

## TECHNIQUES ALGORITHMIQUES ET PROGRAMMATION

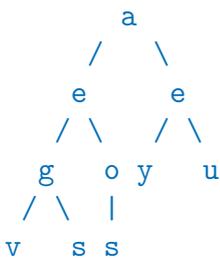
Test de contrôle continu  
1h - aucun document autorisé

---

### Tas minimum de lettres

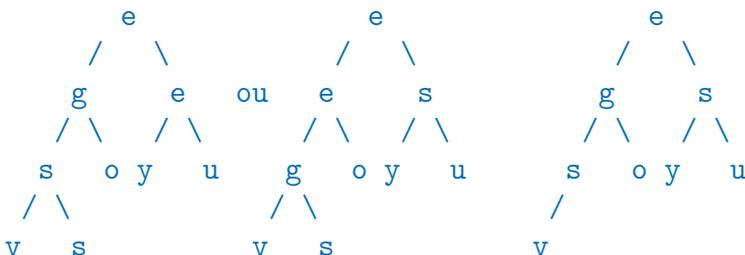
**Question 1.** *Donnez le tas résultant de l'insertion dans un tas minimum supposé vide au départ des 10 lettres du mot **voyageuses** ajoutées une par une en le lisant de gauche à droite. On trie les lettres selon l'ordre alphabétique. Par exemple :  $g < s$  et  $o > e$ .*

**SOLUTION.** [4 pts]



**Question 2.** *Appliquer deux fois l'opération de dépilement (**pop**) au tas de la question précédente. Donner les tas obtenus après l'exécution de chacune des deux opérations.*

**SOLUTION.** [3 pts (avec question suivante)]

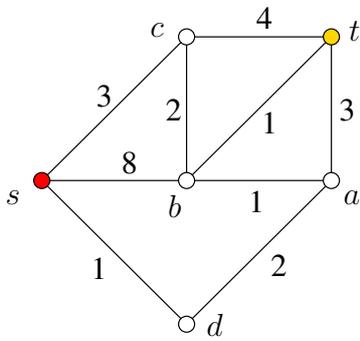


**Question 3.** *Quelles sont dans l'ordre les deux lettres ainsi supprimées ?*

**SOLUTION.** a puis e.

### Algorithme A\*

On considère le graphe valué suivant, ainsi que les deux heuristiques  $h_1$  et  $h_2$  estimant la distance entre un sommet donné et  $t$  :



	$(s, t)$	$(a, t)$	$(b, t)$	$(c, t)$	$(d, t)$	$(t, t)$
$h_1$	3	1	1	2	2	0
$h_2$	5	2	1	1	6	0

**Question 4.** Démontrez que si une heuristique  $h$  vérifie l'inégalité triangulaire et sous-estime la distance, alors  $h$  est monotone.

**SOLUTION.** [4 pts] Il faut montrer que inégalité triangulaire + sous-estimation implique monotonie. On a :

$$\begin{aligned}
 h(x, t) &\leq h(x, y) + h(y, t) && \text{(inégalité triangulaire)} \\
 &\leq \text{dist}(x, y) + h(x, t) && \text{(} h() \text{ sous-estime dist())} \\
 &\leq \omega(x, y) + h(x, t) && (\omega(x, y) = \text{coût d'un chemin de } x \text{ à } y)
 \end{aligned}$$

Ce qui est la définition de la monotonie.

**Question 5.** Appliquez l'algorithme  $A^*$  sur le graphe ci-dessus entre  $s$  et  $t$  en utilisant l'heuristique  $h_1$ . Indiquez dans la table ci-dessous dans quel ordre les sommets sont ajoutés à  $P$  en donnant à chaque fois  $\text{parent}[u]$ ,  $\text{coût}[u]$  et  $\text{score}[u]$  au moment de l'ajout de  $u$  dans  $P$ .

$P$	$s$	
parent	$\perp$	
coût	0	
score	3	

Donnez le chemin de  $s$  à  $t$  ainsi construit et son coût.

**SOLUTION.** [3 pts]

$P$	$s$	$d$	$a$	$c$	$b$	$t$
parent	$\perp$	$s$	$d$	$s$	$a$	$b$
coût	0	1	3	3	4	5
score	3	3	4	5	5	5

Note : En fait, le  $c$  est facultatif, puisque il a le même score que  $b$  et  $t$ . Mais si on fait le tas normalement, c'est bien lui qui sera au sommet.

