School of Computing and Information Systems
COMP20007 Design of Algorithms
Semester 1, 2019
Practice Mid Semester Test

Instructions to Students

- **Do not start the exam until instructed to do so.** You may read this page now.
- You may fill out your student number now in the box provided at the bottom right of this page.
- You must have your student card on display during this test.
- The total time allowed for this test is 40 minutes, including 5 minutes of reading time.
- This test contains 5 questions worth a total of 10 marks and contributes 10% of your final grade. Questions are divided into subtasks worth 0.5 or 1 mark.
- This is a closed book exam. You should **not** have any study notes of any kind, including electronic devices (no calculators, phones, etc).
- Any student seen looking at their phone (or similar) or another student's paper during the test will have their paper removed immediately, and will be referred to the School of Engineering for a breach of academic honesty.
- At the end of the mid-semester test, you **must** stop writing and turn over the test paper. Otherwise, a mark deduction applies.
- Answer all questions on the provided space. If you need more space for an answer, use the overflow section at the back of the paper. Clearly indicate near the original space that you have done so, and conversely, in the overflow section clearly state which question and subtask you are referring to.

Student Number:

Question 1 ($4 \times 0.25 + 1 \times 1 = 2$ marks)

Classify the following functions (1.a, 1.b, 1.c, 1.e) as either $O(1), O(\log n), O(n)$ or $O(2^n)$ (choosing the best bound)

1.a $2000\sqrt{n} + 7n$

1.b $n^{0.001}$

1.c $\sum_{i=0}^{n} i$

1.d $\log_{10}(n^2)$

1.e Solve the following recurrance relation for $n \geq 2$ and give your answer using Big-Theta notation. You must show all working.

$$T(n) = T(n-1) + 3n, \quad T(2) = 1$$

Question 2 ($2 \times 1 = 2$ marks)

2.a Write an *exhaustive search* algorithm which runs in $O(n^2)$ time to determine whether or not a given array `A` of `n` `int`s contains a pair of elements which sum to `t`.

Implement the C function `bool pair_sum(int A[], int n, int t)`. It should return `true` or `false`.

2.b Describe an algorithm which solves the above pair sum problem faster than $O(n^2)$. Your algorithm should have space complexity no worse than $O(n)$.

Question 3 ($2 \times 0.5 + 1 \times 1 = 2$ marks)

3.a What is the time complexity of the following algorithm in Big-Oh notation in terms of n?

```
func Algorithm(int n):
  q <- empty queue
  s <- empty stack
  for i = 0 ... n - 1: // including n - 1
    if i % 2 == 0:
      q.enqueue(i)
    if i % 3 == 0:
      s.push(i)
  while q is not empty:
    print(q.dequeue())
  while s is not empty:
    print(s.pop())
```

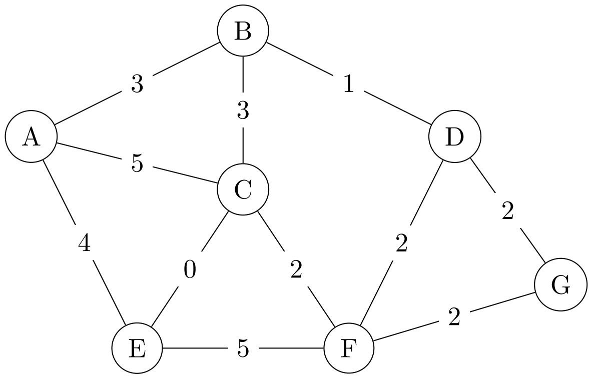
3.b What is the output of the algorithm above when run on the input $n = 10$?

3.c Order the following operations in increasing order of time complexity (*i.e.,* fastest to slowest):

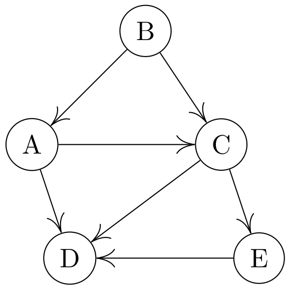
(i) Searching for a specific element in a sorted array
(ii) Sorting an array using selection sort
(iii) Deleting the last element in a singly linked list
(iv) Popping an element from a stack

Question 4 ($2 \times 1 = 2$ marks)

4.a Run Dijkstra's algorithm on the following graph, using A as the source, to compute the cost of the shortest path from A to F.



