Two hours

UNIVERSITY OF MANCHESTER
DEPARTMENT OF COMPUTER SCIENCE

Advanced Algorithms 1

Date: Thursday 23rd January 2020

Time: 14:00 - 16:00

Please answer all THREE Questions
Each question is worth 20 marks

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

1. Union find

   a) Let $f : \mathbb{N} \to \mathbb{N}$ be a function. What does it mean to say that f is *non-elementary*?
   (2 marks)

   b) Using pseudocode, write the top-level `union-find` algorithm for finding the connected components of a graph $G = (V, E)$, using the building-blocks `make-set`(v), `find`(v) and `union`(S,T), where $v \in V$, $S \subseteq V$ and $T \subseteq V$. Explain how this algorithm works.
   (4 marks)

   c) Suppose that, in the implementations of `make-set`(v), `find`(v) and `union`(S,T), sets of vertices are stored as lists, but with each vertex given a pointer to the list that contains it. Explain how, by recording the size of each cell, we can ensure that no element is moved to a different list more than $\log |V|$ times. Justify your solution.
   (6 marks)

   d) Suppose now that sets of vertices are stored as trees, rather than lists. This allows us to employ 'path compression' in `find`. Explain what this is. (You may draw a diagram.)
   (6 marks)

   e) Give an upper bound on the complexity of `union-find` employing both comparison of cell-sizes and path-compression. (You need not establish this result.) How is this connected to non-elementary functions?
   (2 marks)

2. Perfect matchings, flow networks and stable matchings.

    a) Let $G = (V, W, E)$ be a bipartite graph. What is a *matching* for G? What does it mean to say that a matching is *perfect*? (2 marks)

    b) Define the term *flow network*. In the context of flow networks, what is a *flow*? (4 marks)

    c) The problem MATCHING is defined as follows

        MATCHING
        Given: a bipartite Graph G
        Return: Yes if G has a perfect matching
        No otherwise.

        Explain how MATCHING may be solved in polynomial time by reduction to the task of flow optimization in flow networks. Show the correctness of your reduction. You may assume that integral optimal flows exist for any flow network having integral capacities, and can be computed in polynomial time. (4 marks)

    d) Suppose we are given a set of n boys and n girls. Each boy ranks all of the girls in (strict) order or preference; each girl ranks all of the boys in (strict) order or preference. In this context, what is meant by a *stable matching*? (2 marks)

    e) Explain the Gale-Shapley algorithm for computing stable matchings. You may use pseudo-code, but if so, you should explain the steps and the meanings of the variables used. (6 marks)

    f) Give an upper bound for the running time of this algorithm. Justify your answer. (2 marks)

3. Complexity classes

a) Let F be a set of functions $f : \mathbb{N} \to \mathbb{N}$. Explain the terms

i) $\textsc{Time}(F)$
ii) $\textsc{NTime}(F)$
iii) $\textsc{Space}(F)$
iv) $\textsc{NSpace}(F)$.

(4 marks)

b) Writing $\textsc{Time}(f)$ for $\textsc{Time}(\{f\})$ and so on, explain carefully why $\textsc{NTime}(f) \subseteq \textsc{Space}(f)$ for f sufficiently fast-growing. State any general assumptions about Turing machines your argument makes. (4 marks)

c) In the context of Turing machines, what is meant by the *configuration graph* for a Turing machine M with input x? How are accepting computations understood in terms of this (directed) graph? (4 marks)

d) Using the notion of configuration graphs or otherwise, explain carefully why $\textsc{NSpace}(f) \subseteq \textsc{Time}(2^{O(f)})$ for f sufficiently fast-growing. (4 marks)

e) Define the complexity classes $\textsc{PTime}$ and $\textsc{ExpTime}$. (2 marks)

f) We saw in lectures that, for $f(n)$ a non-zero polynomial, we have $\textsc{Time}(f(n)) \subsetneq \textsc{Time}(f(2n+1)^3)$. Use this fact to show that $\textsc{PTime} \subsetneq \textsc{ExpTime}$. (2 marks)

END OF EXAMINATION