JIE MEI z5173405

Question 4

According to the question, we can apply dynamic programming to divide this problem into n sub-problems and find the optimal solution for all sub-problems. Recursively solve each sub-problem. Assume $opt(i, K)$ that meaning when the fixed vertex is i, the length happens to be the maximum weight of k. We need to traverse every vertex in the graph G(V,E). the vertex $i_1, i_2, \ldots, i_n \in V$ represent that the path end at i. Find all vertices that can go to i and record as $v_1, v_2, \ldots, v_n \in V$. $W(v_1, i_1)$ represent that the weight value of vertex v_1 to i_1.

Then perform recursive calculations according to the following formula:

It is obvious that base case is $\boldsymbol{opt(i, 0) = 0}$

When the length = 1: $\boldsymbol{opt(i, 1) = \mathbf{Max}\,(W(v_1, i), W(v_2, i), \ldots, W(v_n, i))}$

When the length = K: $\boldsymbol{opt(i, K) = \mathbf{Max}\,(W(v_1, i) + opt(v_1, K-1), W(v_2, i) + opt(v_2, K-1), \ldots, W(v_n, i) + opt(v_n, K-1))}$

We need to record the maximum weight information obtained by each vertex in an array.

Max weight = $\boldsymbol{Max(opt(i_1, K), opt(i_2, K), \ldots, opt(i_n, K))}$

Finally, we can get the path in G of length exactly K that has maximum total weight.