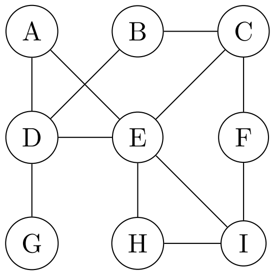
4.b Give the vertices of the following graph in topological order (*i.e.,* the return value of topological sort).



Question 5 ($1 \times 0.5 + 1 \times 1 + 1 \times 0.5 = 2$ marks)

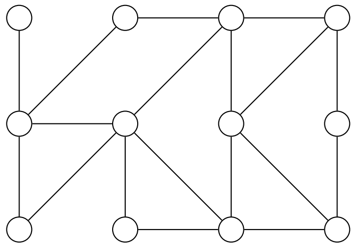
5.a How many edges are there in a forest with $\underbrace{n \text{ vertices}}$ and c connected components?

5.b Draw the DFS forest for the following graph, starting from A, break ties in alphabetic order.



What order are the vertices visited in?

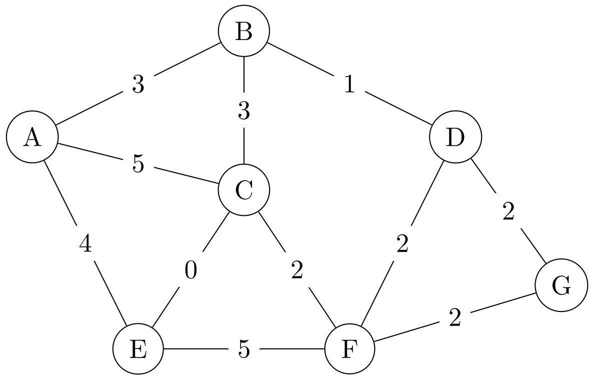
5.c Draw two different *spanning trees* for the following graph



Extra Questions

These questions aim to provide more practice problems, however are in addition to the practice mid-semester test. That is to say, the test up until this point is representative of the length of a 40 minute test.

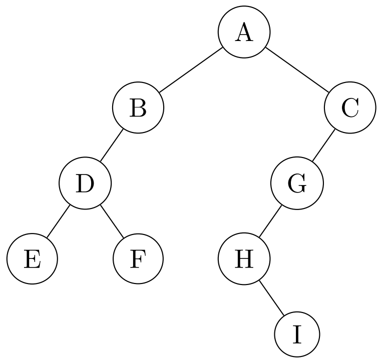
Extra.a Use Prim's algorithm to find the minimum spanning tree for the following graph.



Start with A and break ties in alphabetic order.

Draw the minimum spanning tree:

Extra.b Consider the following binary tree.



Write the order of the nodes visited in an *inorder* traversal starting at A.

Write the order of the nodes visited in a *postorder* traversal starting at A.

Extra.c Write the adjacency matrix for a directed graph given by the following adjacency list:

$$\begin{aligned} A &: \begin{bmatrix} B \end{bmatrix} \\ B &: \begin{bmatrix} C, E \end{bmatrix} \\ C &: \begin{bmatrix} D \end{bmatrix} \\ D &: \begin{bmatrix} B \end{bmatrix} \\ E &: \begin{bmatrix} A, C, D \end{bmatrix} \end{aligned}$$

Extra.d Draw the undirected graph given by the following adjacency matrix:

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 0 & 1 \\ B & 1 & 0 & 1 & 1 \\ C & 0 & 1 & 0 & 1 \\ D & 1 & 1 & 1 & 0 \end{array}$$

Extra space

You may use this page as additional space for any of your answers, as needed. If you use this page, you must clearly indicate near the original space that you have done so, and conversely, in the overflow section clearly state which question and subtask you are referring to.

END OF TEST