

## CS 2501: DSA2 Quiz Booklet

This booklet contains question pools for the quizzes for this course. You should use this resource to study potential questions that will be asked on the quizzes.

DO NOT write on this booklet, as we may use it multiple times in future weeks.

There are 4 pages to this quiz booklet.

*A crash reduces  
Your expensive computer  
To a simple stone.*

## Graphs - Basic

### Short Answer Questions

1. Given an undirected graph with  $V$  nodes (no loops allowed). What is the maximum number of edges the graph can have?
2. Given a directed graph with  $V$  nodes (no loops allowed). What is the maximum number of edges the graph can have?
3. Name one advantage and one disadvantage of storing a graph as an adjacency matrix.
4. Name one advantage and one disadvantage of storing a graph as an adjacency list.
5. Describe one way we can handle storing costs of graph edges. Make sure to answer for an adjacency list AND an adjacency matrix.
6. Does breadth-first search always find the shortest path between two nodes for an undirected, unweighted graph?
7. Which has better time-complexity? BFS or DFS?
8. Which has better space-complexity? BFS or DFS?
9. What is the run-time of BFS? Explain your answer.
10. What is the run-time of DFS? Explain your answer.
11. Briefly describe how you might use DFS to count the number of disconnected components (disconnected sub-graphs) in a graph.
12. What is the run-time of topological sort?
13. What is the run-time of Dijkstra's Algorithm?
14. Dijkstra's Algorithm may not work if given negative cost edges. Provide a counter-example to illustrate this.

### Coding Questions

1. Pseudo-code a method that performs a breadth-first search on a graph. Assume each node stores a number num. Print out each num as you visit each node.
2. Pseudo-code a method that performs a depth-first search on a graph. Assume each node stores a number num. Print out each num as you visit each node.
3. Pseudo-code the topological sort algorithm.
4. Pseudo-code Dijkstra's Algorithm.

## Graphs - Advanced

### Short Answer Questions

1. Given a flow network and current flow values, produce the residual graph. A network to use will be drawn on the board.
2. When discussing flow networks, what is backflow? Explain why it is necessary.
3. Describe how the Ford-Fulkerson algorithm uses depth-first search. What purpose does it serve? Why not use breadth-first search?
4. What is the run-time of the Ford-Fulkerson Algorithm? Explain your answer.
5. Define the following terms regarding flow networks: Cut, capacity of a cut, and net-flow of a cut.
6. What does the max-flow, min-cut theorem state? What does this say about algorithms for the two problems?
7. What is a reduction? Why is it useful when comparing algorithms?
8. Describe how you would solve for the maximum-flow of a network that has multiple sources and multiple sinks.

### Coding Questions

1. Psuedo-code the Ford-Fulkerson Algorithm.
2. Prove the flow-value lemma: The the net-flow across any cut is always equal to the flow  $f$
3. Psuedo-code an algorithm that solves the Bipartite matching problem.

## Find-Union: Prims and Kruskals

### Short Answer Questions

1. Give an example of a graph (with edge costs) that has more than one minimum spanning tree.
2. How many edges does a minimum spanning tree have as a function of the original connected graph  $G = (V, E)$ ? Explain why.
3. True or False: If the edge weights of a graph are all unique, then the minimum spanning tree is unique as well. Explain your answer.
4. Given a graph (drawn on the board) and a starting node, step through Prim's algorithm. Draw the MST and list the order in which the nodes are added to the tree.
5. What do we mean when we say that the Minimum Spanning Tree problem has optimal substructure?
6. What is the runtime of Prim's algorithm?
7. What is the runtime of Kruskal's algorithm?
8. Describe how the `findSet(int i)` method works for a find-union data structure? What is the runtime?
9. Describe how the `union(int i, int j)` method works for a find-union data structure. What is the runtime?
10. When implementing a find-union structure, what is union by rank? Why is it useful?
11. When implementing a find-union structure, what is path compression? Why is it useful?

### Coding Questions

1. Write pseudo-code for Prim's algorithm
2. Write pseudo-code for Kruskal's algorithm
3. Pseudo-code the `findSet(int i)` method of a find-union structure.
4. Pseudo-code the `union(int i, int j)` method of a find-union structure.