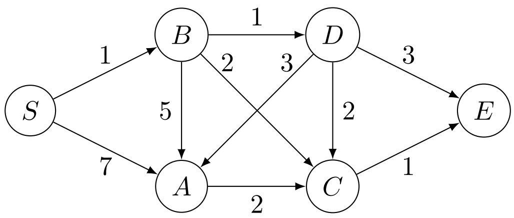
Tutorial 10 : Shortest paths, diameter

Graph theory, 1st semester.

2022

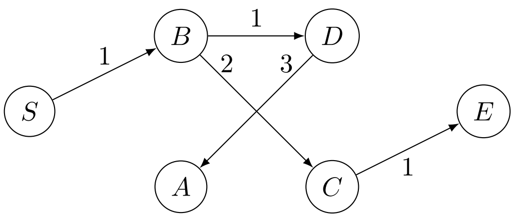
Exercise 1 — ***Dijkstra algorithm***

Let G be the following graph :



1. Draw a shortest path tree containing for each node a shortest path from S to that node.

▶ **Correction**



2. In which order the Dijkstra algorithm would enumerate all the nodes of the graph ?

▶ **Correction**

Dans l'ordre du poids d'un plus court chemin de S vers les autres nœuds : S, B, D, C, E, A

3. Apply the algorithm to compute the weight of the shortest paths from S to every other node.

▶ **Correction**

On obtient le tableau suivant où la première ligne est l'initialisation, la seconde correspond à la première itération, X est l'ensemble des nœuds qui n'ont pas encore été explorés, u est le nœud de X choisit pendant cette itération (celui qui n'est pas encore exploré et qui est le plus proche de S), et $d(v)$ est la meilleure distance de S à v calculée à la fin de l'itération.

Iteration	X	u	$d(S)$	$d(A)$	$d(B)$	$d(C)$	$d(D)$	$d(E)$
Avant la première	-	-	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	S, A, B, C, D, E	S	0	7	1	$+\infty$	$+\infty$	$+\infty$
2	A, B, C, D, E	B	0	6	1	3	2	$+\infty$
3	A, C, D, E	D	0	5	1	3	2	5
4	A, C, E	C	0	5	1	3	2	4
5	A, E	E	0	5	1	3	2	4
6	A	A	0	5	1	3	2	4

4. Change the algorithm in order to compute the shortest paths.

▶ **Correction**

On construit un second tableau contenant l'information $pred(v)$ qui est le prédecesseur de v dans un plus court chemin de S à v, (sauf pour S). Au début, $pred(v)$ est inconnu. Puis, quand on modifie la valeur de $d(v)$ lors de l'itération où u est choisit, on pose $pred(v) = u$.

Iteration	X	u	$pred(A)$	$pred(B)$	$pred(C)$	$pred(D)$	$pred(E)$
Avant la première	-	-	?	?	?	?	?
1	S, A, B, C, D, E	S	S	S	?	?	?
2	A, B, C, D, E	B	B	S	B	B	?
3	A, C, D, E	D	D	S	B	B	D
4	A, C, E	C	D	S	B	B	C
5	A, E	E	D	S	B	B	C
6	A	A	D	S	B	B	C

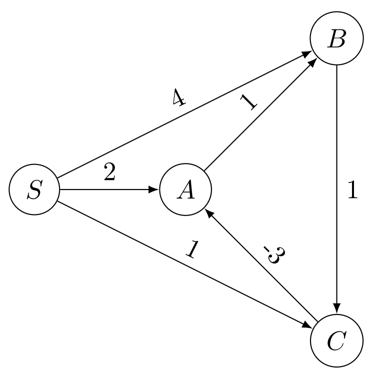
Exercise 2 — *Ford-Bellman algorithm*

1. Let G be a weighted graph and s be a node of G. Let $d^k(u)$ be the weight of a shortest path from s to u using at most k arcs. Show that there exists an absorbing directed cycle if and only if there exists a node u such that $d^{n-1}(u) > d^n(u)$.

