Objectives:

1. Introduce Shortest Path Algorithms (Floyd and Dijkstra).

Reading Assignment:

Neapolitan and Naimipour: Chapters 3.2, 4.2.

Contents:

1. Why Shortest Path Algorithms are necessary. (\(\frac{1}{2}\) hour)
2. Definitions (\(\frac{1}{2}\) hour)
3. Adjacency Matrix (\(\frac{1}{2}\) hour)
4. Floyd’s Algorithm for Shortest Paths (\(\frac{3}{4}\) hour)
5. Dijkstra’s Algorithm for Single-Source Shortest Paths (\(\frac{3}{4}\) hour)
1. Why Shortest Path Algorithms are necessary. (1/2 hour)
   - Illustrate using an example of air-travelers trying to determine the shortest way to fly from one city to another when a direct flight does not exist.

2. Definitions (1/2 hour)
   - vertex – a circle or point on the graph.
   - edge – a line from one vertex to another.
   - digraph – directed graph.
   - weighted graph – graph where edges have values associated with them.
   - path – a sequence of vertices such that there is an edge from each vertex to its successor.

3. Adjacency Matrix (1/2 hour)
   - A way to solve a shortest path optimization problem.
   - Example:
     
     |   | 1 | 2 | 3 | 4 | 5 |
     |---|---|---|---|---|---|
     | 1 | 0 | 1 | 3 | 1 | 4 |
     | 2 | 8 | 0 | 3 | 2 | 5 |
     | 3 | 10| 11| 0 | 4 | 7 |
     | 4 | 6 | 7 | 2 | 0 | 3 |
     | 5 | 3 | 4 | 6 | 4 | 0 |

4. Floyd’s Algorithm for Shortest Paths (3/4 hour)
   ```c
   void floyd (int n, const number W[ ][ ], number D[ ][ ] )
   {
       index i, j, k;

       D=W;
       for (k=1, k<=n, k++)
           for (i=1, i<=n, i++)
               for (j=1, j<=n, j++)
                   D[i][j] = minimum(D[i][j], D[i][k]+D[k][j]);
   }
   ```

5. Dijkstra’s Algorithm for Single-Source Shortest Paths (3/4 hour)
   ```c
   Y = {v1};
   F = NULL;
   While (the instance is not solved)
   {
       select a vertex v from V – Y, that has a shortest path from v1, using only vertices in Y as intermediates
       add the new vertex v to Y;
   }
   ```
add the edge (on the shortest path) that touches v to F;

if (Y==V)
    the instance is solved; // solution check
}
Handouts, etc. for Lecture: None.