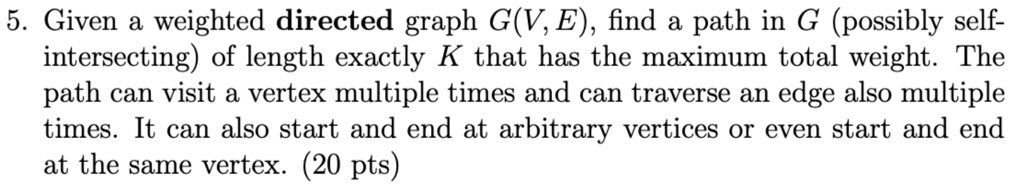
COMP9101 Ass03
Name: Zanning Wang
zID: z5224151



5. Given a weighted **directed** graph $G(V, E)$, find a path in G (possibly self-intersecting) of length exactly K that has the maximum total weight. The path can visit a vertex multiple times and can traverse an edge also multiple times. It can also start and end at arbitrary vertices or even start and end at the same vertex. (20 pts)

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)