x[1], \ldots, x[n]$ and $y[1], \ldots, y[m]$. The standard dynamic programming solution computes the length $L[i, j]$ of the longest common subsequence for all smaller arrays of sizes $i \leq n$ and $j \leq m$, respectively:

```
for i = 0 to n do
for j = 0 to m do
  if ij = 0 then L[i,j] ← 0
           else XXX
```

Fill in the missing part indicated by XXX. [Answer: one or two lines.]

Question 5. This question is about the Landau symbols, and the notion of worst-case time T_n taken by an algorithm on a problem of size n.

(i) True or false: If T_n is not $O(n)$, then it must be $\Omega(n)$.

(ii) True or false: If $T_n = o(n)$ then $T_n = O(n)$.

(iii) Let X_n denote an input of size n, let $\textsc{time}(X_n)$ denote the time taken by an algorithm on input X_n, and let T_n denote the worst-case time for that algorithm. List all correct conclusions if I know that $T_n = \Omega(n^2)$:

  (1) There exist constants $C > 0$ and $n_0 > 0$ such that for all $n \geq n_0$, $\textsc{time}(X_n) \geq Cn^2$ for all inputs X_n.

  (2) There exist constants $C > 0$ and $n_0 > 0$ such that for all $n \geq n_0$, $\textsc{time}(X_n) \geq Cn^2$ for some input X_n.

(iv) When we say that the worst-case time lower bound for solving a given problem of size n is at least $4n^3$, what do we mean (choose one):

  (1) Every algorithm must take at least $4n^3$ time on every input.

  (2) Every algorithm must take at least $4n^3$ time on some input.

  (3) Some algorithm must take at least $4n^3$ time on every input.

  (4) Some algorithm must take at least $4n^3$ time on some input.

Question 6.

(i) We know that Strassen's matrix multiplication takes RAM model time $\Theta\left(n^{\log_2 7}\right)$. Let us be more precise and modify the model. Assume that T_n is the number of multiplications performed if we use Strassen's method for multiplying two $n \times n$ matrices, where n is a power of two. Give (1) the exact recurrence for T_n and (2) the value of T_{64} (as an explicit formula).

(ii) Let an integer $n \geq 1$ be stored in binary. Give the precise number of bits used (as a function of n).

(iii) What is the minimum edit distance between strings `gaga` and `googol`?

Question 7. The binomial numbers $\binom{n}{k}$, $0 \leq k \leq n$, satisfy the well-known identity

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}, \binom{n}{0} = \binom{n}{n} = 1.$$

(i) In the RAM model, what is the complexity if we try to compute $\binom{n}{17}$ by a recursive program that mimics the definition? Answer in $\Theta(\cdot)$ notation.

(ii) In the RAM model, what is the complexity if we try to compute $\binom{n}{17}$ by dynamic programming? Answer in $\Theta(\cdot)$ notation.

Question 8. Let $D[i, j]$ be the distance between cities i and j, where $1 \leq i, j \leq n$. We would simply like to know—yes or no—if our traveling salesman can visit all cities in a tour given that he/she does not want to travel more than 200 kilometers on any given leg. With what you have seen thus far in the course, how would you solve this problem, and what is your complexity (in the RAM model)? [Answer: about three lines, no explicit algorithms please.]