

Solutions to Section Handout #7

1. Coding depth-first search

```
/*
 * Implementation notes: pathExists, pathExistsVisited
 * -----
 * The pathExists function is a wrapper for pathExistsVisited, which takes
 * an extra parameters to record the set of visited nodes.
 */

bool pathExists(Node *n1, Node *n2) {
    Set<Node *> visited;
    return pathExistsVisited(n1, n2, visited);
}

bool pathExistsVisited(Node *n1, Node *n2, Set<Node *> & visited) {
    if (n1 == n2) return true;
    if (visited.contains(n1)) return false;
    visited.add(n1);
    for (Arc *arc : n1->arcs) {
        if (pathExistsVisited(arc->finish, n2, visited)) return true;
    }
    return false;
}
```

2. Coding breadth-first search

```
/*
 * Implementation notes: pathExists
 * -----
 * The queue keeps track of unexplored nodes in order by hop count from n1.
 */

bool pathExists(Node *n1, Node *n2) {
    Set<Node *> visited;
    Queue<Node *> queue;
    queue.enqueue(n1);
    while (!queue.isEmpty()) {
        Node *node = queue.dequeue();
        if (node == n2) return true;
        if (!visited.contains(node)) {
            visited.add(node);
            for (Arc *arc : node->arcs) {
                queue.enqueue(arc->finish);
            }
        }
    }
    return false;
}
```

3a) Lounge, Conservatory, BallRoom, BilliardRoom, Library, Hall, DiningRoom, Kitchen, Study

3b) Kitchen, BallRoom, DiningRoom, Study, BilliardRoom, Conservatory, Hall, Lounge, Library

3c)

```

Fix distance to Lounge at 0
Process the arcs out of Lounge (Conservatory, DiningRoom, Hall)
  Enqueue the path: Lounge -> Conservatory (3)
  Enqueue the path: Lounge -> DiningRoom (4)
  Enqueue the path: Lounge -> Hall (8)
Dequeue the shortest path: Lounge -> Conservatory (3)
Fix distance to Conservatory at 3
Process the arcs out of Conservatory (BallRoom, BilliardRoom, Lounge)
  Enqueue the path: Lounge -> Conservatory -> BallRoom (7)
  Enqueue the path: Lounge -> Conservatory -> BilliardRoom (10)
  Ignore Lounge because its distance is fixed
Dequeue the shortest path: Lounge -> DiningRoom (4)
Fix distance to DiningRoom at 4
Process the arcs out of DiningRoom (BallRoom, Hall, Kitchen, Lounge)
  Enqueue the path: Lounge -> DiningRoom -> BallRoom (11)
  Enqueue the path: Lounge -> DiningRoom -> Hall (12)
  Enqueue the path: Lounge -> DiningRoom -> Kitchen (15)
  Ignore Lounge because its distance is fixed
Dequeue the shortest path: Lounge -> Conservatory -> BallRoom (7)
Fix distance to BallRoom at 7
Process the arcs out of BallRoom (BilliardRoom, Conservatory, DiningRoom, Kitchen)
  Enqueue the path: Lounge -> Conservatory -> BallRoom -> BilliardRoom (14)
  Ignore Conservatory because its distance is fixed
  Ignore DiningRoom because its distance is fixed
  Enqueue the path: Lounge -> Conservatory -> BallRoom -> Kitchen (14)
Dequeue the shortest path: Lounge -> Hall (8)
Fix distance to Hall at 8
Process the arcs out of Hall (DiningRoom, Library, Lounge, Study)
  Ignore DiningRoom because its distance is fixed
  Enqueue the path: Lounge -> Hall -> Library (15)
  Ignore Lounge because its distance is fixed
  Enqueue the path: Lounge -> Hall -> Study (12)
Dequeue the shortest path: Lounge -> Conservatory -> BilliardRoom (10)
Fix distance to BilliardRoom at 10
Process the arcs out of BilliardRoom (BallRoom, Conservatory, Library)
  Ignore BallRoom because its distance is fixed
  Ignore Conservatory because its distance is fixed
  Enqueue the path: Lounge -> Conservatory -> BilliardRoom -> Library (14)
Dequeue the shortest path: Lounge -> DiningRoom -> BallRoom (11)
Ignore this path because the distance to BallRoom is fixed
Dequeue the shortest path: Lounge -> DiningRoom -> Hall (12)
Ignore this path because the distance to Hall is fixed
Dequeue the shortest path: Lounge -> Hall -> Study (12)
Fix distance to Study at 12
Process the arcs out of Study (Hall, Kitchen, Library)
  Ignore Hall because its distance is fixed
  Enqueue the path: Lounge -> Hall -> Study -> Kitchen (15)
  Enqueue the path: Lounge -> Hall -> Study -> Library (19)
Dequeue the shortest path: Lounge -> Conservatory -> BallRoom -> BilliardRoom (14)
Ignore this path because the distance to BilliardRoom is fixed
Dequeue the shortest path: Lounge -> Conservatory -> BallRoom -> Kitchen (14)
Fix distance to Kitchen at 14
Process the arcs out of Kitchen (BallRoom, DiningRoom, Study)
  Ignore BallRoom because its distance is fixed
  Ignore DiningRoom because its distance is fixed
  Ignore Study because its distance is fixed
Dequeue the shortest path: Lounge -> Conservatory -> BilliardRoom -> Library (14)

Shortest path: Lounge -> Conservatory -> BilliardRoom -> Library (14)

```

3d)

3: Conservatory - Lounge
3: Kitchen - Study
4: BallRoom - Conservatory
4: BilliardRoom - Library
4: DiningRoom - Lounge
4: Hall - Study
7: BallRoom - BilliardRoom
7: BallRoom - DiningRoom (not needed)
7: BallRoom - Kitchen
7: BilliardRoom - Conservatory (not needed)
7: Hall - Library (not needed)
7: Library - Study (not needed)
8: DiningRoom - Hall (not needed)
8: Hall - Lounge (not needed)
11: DiningRoom - Kitchen (not needed)

The minimum spanning tree therefore looks like this:

