

NOM :

PRENOM :

## SOLUTION CONTROLE I.A. (I4RSD11) - 15/03/2023

- Durée 1h15
- Documents non autorisés. Seule une **fiche de révision A4 recto-verso** est autorisée.
- Le sujet doit être rendu avec la copie, vous pouvez vous servir des schémas fournis.

### EXERCICE 1 : ALGORITHME A\* (9 PT)

#### 1.1 Fonctions $g$ et $h$

a.  $g(u_0) = 0$  pour  $u = u_0$ .       $g(v) = g(u) + k(u,v)$  pour  $\forall v \in \text{Suc}(u)$

b.  $h(A) = 100$ .

c.  $h(B) = 0 \Rightarrow h$  est *coïncidente*.

d.  $h$  est-elle monotone ?

$k(u, v)$  = poids du container déplacé. Chaque action ne déplace qu'un seul container.

Cas où l'action place correctement le container :  $h(u) - h(v) = k(u,v)$

Cas où l'action déplace un container qui était bien placé :  $h(u) - h(v) = -k(u,v)$

Cas où l'action manipule un container mal placé vers un emplacement où il est toujours mal placé :

$$h(u) - h(v) = 0$$

Dans les 3 cas, on a  $h(u) - h(v) \leq k(u,v)$  donc  $h$  est monotone.

Comme  $h$  est également coïncidente,  $h$  est *minorante*.

Démonstration directe :

Chaque action ne déplace qu'un seul container à la fois et ne peut donc bien placer qu'un seul container à la fois.

Il faudra donc au moins autant d'actions que de containers mal placés pour arriver à une solution. Le coût de ces actions sera égal au poids total de tous les containers déplacés.

Ceci est un minimum car à une étape donnée il est parfois nécessaire de déplacer un container mal placé sans pour autant pouvoir le déposer directement à la position attendue, celle-ci est étant occupée par un autre container.

