Programming Design, Spring 2016
Suggested Solution for Homework 7

Solution provider: Parker Chiang
Department of Information Management
National Taiwan University

Problem 1

(a) The adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

(b) By definition, we have

$$A^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

(c) $A^{(2)}_{ij} = 1$ means i can get to j in exactly two steps. The graph is in Figure 1.

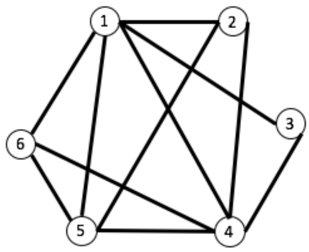

Figure 1: $A^{(2)}$ graph

(d) Find $B^{(2)} = A^{(2)} + A^{(1)} + I$. Count the number of js that makes $B^{(2)}_{ij} > 0$.

(e) The following pseudocode works:

```
void addMatrix(int [][] map, int k){
  int [][] ori_map = map
  for (i from 1 to k){
    matrix_product(map,ori_matrix)
  }
}

int main{
  cin i , k
  addMatrix(map,k)
    for (j){
      if (A[i][j]>=1){
        count++
      }
    }
  cout count
}
```

Problem 2

Please see the attached CPP file.

Problem 3

Please see the attached CPP file.