University of Limerick

Ollscoil Luimnigh

Faculty of Science and Engineering

Department of Computer Science
and Information Systems

Final Assessment Paper

Academic Year:	2015/2016	Semester:	Spring
Module Title:	Data Structures and Algorithms	Module Code:	CS4115
Duration of Exam:	2 hours	Percent of Semester Marks:	60
Lecturer:	P. Healy	Paper marked out of:	60

Instructions to Candidates:

- There are two sections to the paper: Section 1 (short questions) and Section 2 (long questions)
- The mark distribution is 15 marks for Section 1 and 45 marks for the Section 2
- Answer all questions in all sections

Section 1. Short Questions (5×3 marks).

- Please put your answers to these questions in the answer book provided to you, labelling your answers 1.1, 1.2, etc.
- Justify all answers; a justified answer is better than a guessed answer.

1. If two functions $f(n)$ and $g(n)$ are both $O(h(n))$ are the two functions asymptotically equal (differing by only a constant multiple)? That is, if $f(n) = g(n) = O(h((n))$ is it the case that $f(n) = O(g(n))$. True or false? Justify your answer.
2. How many passes of radix sort are required to sort an array of integers if we are limited to using 256 buckets on any pass? Does the size of integer being sorted play a role? Justify your answer.
3. What is the maximum number of nodes possible in a d-ary heap of height k? For node number i, what is the number of its parent?
4. In a graph the degree of a vertex, d_v, is the number of neighbours v has (or the number of edges incident upon it). Call S the sum of all of the degrees, that is, $S = \sum_{v \in V} d_v$. Give an example of a graph $G = (V, E)$ where S is odd, or explain why there can not be such a G.

5. In the strict specification of the priority queue ADT the operation `decrease_p()`, which decreases the priority of a node in the queue but leaving it remain in the queue is not mentioned. Describe a situation where this operation could be of use.

Section 2. Long Questions (45 marks).

- Please put your answers to these questions in the answer book provided to you
- Label your answers 2.1, 2.2, and 2.3 in your answer books

1. Fibonacci Numbers and Binary Trees **(15 marks.)**

   (a) Fibonacci numbers have lots of uses in computer science – heaps and trees, for example. Prove by induction that

   $$F_n = 1 + \sum_{i=0}^{n-2} F_i, \quad n \geq 2 \text{ and } \quad F_0 = 0, F_1 = 1$$

   (8 marks.)

   (b) In a binary tree every node has two pointers that are used to point to its two possible offspring. If the node does not have an offspring the unused pointer is called a *null* pointer. Prove that the number of *null* pointers in a binary tree containing n nodes is $n + 1$. (7 marks.)

2. Binary- and d-Heaps **(15 marks.)**

   (a) How many nodes are in the heap shown in Figure 2? Justify your answer. A principled answer will be considered better than a "brute-force" answer. (5 marks.)

   (b) As chief algorithmist at a big data company you are asked to develop a new Abstract Data Type called a Querity Prue (QP). The operations it should support are `insert()` and `findMin()`. What asymptotic running times can you give for this ADT? What will be your implementation? (4 marks.)

   (c) We have seen that a binary heap is a data structure that can support the `insert()` and `deleteMin()` operations required of the Priority Queue ADT in $\mathcal{O}(\log n)$-time. What are the comparable running times for a d-heap? (3 marks.)

   (d) In d-heaps, the second last child of node i is at position di. Use this fact to derive an expression for the position of the parent of node i. (3 marks.)

3. Graph Algorithms. **(15 marks.)**

   (a) Figure 1 over provides a pseudo-code implementation of Dijkstra's algorithm. Describe in detail how it works. (**Important:** I do not want a line-by-line, blow-by-blow description of the algorithm. I want evidence that you know what the code is trying to achieve.)

   For the graph shown over in Figure 3 show how the algorithm proceeds when computing the shortest path from s to e. A matrix along the lines of what we used in class will be a good way of showing your calculations / steps. (9 marks.)

(b) Given two columns of data, comprising currency names and exchange rate (expressed as, say, 1.34USD= 1€) explain how you could use a graph and an algorithm we have studied to make money fast. That is, is there a sequence of exchanges that could be found that would result in instant profit? For instance, if the currencies are x, y and z, and the exchange rate is $1x = 2y$, $1y = 2z$ and $1x = 3z$ then $300z$ will buy $100x$, which in turn will buy $200y$ and then $400z$. So $300z$ will buy $400z$ and a profit of $100z/300z = 33.3\%$ results. (6 marks.)

```
function Dijkstra(Graph, source):
    dist[source] ← 0

    create vertex set Q

    for each vertex v in Graph:
        if v ≠ source
            dist[v] ← INFINITY
            prev[v] ← UNDEFINED

        Q.add_with_priority(v, dist[v])


    while Q is not empty:
        u ← Q.extract_min()
        for each neighbor v of u:
            alt = dist[u] + length(u, v)
            if alt < dist[v]
                dist[v] ← alt
                prev[v] ← u
                Q.decrease_priority(v, alt)

    return dist[], prev[]
```

Figure 1: Pseudo-code for Dijkstra's algorithm.

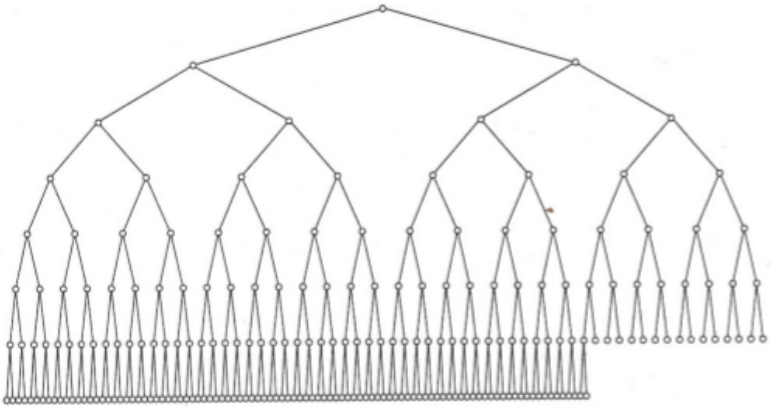

Figure 2: A binary heap.

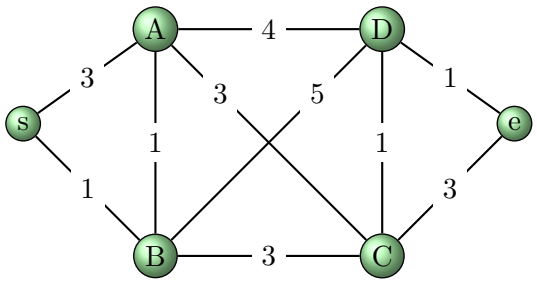

Figure 3: A six-node graph.