UNIVERSITY OF BRISTOL

May / June 2018 Examination Period

FACULTY OF ENGINEERING

First Year Examination for the Degree of
Bachelor of Science/Master of Engineering

COMS-10007
Algorithms

TIME ALLOWED:
2 hours

Answers to COMS-10007: Algorithms

Q1. This question is about limits and alogrithmic complexity.

(a) Define what is meant by a limit in

$$\lim_{x\to\infty} f(x) = c$$

[5 marks]

(b) State L'Hôpital's rule for limits at infinity. *[5 marks]*

(c) Calculate the limit

$$\lim_{x\to\infty} \frac{\log_2 x}{x}$$

[5 marks]

(d) Order the following list of sets so each set is a subset of the one that comes after it:

$$[O(n\log n), O(n^2), O(n), O(n!), O(\log n), O(1)]$$

[5 marks]

(e) In *selection sort* the list is divided into two sublists, an initially empty sorted list and an unsorted list. In each iteration the unsorted list is searched for its smallest element and that is added to the end of the sorted list. What is big-O for this algorithm? What is big-Omega? *[5 marks]*

(f) In *bogosort* the list is permuted and then checked to see if it is in order; this is repeated until the list is sorted. What is the big-O for this algorithm. *[5 marks]*

Solution:

a) For all ϵ there exists an x_0 such that for all $x > x_0$, $|c - f(x)| < \epsilon$. [5 for exactly correct, 2 for rough attempt.]

b) If the limits of $f(x)$ and $g(x)$ are both infinite or both zero and the functions are differentiable then

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}$$

[5 marks for correct, 3 if they forget the condition on f and g].

c)

$$\lim_{x\to\infty} \frac{\log_2 x}{x} = \frac{1}{\ln 2} \lim_{x\to\infty} \frac{1}{x} = 0$$

[5 marks for correct, 2 if they recognize they need to differentiate]

d) $O(1) \subset O(\log n) \subset O(n) \subset O(n\log n) \subset O(n^2) \subset O(n!)$ [5 for all correct, 3 with one error, 1 with two].

e) Searching is always a n operation, the list to search gets smaller each time so we have the $\sum_{i=1}^{n} i$ and the algorithm is n^2 for best and worst cases. [2 for argument, 1 for big O, 2 for big Omega].

f) There are $n!$ permutations and checking for sortedness is n so $O(nn!)$. [5 for correct, 2 for O(n!)].

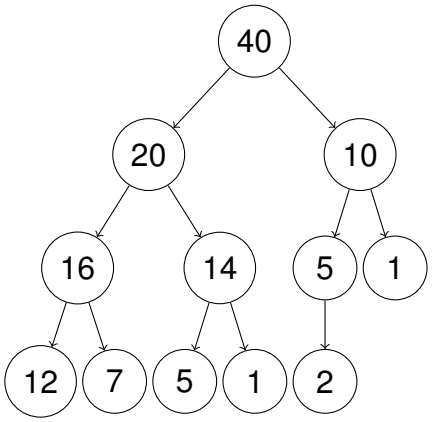
Q2. This question is about sorting and about binary heaps.

(a) Discuss the quicksort algorithm and its algorithmic complexity. *[6 marks]*

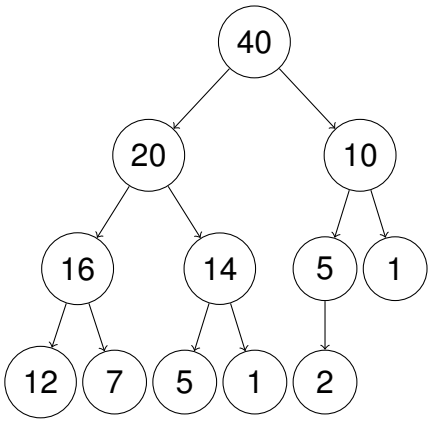
(b) Show by example, or otherwise, why it is crucial to remove the pivot at each iteration when using the quicksort algorithm. *[6 marks]*

