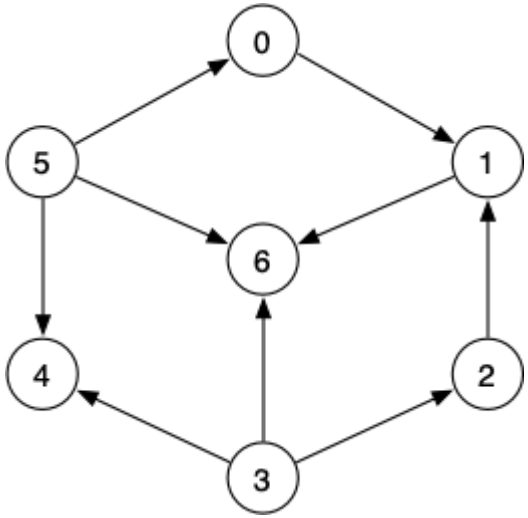
Quiz 5

Deadline	Saturday, 18 July 2020 at 11:59PM
Latest Submission	Thursday, 16 July 2020 at 9:20AM
Raw Mark	4.00/4.00 (100.00%)
Late Penalty	N/A
Final Mark	4.00/4.00 (100.00%)

Question 1 (1 mark)

Which vertices are *reachable* from vertex 3 in the following directed graph



Reminder: reachable(v,w) = there is a path from v to w

(a) ○	none
(b) ○	2, 4, 6
(c) ◉	1, 2, 4, 6
(d) ○	1, 2, 4, 5, 6
(e) ○	0, 1, 2, 4, 5, 6
(f) ○	None of the above

✔ Your response was correct.
Mark: 1.00

Question 2 (1 mark)

Which of the following would require both *weighted and directed* edges in order to be represented as a graph?

(a) ○	Facebook friends (people and their "friends")
(b) ○	The Web (pages and URLs)
(c) ◉	The road network (incl. distances and one-way streets)

(d) ○	Dependencies between files in a Makefile
(e) ○	None of the above scenarios require both weight and direction.
(f) ○	All of the above scenarios require both weight and direction.

✔ Your response was correct.
Mark: 1.00

Question 3 (1 mark)

A *transitive closure* matrix indicates *reachability* in a graph. An entry in this matrix `tc[v][w]` is set to 1 if there is a path from `v` to `w`, otherwise is set to zero.

For a graph with V vertices, this requires V^2 matrix cells. Storing each cell as an integer is wasteful. However, since each cell only has to store 1 or 0, we could store each cell as s single bit in a bit-string.

For a graph with 1000 vertices, how many bytes would be needed to store the transitive closure matrix if each row was represented by a V-bit bit-string? You can ignore the array of pointers to each matrix row; just count the bytes required for the bit-strings.

(a) ○	1000000 bytes
(b) ◉	125000 bytes
(c) ○	100000 bytes

(d) ○	10000 bytes
(e) ○	None of the above

✔ Your response was correct.
Mark: 1.00

Question 4 (1 mark)

In Dijkstra's algorithm for single-source shortest path, there are two arrays: `pred[]` and `dist[]`.

What is each of these arrays used for?

(a) ○	*`pred[v]`* indicates the predecessor of *v* (e.g. 2 is predecessor of 3) *`dist[v]`* is the length of the shortest path in which *v* is an intermediate node
(b) ○	*`pred[v]`* is an array of predicates indicating whether vertex `v` has been visited *`dist[v]`* is the length of the edge along which *v* was reached on the shortest path
(c) ○	*`pred[v]`* is the predecessor of *v* in the shortest path *`dist[v]`* is the distance between *v* and its predecessor
(d) ◉	*`pred[v]`* holds the vertex immediately before *v* along the shortest path *`dist[v]`* is the length of the shortest path from the source to vertex *v*

(e)	None of the above

✔ Your response was correct.
Mark: 1.00