

# Rapport TP2 ACT

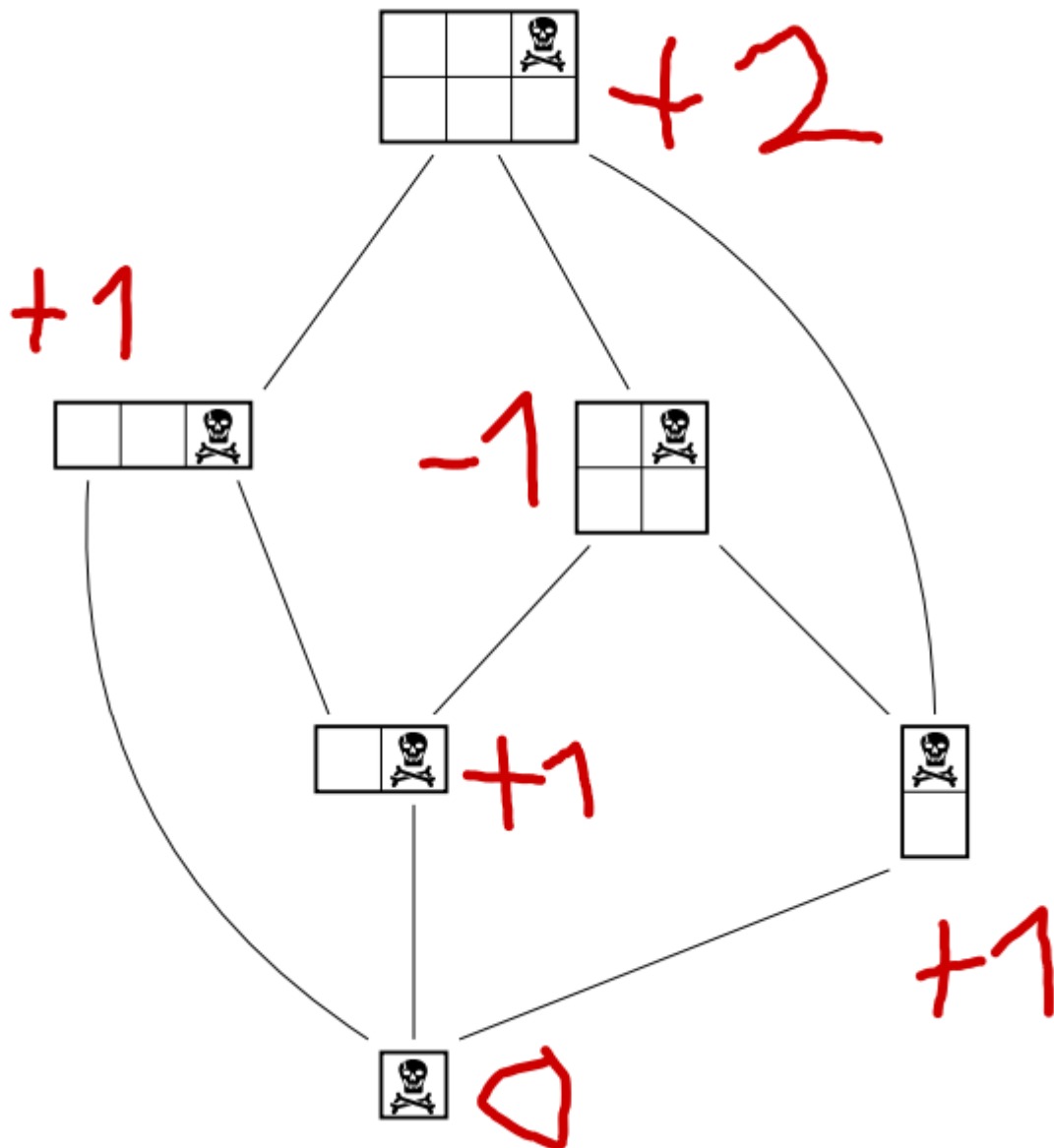
Gaspar Henniaux - Marwane Ouaret

Lien Github : <https://github.com/pargass/ACT-TP/tree/main/tp2>

## Question 1

Pour définir une grille de manière unique il faut le nombre de lignes, le nombre de colonnes et les coordonnées de la case piégée

## Question 2



PROF

à justifier...

## Question 3

étant donnée une configuration  $(m, n, i, j)$ , pour obtenir tous ses successeurs on peut appliquer les règles suivantes :

à écrire... (voir algo)

## Question 4

Pour calculer la valeur d'une configuration à partir des valeurs de ses successeurs, on peut appliquer la formule suivante :

- si parmi les successeurs il y a des valeurs négatives, alors la valeur de la position s'obtient en prenant la valeur absolue de la plus haute valeur négative et d'ajouter 1
- sinon, la valeur de la position s'obtient en prenant l'opposée de la plus haute valeur positive (dans l'idée de retarder la défaite au maximum)

## Question 5

```
function position_value(m, n, i, j):  
    si il ne reste qu'une case alors  
        return 0  
    sinon  
        valeur_successeurs = valeur de tous les successeurs de la  
position (m, n, i, j)  
        si il y a des valeurs négatives dans valeur_successeurs ou 0  
alors  
            return abs(valeur négative la plus haute) + 1  
        sinon  
            return -valeur positive la plus haute
```

## Question 6

PROF

pour la configuration  $(10, 7, 7, 3)$ , le temps d'exécution est de 192 secondes tandis que pour la configuration  $(10, 7, 5, 3)$  le temps d'exécution est de 414 secondes

## Question 7

Cette différence s'explique par le fait que dans le second cas plein de positions sont recalculées plusieurs fois, ce qui n'est pas le cas dans le premier cas

la complexité de cet algorithme est exponentielle car dans la boucle principale on appelle la fonction `position_value`  $m + n - 2$  fois