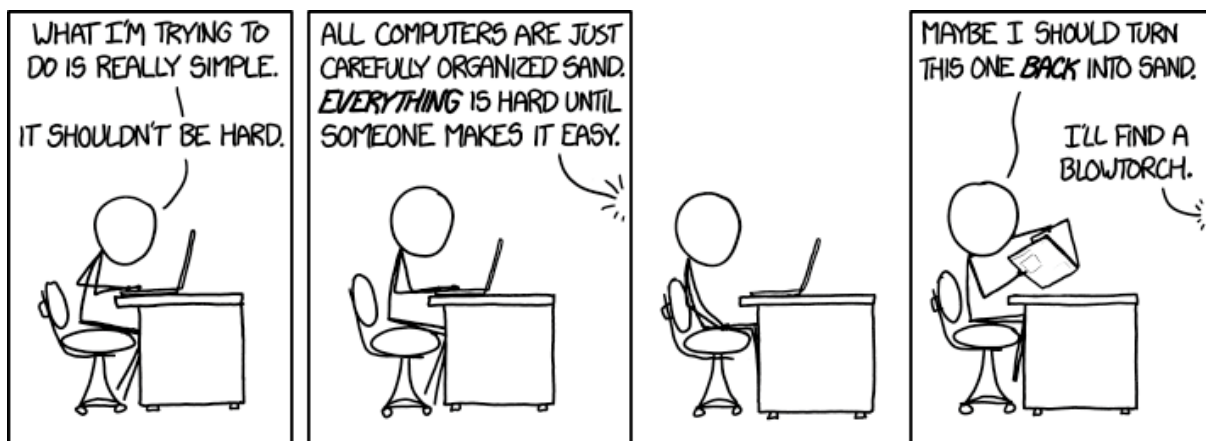


# CITS3001 Mid-semester Test 2016

**Fifty minutes**

**Answer all four questions**

**Total marks 60**



**When showing the operation of an algorithm, include enough detail to make it clear that you understand how *the algorithm* solves the problem.**

**Question 1: The Travelling Salesman Problem****(15 marks)**

*Briefly* describe the principles, operation, and performance issues of any *one* commonly-used approximation algorithm for the Travelling Salesman Problem.

Illustrate your answer using the distances in Fig. 1 for all locations a postman needs to visit in a city.

|          | <b>A</b>  | <b>B</b>  | <b>C</b>  | <b>D</b>  | <b>E</b>  |
|----------|-----------|-----------|-----------|-----------|-----------|
| <b>A</b> | –         | <b>50</b> | <b>20</b> | <b>30</b> | <b>22</b> |
| <b>B</b> | <b>50</b> | –         | <b>45</b> | <b>60</b> | <b>42</b> |
| <b>C</b> | <b>20</b> | <b>45</b> | –         | <b>40</b> | <b>26</b> |
| <b>D</b> | <b>30</b> | <b>60</b> | <b>40</b> | –         | <b>29</b> |
| <b>E</b> | <b>22</b> | <b>42</b> | <b>26</b> | <b>29</b> | –         |

**Fig. 1: Locations in a city and the distances between them (for Question 1).**

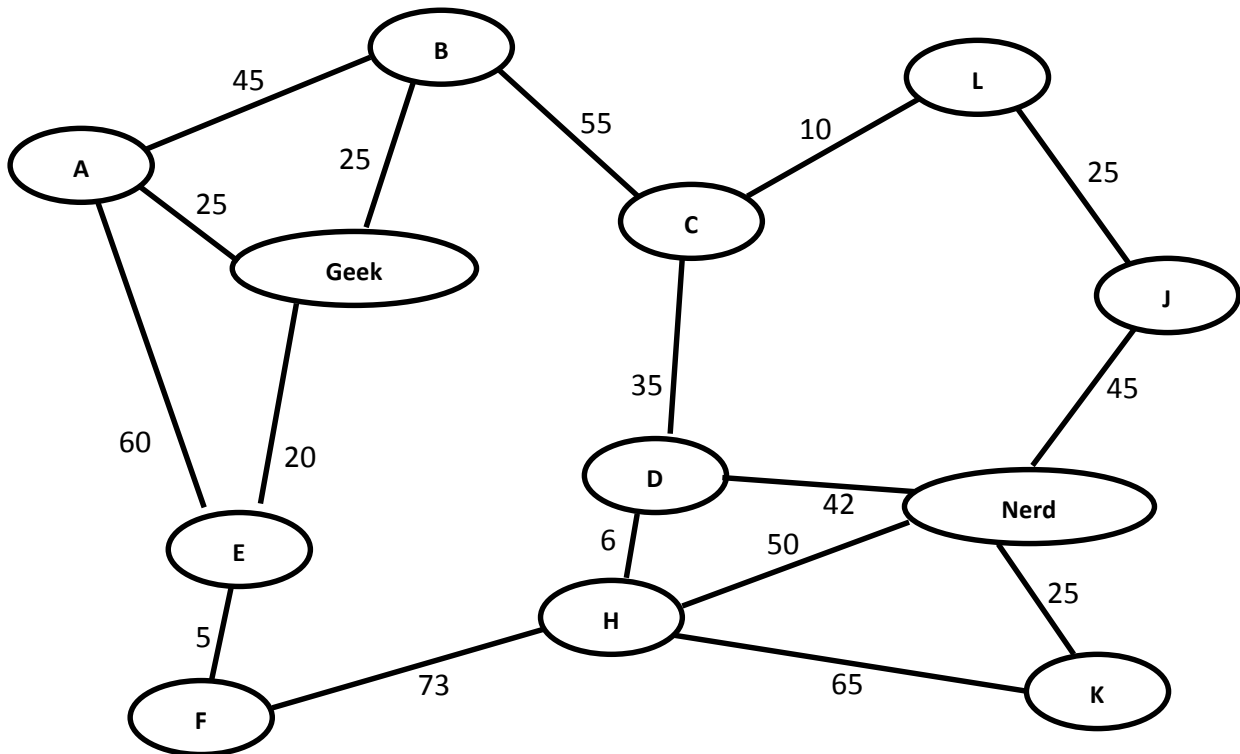
Make sure that you show all relevant operational details of your chosen algorithm.