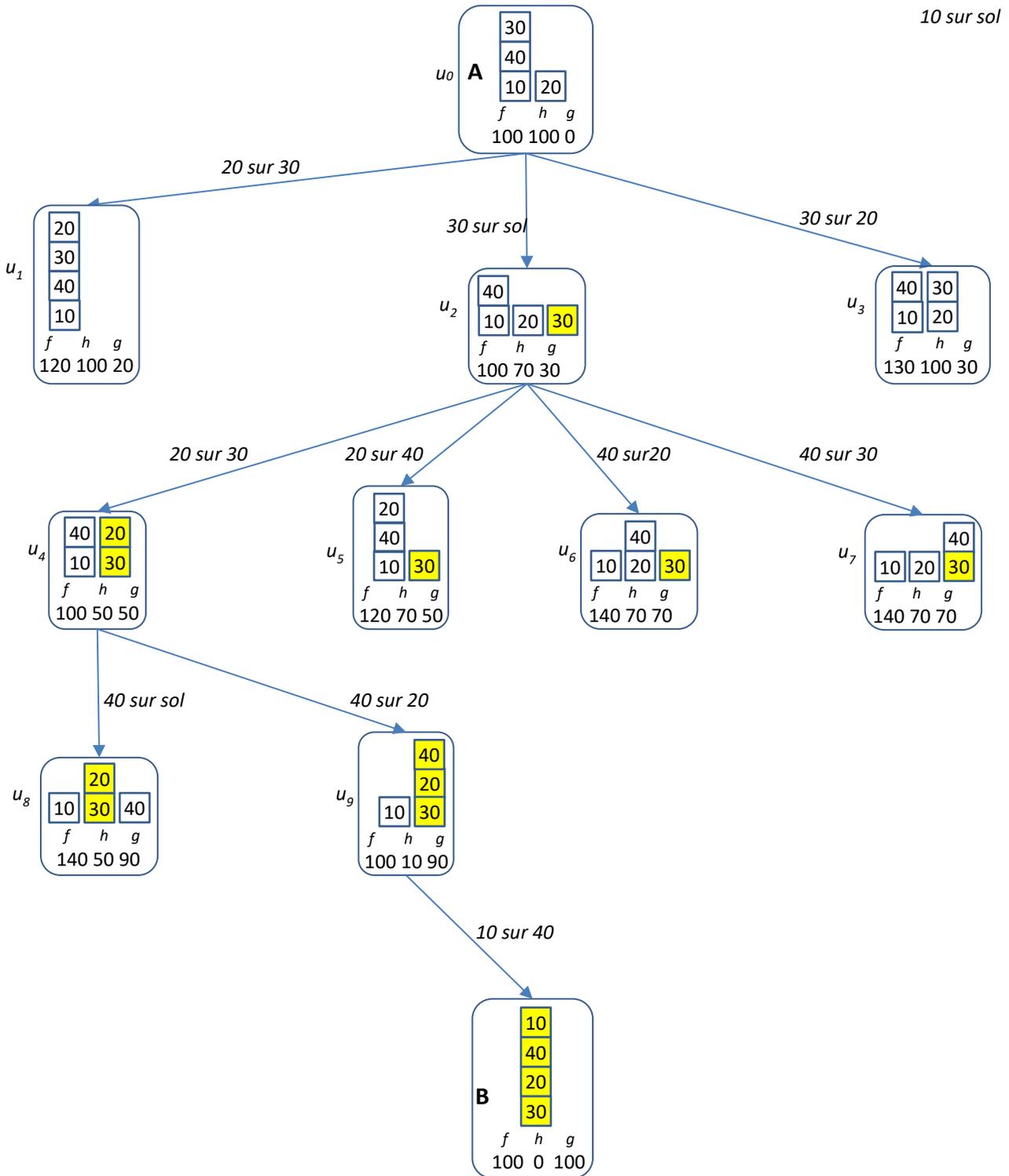
L'estimation est donc parfois exacte mais parfois optimiste  $\Rightarrow$  c'est un **minorant**.

