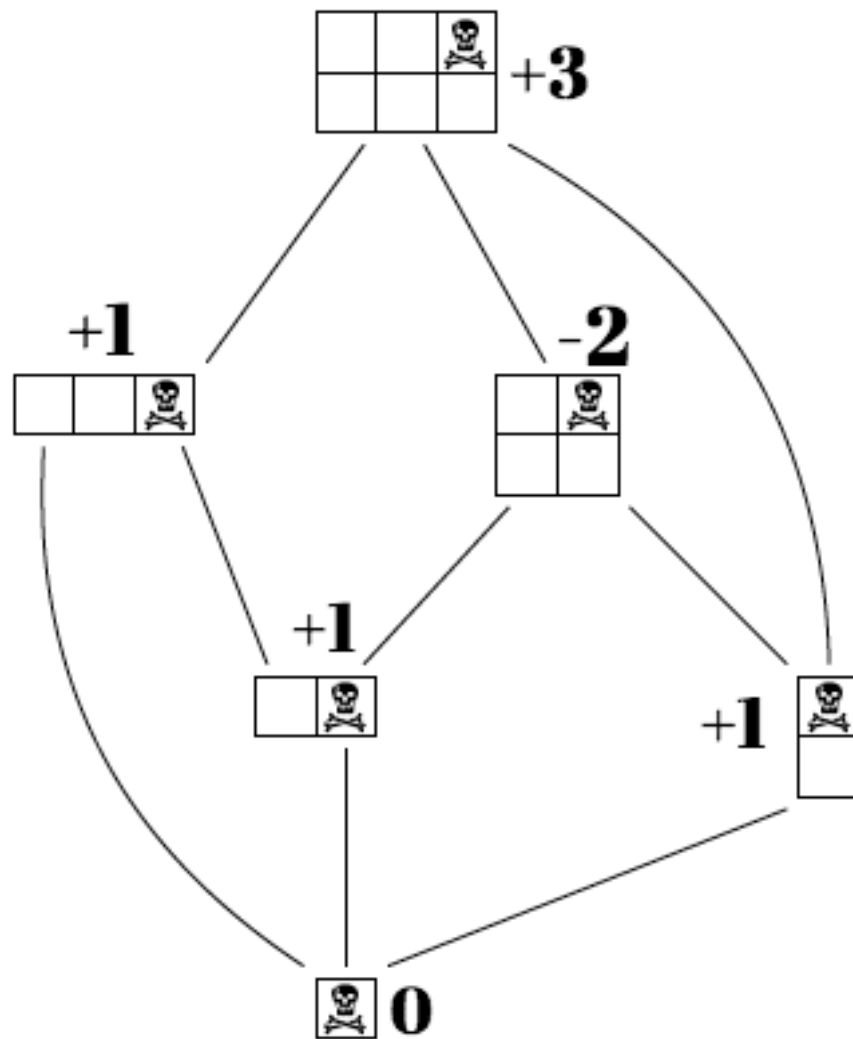


TP2 Rendu

--SHA Qianqian, MA Ling

QUESTION 1 :



QUESTION 2 :

Formule mathématique:

ranger les successeurs(x) de le plus petit au plus grand.

S'il existe le successeur qui est négative ou 0.

$$\text{Valeur} = (\text{abs}(\max(\text{successeurs}(x))) + 1)$$

Si tous les successeurs sont positive.

$$\text{Valeur} = -(\max(\text{successeurs}(x)) + 1)$$

QUESTION 3 : (Version Naïve)

Réponse :

```
le resultat de (10, 7, 7, 3) = 11  
le temps de (10,7,7,3): 13s  
le resultat de (10, 7, 5, 3) = 15  
le temps de (10,7,5,3): 28s
```

Pseudo – Code:

Les façons de coupe le chocolat :

1.Coupe la partie à la gauche de ce chocolat

```
for(int k = 1; k<=i; k++) add(m-k,n,i-k,j);
```

2.Coupe la partie à la droite de ce chocolat

```
for(int k = 1; k< m-i; k++)add(m-k,n,i,j);
```

3. Coupe la partie en haut de ce chocolat

```
for(int k = 1; k<=j; k++) add(m,n-k,i,j-k);
```

4. Coupe la partie en bas de ce chocolat

```
for(int k = 1; k < n-j; k++).add(m,n-k,i,j);
```

Obtenir le résultat du successeur

ranger les successeurs(x) de le plus petit au plus grand.

S'il existe le successeur qui est négative ou 0.

$$\text{Valeur} = (\text{abs}(\max(\text{successeurs}(x))) + 1)$$

Si tous les successeurs sont plus de 0.

$$\text{Valeur} = -(\max(\text{successeurs}(x)) + 1)$$

Itération :

Tout le chocolat a beaucoup de façons de couper, et chaque partie a beaucoup de façons de couper aussi. L'itération arrête quand ce chocolat laisse juste un morceau avec le petit gens dedans.

