Algorithmics and complexity

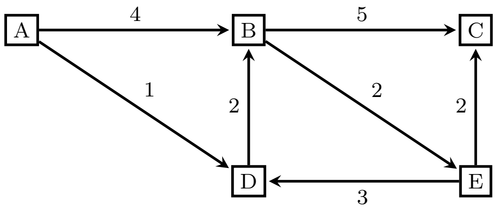
TD 2/7 – Paths and trees

Training exercises

Note: The solution elements given here are not complete. Their purpose is only to guide you. We encourage you to write a proper answer, as you would do for the exam. If any question remains, feel free to ask your tutorial supervisor for help.

Exercice 1 : Application of the shortest paths algorithm

Apply the shortest paths algorithm to the following graph starting with node A.



Solution elements :

Here are the paths and the distance for each arrival node:

- D: A→D (1)
- B: A→D→B (3)
- E: A→D→B→E (5)
- C: A→D→B→E→C (7)

Exercice 2 : Bandwidth problem

We are interested in the data transmission in a computer network. The network is represented by a directed graph $G(V, E)$, where each router corresponds to a vertex of V. A transmission speed in MB/s is associated to each edge $(i, j) \in E$, which represents the quality of the connection from i to j.

Consider u and v, which are any two vertices of G. Let C be a path from u to v. The transmission speed on this path is that of the edge with the lowest transmission speed. Our goal is to find out, among all the possible paths from u to v, the one with the maximum transmission speed.

Question 1

Please propose an adaptation of the shortest paths algorithm.

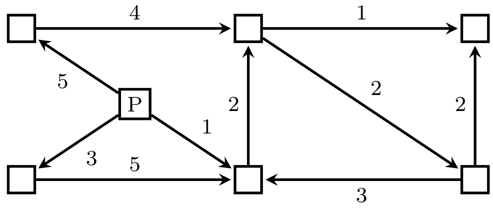
Solution elements :

You need to modify three steps in the shortest paths algorithm:

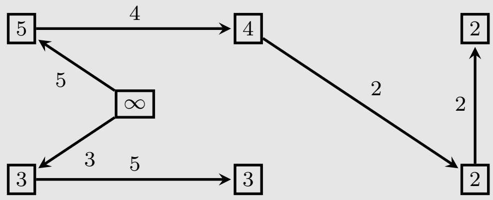
- the extraction of an element of the frontier
- the computation of the distance
- the comparison between distances

Question 2

Apply your algorithm to the graph below. For each vertex i, please give the optimal path that connects it to P as well as its transmission speed.



Solution elements :



Exercice 3 : Minimum spanning tree – dynamic version

Consider a positive weighted undirected graph $G = \langle V, E, w \rangle$ (with $w : E \to \mathbb{N}^+$) and a Minimal Spanning Tree (MST) T of G. We denote by extension $w(T) = \Sigma_{e \in T} w(e)$ the weight of T.

Suppose that the weight $w(e)$ of an edge $e = (x, y)$ with $e \in E \subseteq V \times V$ has changed since the computation of T. We have a new graph $G' = \langle V, E, w' \rangle$ with $w(e) \neq w'(e)$ and $\forall a \in E \setminus \{e\}.\ w(a) = w'(a)$.

We want to modify T such that it remains a minimal spanning tree. Of course, we could recalculate T by rerunning, for example, Kruskal's algorithm on G. It would cost $\mathcal{O}(m \times log(m))$ in time with $m = |E|$. Could we do better?

Question 1

How to update the minimum spanning tree T when the weight of an edge $e \in T$ is decreased by $w' = w(e) - w'(e) > 0$.

Solution elements :

Nothing to do here, T is still an MST.

Use a reductio ad absurdum.

Question 2

How to update T when the weight of an edge $e \notin T$ is increased by $w' = w'(e) - w(e) > 0$.

Solution elements :

Once again, nothing to do here, T is still an MST.

Use a reductio ad absurdum.

Question 3

How to update T when the weight of an edge $e \notin T$ is decreased by w'. Please propose an algorithm and give its complexity.

Solution elements :

Let $C \subseteq T \cup \{e\}$ be the cycle created by adding e in T and let e' be the edge of C with a weight $w'(e')$ maximal.

$T' = (T \setminus \{e'\}) \cup \{e\}$ is an MST of G'.

The proof is based on the optimality of Kruskal.

Here is the idea behind the proof:

- l'exécution de l'algorithme de Kruskal (qui prend les arêtes dans l'ordre de leur poids) sur G avec le nouveau poids $w'(e) < w'(e')$ est la même jusqu'à ce qu'on atteigne e,
- The execution of the Kruskal algorithm (that uses the edges by increasing weight order) on G with the new weight $w'(e) < w'(e')$ is the same until we reach e,
- e is accepted in the new tree. Reductio ad absurdum.
- The execution then continues in the same way until we reach e'
- e' is rejected because it would create a cycle
- The set of the connected component is now identical for T and T', the end of the execution is therefore the same.

The complexity of updating the MST for this case is $\mathcal{O}(|V|)$ as $|V| - 1$ is the number of edges in the tree.

Question 4

How to update T when the weight of an edge $e \in T$ is increased by w'. Please propose an algorithm and give its complexity.

Solution elements :

You need to consider the two sub-trees obtained by removing the updated edge. Then you need to select the edge with minimal weight among edges that cross the cut. Finally, you need to consider the cycle created by adding this edge and remove from the cycle the edge with maximal weight.

The complexity of updating the MST for this case is $\mathcal{O}(|E|)$.