QUESTION 1 :

+3

-2

+1

+1

+1

-0

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.

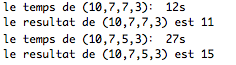
Valeur= (abs (max(successeurs(x))) +1)

Si tous les successeurs sont plus de 0.

Valeur= -(max(successeurs(x)) + 1)

QUESTION 3 :

Réponse :



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.

Valeur= (abs (max(successeurs(x))) +1)

Si tous les successeurs sont plus de 0.

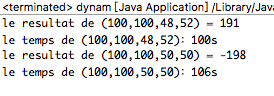
Valeur= -(max(successeurs(x)) + 1)

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:

Réponse :



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:

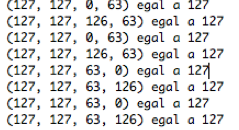
tab = new Integer[m+1][n+1][x+1][y+1];

tab[1][1][0][0] = 0;

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

QUESTION 5 :

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



QUESTION 6 :

L’ordre de grandeur de la complexité est n^4.

QUESTION 7 :

Parce que tous les 8 points ont les successeurs pareils.

QUESTION 8 :

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)));

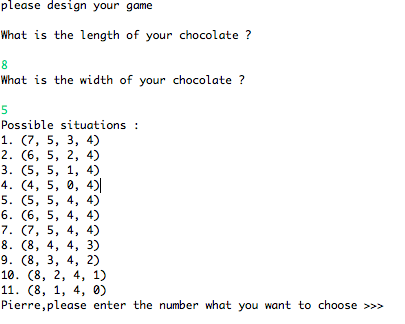
games.add(new Sym(n, m, n-(j+1),i));

games.add(new Sym(n, m, n-(j+1),m-(i+1)));

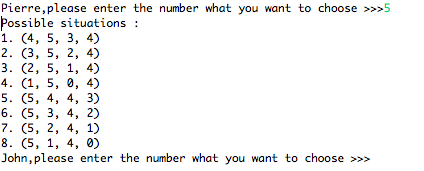
QUESTION 10 :

Description：

**Joueur désigne la taille de tableau lui-même, et le positon 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.**



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



**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.**屏幕快照%202017-10-14%20下午2.02.40.png