**COMP9101 Ass03**

**Name: Zanning Wang**

**zID: z5224151**

文本

描述已自动生成

**Q5.**

This problem similar to the Q1, we can also use dynamic programming to solve the question.

We use dp[X][Y][N] to express the maximum total weight from node X to node Y with passing the number of K edges.

For dp[X][Y][N], the X, Y ∈ vertices V and j∈[0, k]

The base case should be for each X, Y passing the number of 1 edge and 0 edge.

When N = 0, dp[X][Y][0] = 0;

When N = 1, dp[X][Y][1] shows the weight of the adjacent vertices.

Use the w(X, Y) to show that weight from node X to node Y.

The recursion shows below:

dp[X][Y][N+1] = max(dp[X][Y][N] + w(X, Y)); which X, Y should be the adjacent vertices.

Record each node X and Y have been through into a list should be the result path.

The time complexity for this problem should be O(E \* K)