CS300 Homework #4

Hyeonguk Ryu

hyeonguk@kaist.ac.kr

Total 100 points Due: 2020–05–14 23:59:00 KST

- No partial credit will be given.
- Write in English or Korean.

1. (20 points)

The algorithm below prints vertices in order of increasing distance from vertex 1, based on BFS. For example, the algorithm prints "1, 2, 3, 4" or "1, 3, 2, 4", on the graph shown in Figure 1. Note that (distance to 2) = (distance to 3) = 1.

> *Input*: Graph $G = (V, E)$, directed or undirected.
>
> *Output*: All the vertices reachable from vertex 1, in order of increasing distance from vertex 1.

```
for all $u \in V$ do
    visited($u$) = false
$Q$ = [1] (queue containing just 1)
while $Q$ is not empty do
    $u$ = eject($Q$)
    print $u$
    visited($u$) = true
    for all edges $(u, v) \in E$ do
        if not visited($v$) then
            inject($Q$, $v$)
```

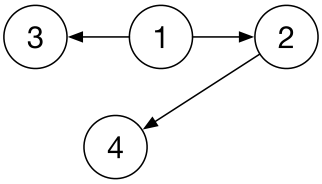


Figure 1

(a) Give a counterexample to the algorithm. (Draw your graph.) (10 points)

(b) Fix the algorithm by adding up to two lines of pseudocode, without deleting or modifying the existing code. (10 points)

2. (20 points)

Suppose Dijkstra's algorithm is run on the graph shown in Figure 2, starting at node 1. Replace *abcdefgh* with your 8-digit student ID number, and draw a table showing the intermediate distance values of all the nodes at each iteration of the algorithm. For example, if your student ID number is 23502114, your graph should look like the graph shown in Figure 3.

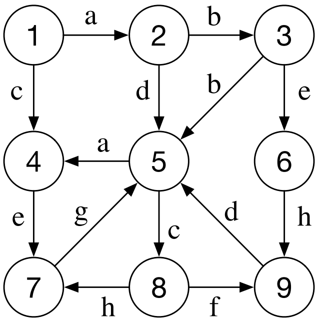
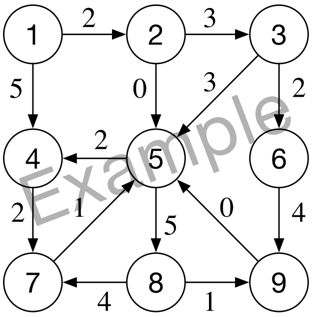


Figure 2: Input graph. Replace *abcdefgh* with your student ID number.



	1	2	3	4	5	6	7	8	9
dist	0	∞	∞	∞	∞	∞	∞	∞	∞
dist	0	2	∞	5	∞	∞	∞	∞	∞
dist	0	2	5	5	2	∞	∞	∞	∞

…

Figure 3: Example with $abcdefgh = 23502114$.

3. Dijkstra's algorithm with negative weights (20 points)

(a) Draw a directed graph with at least one negative edge on which Dijkstra's algorithm fails. Your graph must have at least five vertices, and each vertex must be connected to at least one edge. (10 points)

(b) Draw a directed graph with at least one negative edge on which Dijkstra's algorithm works correctly. Your graph must have at least five vertices, and each vertex must be connected to at least one edge. (10 points)

4. (40 points)

You are given a directed graph $G = (V, E)$ with (possibly negative) weighted edges, in which the shortest path between any two vertices is guaranteed to have at most k edges. Give an algorithm that finds the shortest distance between two vertices u and v in $O(k|E|)$ time. Show *briefly* that your algorithm is correct and runs in $O(k|E|)$ time. Suppose that there is no negative cycle in G, and $|V| - 1 \leq |E|$.