#### 1.2 Arbre généré et exploré par A\*

Voir page suivante

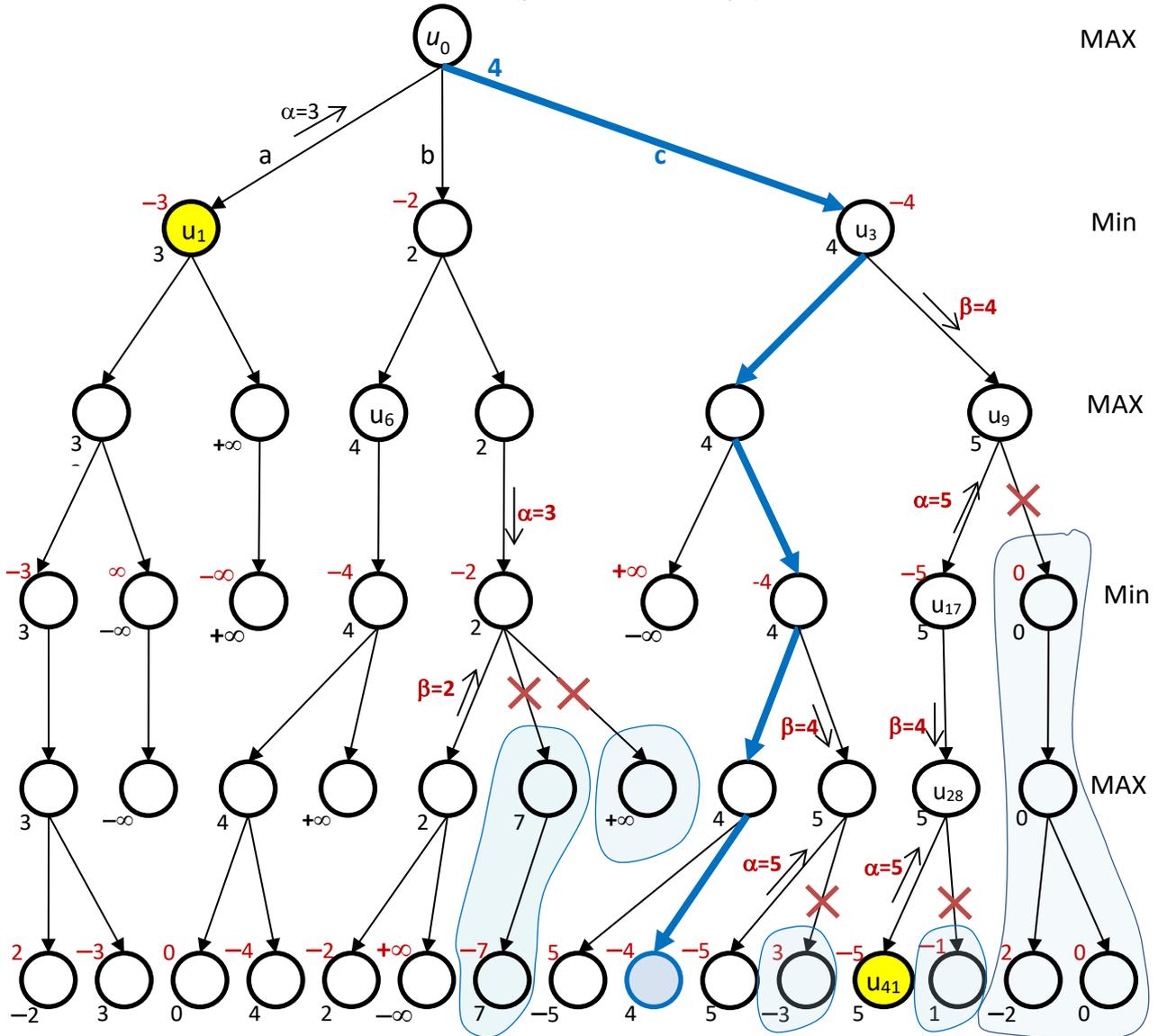
NOM :

ARBRE EXPLORÉ PAR A\*



**EXERCICE 2 : ALGORITHME MINMAX/NEGAMAX ET COUPURES ALPHA-BETA (5 PT)**

ARBRE MINMAX(profondeur 5 coups)



2.1 Le meilleur coup est **c** d'évaluation = **4** (voir ci-dessus).

2.2 Voir valeurs negamax (ci-dessus).

2.3 Voir coupures alpha-beta (ci-dessus).

2.4 Il faut recopier l'évaluation de  $u_1$  (3) en  $u_{41}$  (voir page 4).

Conséquences :