▶ **Correction**

Si $d^{n-1}(u) > d^n(u)$, il existe un chemin de s à u utilisant n arcs et de poids strictement plus petit que le plus court chemin utilisant au plus $n-1$ arcs. Un chemin avec n arcs contient un circuit. Supposons que ce circuit ne soit pas absorbant. Alors je peux le supprimer du chemin pour obtenir un chemin au moins aussi bon et avec strictement moins d'arcs, disons k, ce qui est exclus car alors $d^{n-1}(u) \leq d^k(u) \leq d^n(u) < d^{n-1}(u)$.
Sinon, alors pour tout arc (u, v), $\omega(u, v) + d^{n-1}(u) \geq d^{n-1}(v)$. S'il existe un circuit, disons $(u_1, u_2, \ldots, u_p, u_{p+1} = u_1)$ alors $\sum_{i=1}^{p} \omega(u_i, u_{i+1}) + d^{n-1}(u_i) - d^{n-1}(u_{i+1}) \geq 0$. Or $\sum_{i=1}^{p} \omega(u_i, u_{i+1}) + d^{n-1}(u_i) - d^{n-1}(u_{i+1}) = \sum_{i=1}^{p} \omega(u_i, u_{i+1})$ donc le circuit n'est pas absorbant.

2. Apply the Bellman-Ford algorithm to the following graph.



▶ **Correction**

k	$d^k(S)$	$d^k(A)$	$d^k(B)$	$d^k(C)$
0	0	$+\infty$	$+\infty$	$+\infty$
1	0	2	4	1
2	0	-2	3	1
3	0	-2	-1	1
4	0	-2	-1	0

On constate qu'il y a un circuit absorbant car $d^3(C) > d^4(C)$

Exercise 3 — *Battery charge*

A small robot is moving on a grid. When it reaches a cell, it can turn left, turn right or go straight right but cannot move back. It cannot fall from the grid. If it does not move it consume 1 joule of energy. If it moves it consume 2 joules. It is also possible that the robot obtain 3 joules when moving because of induction wireless charger placed at some places under the grid. We know where are the chargers. The robot starts with a half empty battery. THis is sufficient to visit al least once every cell of the grid. We want to know if the robot may fully charge its battery by moving.

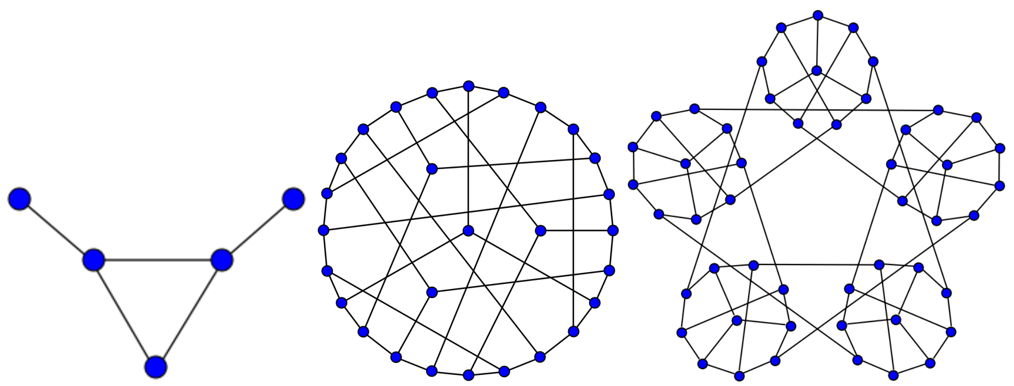
Model this problem with a graph problem.

▶ Correction

On peut le modéliser ainsi : à chaque case correspond 4 nœuds, un par orientation possible, représentant l'état *Le robot entre dans cette case avec cette orientation*, permettant de modéliser le fait que le robot ne peut pas revenir en arrière. On relie une case orientée à ses 3 cases voisines avec l'orientation correspondante.

Par exemple $(1, 1, N) \to (0, 1, W), (2, 1, E), (1, 2, N)$. On pondère un arc avec 2 s'il n'y a pas de chargeur sous le déplacement, et $2 - 3 = -1$ sinon. S'il existe un circuit absorbant alors le robot peut se recharger pleinement.

Exercise 4 — *Quelques diamètres* What is the diameter of the following graphs ?



(Source : wikipedia, par Koko90)

▶ Correction

3, 4 et 7

Exercise 5 — *Diamètre d'un arbre* ...