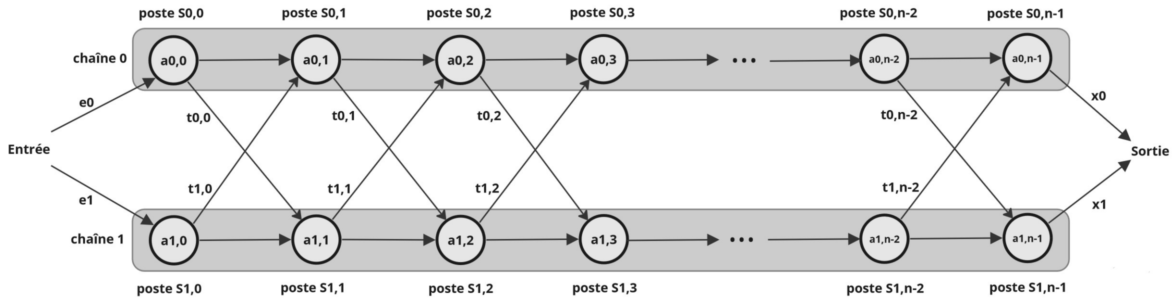
TP 4 PROGRAMMATION DYNAMIQUE

Exercice 1

Un constructeur automobile possède un atelier avec deux chaînes de montage comportant chacune n postes. Chaque véhicule doit passer par les n postes dans l'ordre. Le constructeur cherche à déterminer quels sont les postes à sélectionner sur la chaîne 0 et sur la chaîne 1 pour minimiser le délai de transit d'une voiture à travers l'atelier.
Chaque solution de ce problème d'optimisation est définie par le sous-ensemble de postes de la chaîne 0 utilisés (les postes restant sont choisis dans la chaîne 1). Il y a donc 2^n solutions possibles *i.e.* le nombre de sous-ensembles d'un ensemble à n éléments. Par conséquent, l'approche naïve consistant à considérer tous les chemins possibles est inefficace.
On note, pour tout $i \in [\![0,1]\!]$ et tout $j \in [\![0, n-1]\!]$,

— $S_{i,j}$ le poste j de la chaîne i ;
— e_i le temps d'entrée d'un véhicule sur la chaîne i ;
— $a_{i,j}$ le temps de montage pour le poste j sur la chaîne i ;
— $t_{i,j}$ le temps de transfert d'un véhicule de la chaîne i vers l'autre chaîne après le poste $S_{i,j}$;
— x_i le temps de sortie d'un véhicule de la chaîne i.

On peut représenter ces valeurs à l'aide d'un graphe



La programmation dynamique permet de résoudre ce problème efficacement.

▹ On identifie des sous-problèmes dont les solutions optimales vont nous permettre de reconstituer une solution optimale du problème initial.
  Les sous-problèmes à considérer ici consistent à calculer un itinéraire optimal jusqu'à chacun des postes $S_{i,j}$.
  Par exemple, considérons un itinéraire optimal jusqu'au poste $S_{0,j}$, celui-ci est
  - un chemin direct de l'entrée au poste $S_{0,0}$ (si $j = 0$),
  - soit un chemin optimal jusqu'au poste $S_{0,j-1}$ suivi du poste $S_{0,j}$ (si $j > 0$),
  - soit un chemin optimal jusqu'au poste $S_{1,j-1}$ suivi d'un changement de chaîne et du poste $S_{0,j}$ (si $j > 0$).

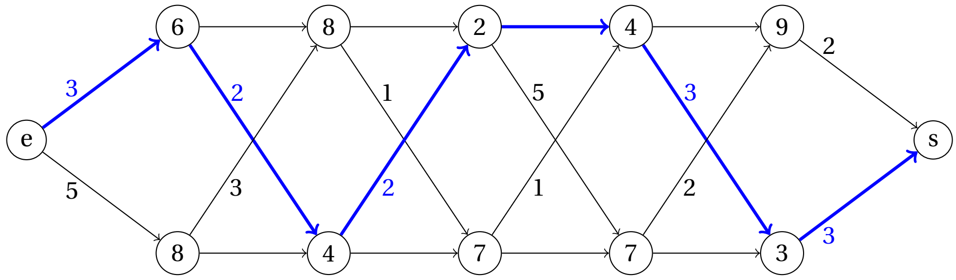
▹ On définit la valeur optimale de manière récursive à partir des valeurs des solutions optimales des sous-problèmes.
  Notons $f_i(j)$ le délai optimal jusqu'à $S_{i,j}$ et f^* le délai optimal total. Pour traverser l'atelier, il faut atteindre ou bien $S_{0,n-1}$ ou bien $S_{1,n-1}$ et sortir de l'atelier. Par conséquent :

$$f^* = \min(f_0(n-1) + x_0, f_1(n-1) + x_1) \qquad \text{avec} \begin{cases} f_0(0) = e_0 + a_{0,0} \\ f_1(0) = e_1 + a_{1,0} \end{cases}$$

et, pour tout $j \in [\![1, n-1]\!]$:

$$f_0(j) = \min\big(f_0(j-1) + a_{0,j}, f_1(j-1) + t_{1,j-1} + a_{0,j}\big),$$
$$f_1(j) = \min\big(f_1(j-1) + a_{1,j}, f_0(j-1) + t_{0,j-1} + a_{1,j}\big).$$

1. Écrire une fonction itérative `chaine(a,t,e,x)` d'arguments quatre listes correspondant aux données de l'énoncé et qui renvoie le délai optimal f^*.
2. Pour pouvoir reconstruire les solutions optimales elles-mêmes, on définit $\ell_0(j)$ et $\ell_1(j)$, pour $j \in [\![1, n-1]\!]$, le numéro de la chaîne (0 ou 1) dont le poste $j-1$ est utilisé par un chemin optimal jusqu'au poste $S_{i,j}$.
   On note ℓ^* le numéro de la chaîne utilisée par un chemin optimal lors du dernier poste.
   Par exemple, pour les données suivantes :



   les solutions optimales associées sont :

j	0	1	2	3	4
$f_0(j)$	9	17	19	23	32
$f_1(j)$	13	15	22	29	29

$f^* = 32$

j	1	2	3	4
$\ell_0(j)$	0	1	0	0
$\ell_1(j)$	0	1	1	0

$\ell^* = 1$

   Adapter la fonction de la question précédente afin qu'elle précise également le chemin optimal.

Exercice 2

En partant du sommet du triangle ci-dessous et en se déplaçant vers les nombres adjacents de la ligne inférieure, le total maximum que l'on peut obtenir pour relier le sommet à la base est égal à 23 :

3
7 4
2 **4** 6
8 5 **9** 3

$3 + 7 + 4 + 9 = 23.$

On modélise ce triangle à l'aide d'une liste de listes.

1. Pour $(i, j) \in [\![0, n-1]\!]^2$ avec $i \leqslant j$, on note $S_{i,j}$ le plus grand total en partant du coefficient à la ligne i colonne j.
   Que vaut $S_{n-1,j}$? Donner une relation pour $0 \leqslant j \leqslant i \leqslant n-2$, une formule reliant $S_{i,j}$, $S_{i+1,j}$ et $S_{i+1,j+1}$.
2. Programmer une fonction calculant le total maximal pour un chemin reliant le sommet du triangle à sa base.
3. Modifier cette fonction pour renvoyer un chemin qui réalise cette somme.
   On codera le chemin par une liste des colonnes visitées depuis la ligne 1 jusqu'à la ligne n.