Homework #87

For CS690, Prof. Carl Gauss

by Theo Park

Due: 31 Feb 1984

Contents

Problem 1: Algorithms **3**

Pseudocode 3

Code Implementation 3

Problem 1: Algorithms

Pseudocode

Problem:
(10 points) Design pseudocode for graph traversal algorithms.

Solution

```
1:  function BFS(G = (V, E), s ∈ V)                    // s is the starting vertex
2:      /* Queue-based implementation                                          */
3:      Q ← empty queue
4:      d ← array initialized to nil

5:      /* Visit the starting vertex                                           */
6:      d[s] ← 0
7:      Enqueue s to Q

8:      while Q is not empty do
9:          u ← vertex dequeued from Q
10:         for each edge v adjacent to u for which d[v] is nil do
11:             d[v] ← d[u] + 1
12:             Enqueue v to Q
13:     return d
```

Or list style

dfs(G)

// Depth first search

1. If v is unmarked
    A. mark v
    B. For each edge (v, w) from v
        (i) DFS(w)

Code Implementation

Problem:
(10 points) Implement the pseudocode in Python.

Solution

```python
def bfs(self, s: int):  # s is the starting node
    q = deque()
```

```
    dist = [-1] * self.n
    path = [[] for _ in range(self.n)]
    visited = [False] * self.n

    q.append(s)
    dist[s] = 0
    path[s] = [0]
    visited[s] = True

    while q:
        u = q.pop()

        for v in self.adjList[u]:
            if not visited[v]:
                q.append(v)
                dist[v] = dist[u] + 1
                path[v] = path[u] + [v]
                visited[v] = True

    return dist, path, visited
```