**Question 2: Iterative-deepening depth-first search**

**(15 marks)**

*Briefly* describe the principles, operation, and performance issues of iterative deepening depth-first search.

Illustrate your answer using the road map of Au in Fig. 3. Find the route from Geek to Nerd that goes through the fewest intermediate cities.



**Fig. 2: Road map of Nerdvana (for Questions 2 and 3).**

Make sure that you show which nodes are expanded, in what order, and why.

Note that within one iteration, when you have expanded a node  $X$  at depth  $k$ , there is no need to expand  $X$  again at any depth  $j \geq k$ .

**Question 3: A\***

**(15 marks)**

*Briefly* describe the principles, operation, and performance issues of A\*.

Illustrate your answer using the road map of Nerdvana in Fig. 3 and the table of straight-line distances in Fig. 3. Find the route from Geek to Nerd that requires the least total distance.

|             |            |             |            |
|-------------|------------|-------------|------------|
| <b>Geek</b> | <b>95</b>  | <b>F</b>    | <b>120</b> |
| <b>A</b>    | <b>135</b> | <b>H</b>    | <b>48</b>  |
| <b>B</b>    | <b>100</b> | <b>J</b>    | <b>40</b>  |
| <b>C</b>    | <b>60</b>  | <b>K</b>    | <b>25</b>  |
| <b>D</b>    | <b>40</b>  | <b>L</b>    | <b>60</b>  |
| <b>E</b>    | <b>100</b> | <b>Nerd</b> | <b>0</b>   |

**Fig. 3: Straight-line distance from each city to Nerd (for Question 3).**

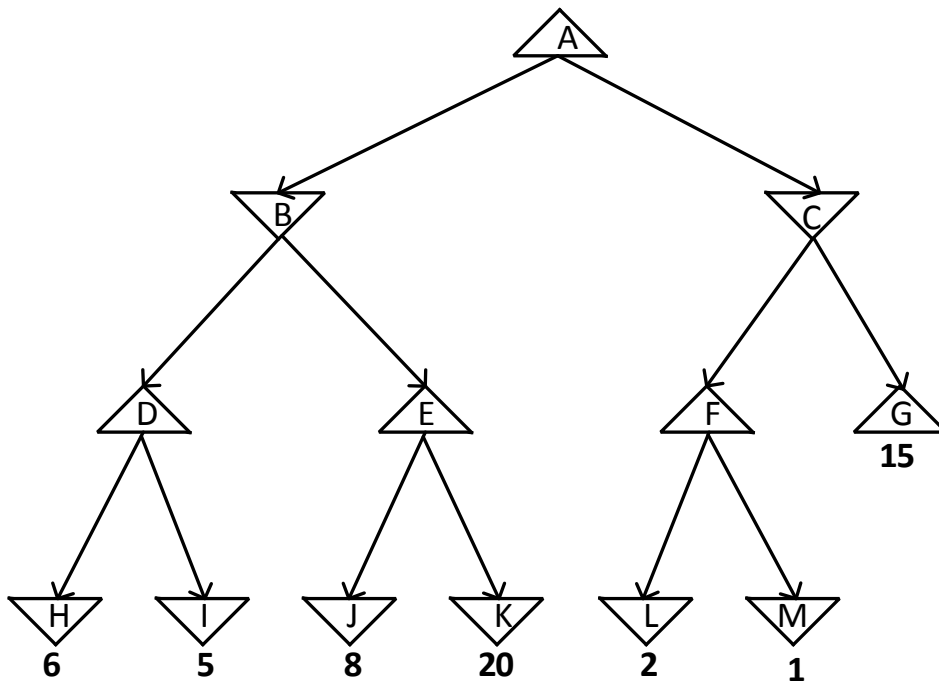
Make sure that you show which nodes are expanded, in what order, and why.

