

1. What is **not** a correct way to allocate memory for one stack? [Hide answers](#)

```
typedef struct StackRep *stack;
typedef struct StackRep {
    int height;
    NodeT *top;
} StackRep;
```

▲ `stack s = malloc(sizeof(StackRep))`

◆ `stack *s = malloc(sizeof(StackRep))` ✓

● `StackRep *s = malloc(sizeof(StackRep))`

■ `stack s = calloc(1, sizeof(StackRep))`

2. Consider the adjacency list representation of graphs. Which of the following is **not** true? [Hide answers](#)

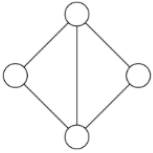


▲ depth-first search is  $O(V+E)$

◆ breadth-first search is  $O(V+E)$

● checking adjacent(v,w) is  $O(1)$  ✓

■ creating a new graph is  $O(V)$



3. This graph does **not** have [Hide answers](#)

▲ a Hamiltonian path

◆ a Hamiltonian circuit

● an Euler path

■ an Euler circuit ✓

4. What is **not** always true about the transitive closure  $tc[v][w]$  of a digraph  $G$ ? [Hide answers](#)



▲ if  $G$  has a directed path from  $v$  to  $w$ , then  $tc[v][w]=1$

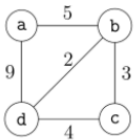
◆ if  $tc[v][w]=1$ , then  $tc[w][v]=1$  ✓

● Warshall's algorithm computes  $tc$  in  $O(V^3)$

■ if  $(v,w) \in \text{edges}(G)$ , then  $tc[v][w]=1$

5. What is **not** true about this graph  $G$  (consider neighbours in alphabetical order):

[Hide answers](#)



▲ DFS-order (start at a): a - b - c - d

◆ BFS-order (start at a): a - b - d - c

● edge(c,d)  $\in \text{MST}(G)$  ✓

■ Dijkstra order (source = a): a - b - d - c