(c) Define a binary heap. *[6 marks]*

(d) Describe how to add the element 25 to this heap. *[6 marks]*



(e) Describe how to remove the element 40 from this heap. *[6 marks]*



Solution:

a) Quicksort splits into elements less than a pivot and elements greater. In the worst case the pivot is the largest or smallest element and the only progress is removing the pivot, this is $O(n^2)$. Usually though the set is split in two, giving $T(n) = 2T(n/2) + An + B$ which is $O(n \log n)$ by the master theorem, however this relies on the splits being equal, if they aren't it is $O(n^2)$. [2 for description, 2 for algorithmic complexity, 2 for worst case]

b) Without the pivot being removed the algorithm does not always converge, for example if you are spliting as less than or equal to, are choosing the first element as the pivot and the list is already sorted. [6 for correct with example, 4 for general idea].

c) In a binary heap the only empty nodes are on the bottom row [2], each element is not smaller than its children [2] and the left child is not bigger than the right [2].

d) basically you add the new element at the first available slot in the bottom row and then bubble it up. [6 for correct, 3 for good attempt]

e) you replace the 40 by one of the elements from the bottom row and then bubble that element down, always swapping with the larger child. [6 for correct, 3 for a good attempt]

Q3. This question is about graph theory and adverserial search.

(a) Define the degree of a node for an undirected graph. *[3 marks]*

(b) Define a walk and a trail for an undirected graph. *[2 marks]*

(c) Explain why a graph with an Eulerian cycle must have only even nodes. *[5 marks]*

(d) Describe Fleury's algorithm for finding the Eulerian cycle. *[5 marks]*

(e) Recall that in the game of nim there are piles of playing pieces, in each go a player can take as many pieces as they want, provided all the pieces are from one pile, they must take at least one piece. The player who removes the last piece wins. Draw the tree for the $(2, 2, 1)$ game of nim and use minimax to decide which move the first player should play. Here $(2, 2, 1)$ means there are initially three piles, with two pieces each in two of the piles and one pile containing one piece. *[15 marks]*

Solution:

a) It is the total number of edges incident to the node. [3 for correct]

b) A walk is an ordered set of nodes and edges with each edge ending in its node, each node being followed by one of its edges. A trail is a walk that doesn't have the same edge twice. [1 mark each]

c) Usual argument due to Euler, going to a node and leaving it again uses up two nodes, so they must all be even, that is have an even degree. [5 marks, 3 marks for basic idea]

d) After you visit an edge delete it [1] After you remove the last edge from a node delete the node [1] Never cross a bridge if you have a choice [2], a bridge is an edge which, if removed, disconnects the graph [1].

e) First construct the tree, so from (2,2,1) the legitimate moves give (2,2,0), (2,1,1) and (2,0,1), leaving out possibilities that are permuations of these one, and continue on with each layer corresponding to a player [5], at the bottom points, ie only one pile left note the winner [5] and then propagate the win or loss back up to the top. [5]

Q4. This question is about minimization.

(a) What problem with gradient flow is solved by conjugate gradients and how is it solved? *[5 marks]*

(b) Explain why it might not be possible to use gradient flow? *[5 marks]*

(c) Describe the downhill simplex algorithm detailing the different moves and when they are employed? *[10 marks]*

(d) When might simulated annealling or the genetic algorithm be used instead of downhill simplex? *[10 marks]*

Solution:

a) Conjugage gradient solves the problem of zigzagging up and down the side of steep valleys by requiring the new direction is perpendicular to the previous one. [5, 3 for rough idea]

b) It is not possible to use gradient flow if the gradient is not known or is hard to calculate [5]

c) The downhill simplex uses a simplex, that is a $n+1$ linearly independent points in an n-dimensional solution space [2] it evaluates the function on each of the points in the simplex and finds the 'worst', that is the one with the highest value. It tries to move the worst node by reflecting it through all the other nodes. If this new node is better than the one it replaces but not the new best point it now repeats by finding the new worst point [2]. If the new point is now the best point it tries to extend further by replacing it with another new point the same distance again in the new direction. [2] If the new point is worse than the point it replaces, it doesn't use this point, instead it shrinks in the worst point, moving it towards the other points [2]. If that doesn't work in the sense of not improving the value, it shrinks all points towards the best point [2].

d) Downhill simplex finds local minima, the two heuristic algorithms find the global minimum. [5 for correct, 2 for some idea].

END OF PAPER