Note that when you have expanded a node  $X$  at  $g(X) = k$ , there is no need to expand  $X$  again at any  $g(X) \geq k$ .

#### Question 4: Game-Playing

(15 marks)

- (a) What is quiescence search? Explain with an example when this search is usually performed. [2+4]
- (b) What are the properties of a good evaluation function for the Minimax algorithm? [3]
- (c) How can we use  $\alpha$ - $\beta$  pruning to reduce the number of nodes examined by the Minimax algorithm? Show the steps in minimizing the number of nodes expanded for the tree in Fig. 4. [2+4]



**Fig. 4:** The game tree for Question 4. The  $\Delta$  nodes are “MAX nodes”, in which it is MAX’s turn to move, and the  $\nabla$  nodes are “MIN nodes.” The terminal nodes show the utility values for MAX.

## Solutions:

### Question 1

Various.

### Question 2

Limit = 0, Geek is not a goal

Limit = 1, G(ABE)

Limit = 2, G( A(BGE) B(AGC) E(AGF))

Limit = 3, G( A(B(AGC)G\*E(AGF)) B(A\*G\*C(BDL)) E(A\*G\*F(EH)))

Limit = 4, G( A(B(A\*G\*C(BDL))G\*E(A\*G\*F(EH))) B(A\*G\*C(B\*D(Nerd)L)) E)

So the solution with the fewest cities is GBCDN.

(GEFHN is also acceptable.)

### Question 3

| $c$  | $g(c)$ | $h(c)$ | $f(c)$ |
|------|--------|--------|--------|
| Geek | 0      | 95     | 95     |

Expand G

| $c$ | $g(c)$ | $h(c)$ | $f(c)$ |
|-----|--------|--------|--------|
| E   | 20     | 100    | 120    |
| B   | 25     | 100    | 125    |
| A   | 25     | 135    | 160    |

Expand E, ignore G and A, sort by  $f$

| $c$ | $g(c)$ | $h(c)$ | $f(c)$ |
|-----|--------|--------|--------|
| B   | 25     | 100    | 125    |
| F   | 25     | 120    | 145    |
| A   | 25     | 135    | 160    |

Expand B, ignore G and A, sort by  $f$

| $c$ | $g(c)$ | $h(c)$ | $f(c)$ |
|-----|--------|--------|--------|
| C   | 80     | 60     | 140    |
| F   | 25     | 120    | 145    |
| A   | 25     | 135    | 160    |

Expand C, ignore B, sort by  $f$

| $c$ | $g(c)$ | $h(c)$ | $f(c)$ |
|-----|--------|--------|--------|
| F   | 25     | 120    | 145    |
| L   | 90     | 60     | 150    |
| D   | 115    | 40     | 155    |
| A   | 25     | 135    | 160    |

Expand F, ignore E, sort by  $f$

| $c$ | $g(c)$ | $h(c)$ | $f(c)$ |
|-----|--------|--------|--------|
| H   | 88     | 45     | 133    |
| L   | 90     | 60     | 150    |
| D   | 115    | 40     | 155    |
| A   | 25     | 135    | 160    |

Expand H, ignore F, replace D, sort by  $f$

| $c$ | $g(c)$ | $h(c)$ | $f(c)$ |
|-----|--------|--------|--------|
| D   | 104    | 40     | 144    |
| N   | 148    | 0      | 148    |
| L   | 90     | 60     | 150    |
| A   | 25     | 135    | 160    |
| K   | 163    | 25     | 188    |

Expand D, ignore C and H, replace N, sort by  $f$

| $c$ | $g(c)$ | $h(c)$ | $f(c)$ |
|-----|--------|--------|--------|
| N   | 146    | 0      | 146    |
| L   | 90     | 60     | 150    |
| A   | 25     | 135    | 160    |
| K   | 163    | 25     | 188    |

So the route from Geek to Nerd that goes through the fewest intermediate cities is Geek, E, F, H, D, Nerd

### Question-4

(a) **Quiescence search:** It is the extended search usually performed at the end of a series of captures to reach quiescent situation where values from the evaluation function are unlikely to change much in near future.

(b) **Properties of a good evaluation function:** It should:

- o To agree with the utility function on terminal states
- o T reflect the probability of winning
- o To be time efficient, to allow maximum search depth

(c) Nodes K and G can be ignored when we start from left to right.

