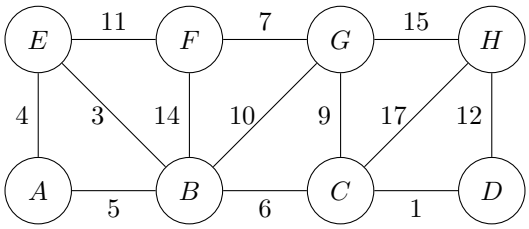
Weekly Assignment 4 :
Dijkstra

September 2023

1. Run Dijkstra's algorithm on the weighted graph below, using vertex A as the source. Write the vertices in the order which they are marked and compute all distances at each step.



2. Consider a directed graph $G = (V, E)$ that models a communication network. Each edge $(u, v) \in E$ has an associated value $r(u, v)$, which is a real number in the interval $(0, 1)$ that represents the reliability of a communication channel from u to v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail. If we assume that these probabilities are independent, then the reliability of a path $p = v_1 \to v_2 \to \cdots \to v_k$ can be computed by taking the product of the probabilities of all the channels contained in it:

$$r(p) \quad = \quad \prod_{i=1}^{k-1} r(v_i, v_{i+1}).$$

Give an efficient algorithm to find the most reliable path — the path p for which $r(p)$ is maximal — from a given source node s to any node in the network. Discuss the time complexity of your algorithm and argue why it is correct.

3. A *d-ary heap* is like a binary heap, but (with one possible exception) non-leaf nodes have d children instead of 2 children.
   (a) How would you represent a d-ary heap in an array?
   (b) What is the height of a d-ary heap of n elements in terms of n and d?
   (c) Give an efficient implementation of EXTRACT-MAX in a d-ary max-heap. Analyze its running time in terms of d and n.
   (d) Give an efficient algorithm of INCREASE-KEY(A, i, k), which flags an error if $k < A[i]$, but otherwise sets $A[i] = k$ and then updates the d-ary max-heap structure appropriately. Analyze its running time in terms of d and n.
   (e) Give an efficient implementation of INSERT in a d-ary max-heap. Analyze its running time in terms of d and n.

Figure 1: Airlines network.

4. Airlines companies sell to travelers flights with or without connections, mostly depending on their airlines network. It can happen that secondary airlines are covered with small and slow aircrafts, hence a longer transit time. It can then be faster to fly via another city (connecting flight) rather than taking a direct flight, but that's not always the case. As connections are unpleasant, direct flights are preferred if the total flight time is equal. Otherwise, if the direct flight takes longer, then the connecting flight is preferred.

   Let us define $easiest[destination]$ = minimum number of successive flights in a shortest path from *origin* to destination.

   In our airlines example, the *easiest* values are (for origin Madrid):

   - 0 for Madrid.
   - 1 for Amsterdam, Berlin, Rome.
   - 2 for Stockholm, Kyiv.
   - 3 for Moscow.

   We can transform the problem of the airline company into a graph problem. Give an efficient algorithm taking as input a weighted graph $G = (V, E)$ and a source vertex s, and as output the values of *easiest* for all vertices. Discuss correctness and efficiency of your algorithm.

5. Given the heap below:

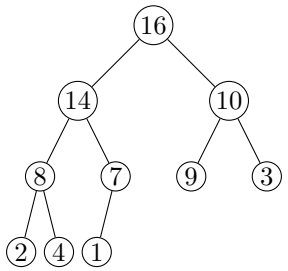


Figure 2: A Max-Heap with 10 elements

(a) insert 11

(b) remove 14