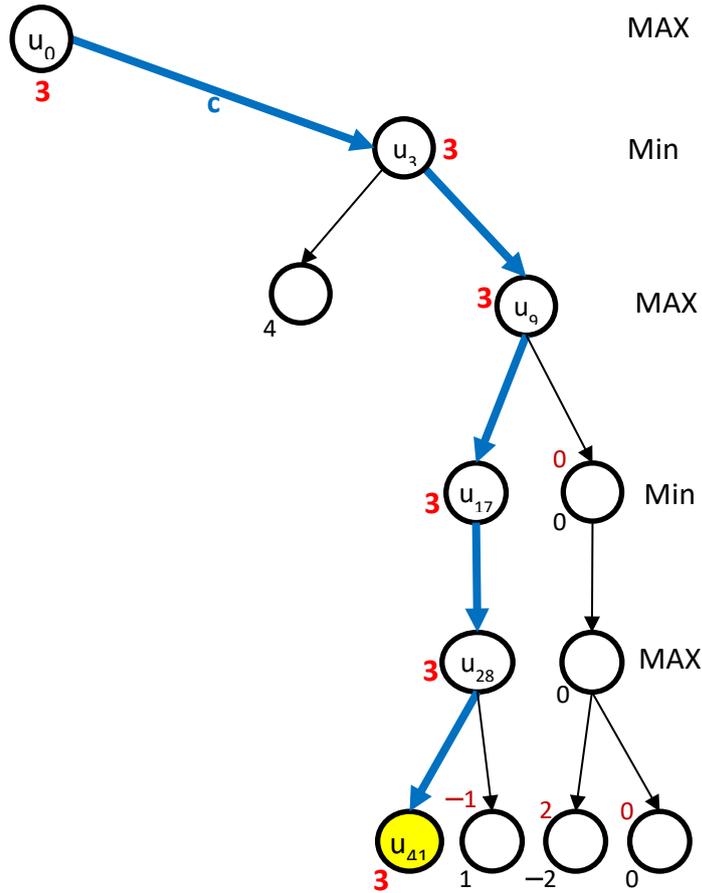
La valeur 3 remonte en  $u_{28}$  puis en  $u_{17}$  puis en  $u_9$  et en  $u_3$ . Le coût de **c** devient égal à 3, ce qui rend les coups **a** et **c** équivalents.

**a** devient donc le meilleur premier coup retourné par l'algorithme.

Les 2 dernières coupures alpha-beta (les plus à droite) n'auraient pas lieu.

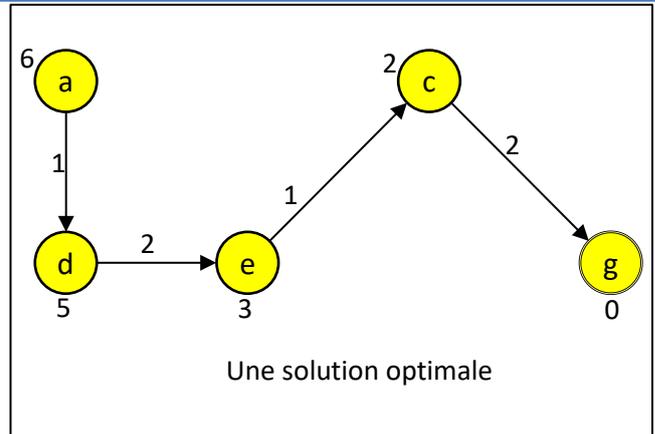
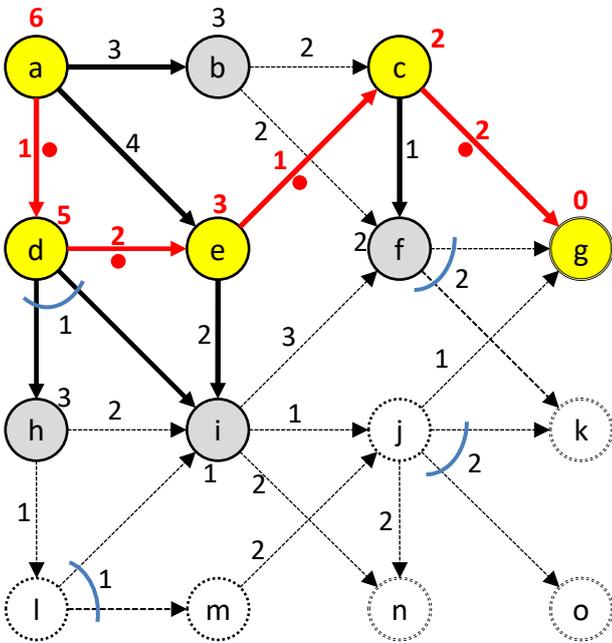
NOM :

PRENOM :



EXERCICE 3 : ALGORITHME AO\*

(6 PT)



- nœud développé et résolu
- nœud non développé mais dont l'heuristique a été utilisée
- nœud non atteint
- connecteur marqué optimal
- connecteur exploré
- connecteur non exploré

NOM :