QUESTION 4 (version dynamique) :

Réponse :

Le résultat de (100,100,48,52): 191
le temps de (100,100,48,52): 67s
Le résultat de (100,100,50,50): -198
le temps de (100,100,50,50): 66s

Pseudo – Code:

Les façons de coupe le chocolat :

```
1.Coupe la partie à la gauche de ce chocolat
for(int k = 1; k<=i; k++) succ.add(m-k,n,i-k,j);
2.Coupe la partie à la droite de ce chocolat
for(int k = 1; k< m-i; k++)succ.add(m-k,n,i,j);
3. Coupe la partie en haut de ce chocolat
for(int k = 1; k<=j; k++) succ.add(m,n-k,i,j-k);
4. Coupe la partie en bas de ce chocolat
for(int k = 1; k < n-j; k++).succ.add(m,n-k,i,j);
```

Faire un tableau pour conserver les résultats:

```
tab = new Integer[m+1][n+1][x+1][y+1];
tab[1][1][0][0] = 0;
```

La méthode de calculer les résultats sont comme le Question 3.

QUESTION 5 (version sym)

Essayer tous les points de ce tableau, et conserver les points quand son résultat est 127.

```
(127, 127, 0, 63) egal a 127
(127, 127, 126, 63) egal a 127
(127, 127, 0, 63) egal a 127
(127, 127, 126, 63) egal a 127
(127, 127, 63, 0) egal a 127
(127, 127, 63, 126) egal a 127
(127, 127, 63, 0) egal a 127
(127, 127, 63, 126) egal a 127
```

QUESTION 6 :

L'ordre de grandeur de la complexité est $m^2 * n^2$.

Justifier:

pour obtenir quels i et j donnent les positions(par exemple 127). ----- $O(m*n)$

pour obtenir une valeur précise ,on calcule toutes i et j ----- $O(m*n)^2$.

Notre tableau pour stoker la resultat est 4 dimension aussi. ----- $O(m^2*n^2)$

Donc c'est $O(m^2*n^2)$

QUESTION 7 :

- 1) Les premières 4 configurations sont symétries à l'axe m ou n. On peut seulement changer la direction de chocolat.
- 2) Les suivants 4 configurations sont rotations des 4 premières configurations. On peut retourner 90 degree pour changer son direction.

Symétries et rotations n'influent pas ses valeur,. leurs tableus ont la même taille. Donc tous les 8 points ont les mêmes successeurs.

QUESTION 8 :

Oui, si on profiter les transformations obtenu de Q7, et stocker les 8 résultats calculé, on aura 8 fois plus vites.

QUESTION 9 :

Parce que en effet les quarte points symétries ont le même résultat. Quand la longueur de tableau est plus longue que la largeur, on essaye juste les points au coin de la gauche de ce tableau. Si on trouve un point qui a le résultat comme on veut, c'est à dire que les autres quartes points ont le même résultat. Quand la longueur de tableau est plus courte que la largeur, on tourne d'abord ce tableau et puis fait la même chose.

```
tab = new Integer[m+1][n+1][(m+1)/2][(n+1)/2];
tab[1][1][0][0] = 0;
Set< Sym> games = new HashSet< Sym>();
    for (int i = 0; i < (m+1)/2; i++) {
        for (int j = 0; j < (n+1)/2; j++) {
            int res = new Sym(m, n, i, j).dynamique();
            if (res == value) {
                games.add(new Sym(m, n, i, j));
                games.add(new Sym(m, n, m-(i+1), j));
                games.add(new Sym(m, n, i, n-(j+1)));
                games.add(new Sym(m, n, m-(i+1), n-(j+1)));
                games.add(new Sym(n, m, j, i));
                games.add(new Sym(n, m, j, m-(i+1)));
                games.add(new Sym(n, m, n-(j+1), i));
                games.add(new Sym(n, m, n-(j+1), m-(i+1)));
            }
        }
    }
```

QUESTION 10 (Version Game)

Description:

Joueur désigne la taille de tableau lui-même, et le position de ce petit gens va être n'importe où. On ne dessine pas le tableau et utilise la coordonnée à représenter ce chocolat. On a deux joueurs, pierre et John, pierre coupe ce chocolat d'abord.

```
please design your game
```

```
What is the length of your chocolate ?
```

```
8
```

```
What is the width of your chocolate ?
```

```
5
```

```
Possible situations :
```

1. (7, 5, 3, 4)
2. (6, 5, 2, 4)
3. (5, 5, 1, 4)
4. (4, 5, 0, 4)|
5. (5, 5, 4, 4)
6. (6, 5, 4, 4)
7. (7, 5, 4, 4)
8. (8, 4, 4, 3)
9. (8, 3, 4, 2)
10. (8, 2, 4, 1)
11. (8, 1, 4, 0)

```
Pierre,please enter the number what you want to choose >>>
```

Après la coupe de Pierre, John peut couper ce chocolat.

```
Pierre,please enter the number what you want to choose >>>5
```

```
Possible situations :
```

1. (4, 5, 3, 4)
2. (3, 5, 2, 4)
3. (2, 5, 1, 4)
4. (1, 5, 0, 4)
5. (5, 4, 4, 3)
6. (5, 3, 4, 2)
7. (5, 2, 4, 1)
8. (5, 1, 4, 0)

```
John,please enter the number what you want to choose >>>
```

La dernière fois, quand il y a juste deux morceaux de chocolats, et pierre mange le morceau sans le petits John et donne le morceau reste à John, John échoue ce jeu.

Possible situations :

1. (1, 1, 0, 0)

Pierre, please enter the number what you want to choose >>>1

Pierre, you are failed