Avec  $h_1 : s \rightarrow d \rightarrow a \rightarrow b \rightarrow t$ , et  $\text{coût}[t] = 5$ .

**Question 6.** Mêmes questions en utilisant l'heuristique  $h_2$ .

$P$	$s$	
parent	$\perp$	
coût	0	
score	5	

Donnez le chemin de  $s$  à  $t$  ainsi construit et son coût.

SOLUTION. [3 pts]

$P$	$s$	$c$	$b$	$t$
parent	$\perp$	$s$	$c$	$b$
coût	0	3	5	6
score	5	4	6	6

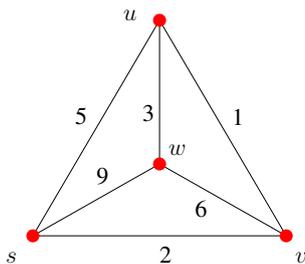
Avec  $h_2 : s \rightarrow c \rightarrow b \rightarrow t$ , et  $\text{coût}[t] = 6$ .

Question 7. Comparez le coût des deux chemins, expliquez.

SOLUTION. [2 pts] Le chemin obtenu avec  $h_2$  n'est pas optimal, car  $h_2$  n'est pas monotone :  $h_2(d, t) = 6$  alors que  $w(d, a) + h_2(a, t) = 4$ . Alternativement,  $h_2$  ne sous-estime pas les distances non plus :  $\text{dist}_G(d, t) = 4$ .

## Voyageur de commerce

On considère le problème du voyageur de commerce sur un ensemble de 4 points  $V = \{s, u, v, w\}$  dont la distance  $d(\cdot, \cdot)$  est donnée par la table et le graphe ci-dessous :



$d$	$s$	$u$	$v$	$w$
$s$	0	5	2	9
$u$	5	0	1	3
$v$	2	1	0	6
$w$	9	3	6	0

Dans la suite on notera  $V^* = V \setminus \{s\} = \{u, v, w\}$ . L'algorithme de programmation dynamique vu en cours repose sur la fonction  $D(t, S)$ , définie pour tout sous-ensemble de points  $S \subseteq V^*$  et tout point  $t \in S$ , comme étant la longueur minimum d'un chemin allant de  $s$  à  $t$  et qui visite tous (et seulement) les points de  $S$ . On rappelle que cette fonction vérifie la formule de récurrence suivante :

$$D(t, S) = \begin{cases} d(s, t) & \text{si } |S| = 1 \\ \min_{x \in S \setminus \{t\}} \{D(x, S \setminus \{t\}) + d(x, t)\} & \text{si } |S| > 1 \end{cases}$$

La table ci-dessous doit accueillir toutes les valeurs de  $D(t, S)$ . Les lignes de la table représentent les points ( $t$ ) et les colonnes les sous-ensembles ( $S$ ). La première colonne est déjà remplie, en notant que certaines cases n'ont pas à être remplies.

$S$	$\{u\}$	$\{v\}$	$\{w\}$	$\{u, v\}$	$\{u, w\}$	$\{v, w\}$	$\{u, v, w\}$
$t$							
$u$	5						
$v$	X						
$w$	X						

**Question 8.** Indiquez par une croix les cases qui ne doivent pas être remplies.

**Question 9.** Complétez les autres cases de la table.

SOLUTION. [1+4 pts (avec la question précédente)]

$S$ $t$	$\{u\}$	$\{v\}$	$\{w\}$	$\{u, v\}$	$\{u, w\}$	$\{v, w\}$	$\{u, v, w\}$
$u$	5	X	X	3	12	X	11
$v$	X	2	X	6	X	15	13
$w$	X	X	9	X	8	8	6

**Question 10.** Expliquez comment on calcule à l'aide de cette table la longueur de la tournée optimale. Donnez cette longueur.

SOLUTION. [2 pts] On calcule la longueur grâce à la dernière colonne, celle correspondant à  $S = V^* = \{u, v, w\}$ . Il faut ajouter le retour  $d(t, s)$  avec  $t = u, v$ , ou  $w$ .

$$\text{OPT}(V, d) = \min_{t \in \{u, v, w\}} \{D(t, V^*) + d(t, s)\} .$$

- $D(u, V^*) + d(u, s) = 11 + 5 = 16$ .
- $D(v, V^*) + d(v, s) = 13 + 2 = 15$ .
- $D(w, V^*) + d(w, s) = 6 + 9 = 15$ .

La longueur minimale est donc 15.