Given: A directed weighted graph with V vertexes and E edges, G(V,E). And integer stand for length K
Aim: Find a path that has maximum weight.

Solution:

In order to find the maximum weighted graph, for each vertexes on the graph we try to calculate all possible weights produced by combining edges that totally have length K. The solution is trying to find the maximum weight on the adjacent edges of each node on G. Select maximum weight on each node and add together until we reach K length, then this is the answer.

1. Set up: There are two array which are Vertex-List = []. Weights that adjacent to the vertex will represented by an array list W = [] at the beginning. Total Weight will be represented as TW = 0 at the beginning; Each vertex in Vertex-List will be represented as i.
2. Sub-problem: For every node in i, we will try all the possible length case, for a value k that in range 0 < k <= K. In each case append the value of Weight on the edge(i, Vertex-List[one of vertex in adjacent of i]) to list W. The value TW in each sub-problem is

$$TW[k] = TW[i-1] + \max(W)$$

After each iterate the value of W will reset to [].

3. In the end the total value of maximum value will be OPT(problem) = $\sum_{j=0}^{K} TW[j]$