

# Problem Set 8

Ekkapot Charoenwanit  
Efficient Algorithms

## Problem 1. BFS and DFS

Consider the following undirected graph  $G$ .

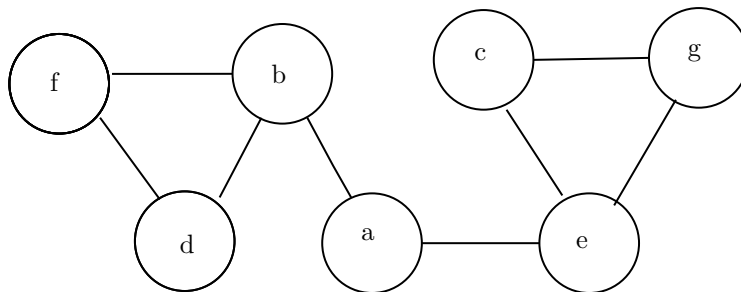


Figure 1: Undirected Graph  $G = (V, E)$

- (1) Use BFS to traverse the graph  $G$  in Figure 1 in the alphabetical order starting from vertex  $a$ . Construct a BFS search tree and identify the frontier set at each level.
- (2) Use DFS to traverse the graph  $G$  in Figure 1 in the alphabetical order starting from vertex  $a$ . Identify the type of each edge of  $G$ .

## Problem 2. Applications of BFS and DFS

- (1) How can you detect a cycle in an **undirected** graph with BFS?
- (2) Given an **unweighted** graph  $G = (V, E)$  and a designated root vertex  $s$ , explain how to compute a shortest path from  $r$  to every other nodes in  $V$  in linear time in the number of vertices  $|V|$  and edges  $|E|$ . Assume that a shortest path from a vertex  $i$  to a vertex  $j$  in an **unweighted** graph is defined as a path with the minimum number of edges.
- (3) Based on your idea in (2), compute a shortest path from  $a$  to every other vertex  $v \in V$  of the graph in Figure 1.

## Problem 3. Dijkstra's Algorithm

- (1) Solve the following shortest path problem, with vertex  $a$  as the source.

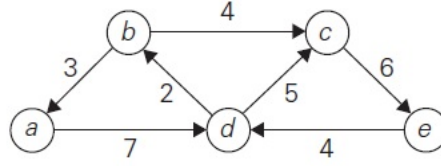


Figure 2: Solve the single-source shortest path with  $a$  as the source.

- (2) How do you apply Dijkstra's algorithm on a **directed** graph  $G = (V, E, w)$  to find a shortest path from every vertex  $v \in V$  to a given destination vertex  $t \in V$  in the  $G$ ?
- (3) Apply the algorithm you proposed in (2) to the graph in Figure 2.
- (4) Analyze the running time of your algorithm in (2) in terms of  $|V|$  and  $|E|$ .
- (5) How do you apply Dijkstra's algorithm on a **undirected** graph  $G = (V, E, w)$  to find a shortest path from every vertex  $v \in V$  to a given destination vertex  $t \in V$  in the  $G$ ?
- (6) Analyze the running time of your algorithm in (5) in terms of  $|V|$  and  $|E|$ .

#### Problem 4. Bellman-Ford

- (1) Apply the Bellman-Ford algorithm to the graph in Figure 2 with vertex  $a$  as the source vertex. Show your work in each pass. Determine whether  $d[v] = \delta(s, v)$  before  $|V| - 1$  passes for all  $v \in V$ .
- (2) Given a directed graph  $G = (V, E)$ , suppose that  $G$  contains negative-weight cycles. Modify the Bellman-Ford algorithm so that it sets  $v.d = -\infty$  for all vertices  $v$  for which there is a **negative-weight cycle** on some path from the source  $s$  to  $v$ .
- (3) How do you detect if a directed graph  $G = (V, E)$  has a **negative-weight cycle** using the Bellman-Ford algorithm? Analyze the running time of your solution in terms of  $|V|$  and  $E$ .

#### Problem 5. Floyd-Warshall and Johnson's algorithm

$$\begin{bmatrix} 0 & 2 & \infty & 1 & 8 \\ 6 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix}$$

Figure 3: Apply the Floyd-Warshall algorithm to the following weight matrix.

- (1) Apply the Floyd-Warshall algorithm to the graph represented by the weight matrix in Figure 3 to find shortest paths among all pairs of vertices  $u, v \in V$ . Give the matrix  $D^{(k)}$  in each step  $k$ .
- (2) Apply the Floyd-Warshall algorithm to the graph represented by the weight matrix in Figure 3 to find the transitive closure of  $G$ . Give the matrix  $T^{(k)}$  in each step  $k$ .

- (3) How can you detect the presence of negative-weight cycles in a directed graph using the output matrix of the Floyd-Warshall algorithm?
- (4) Apply Johnson's algorithm to the graph in Figure 4. Show the values of  $h$  and  $\hat{w}$ .

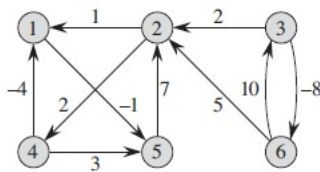


Figure 4: Apply Johnson's algorithm to the graph.