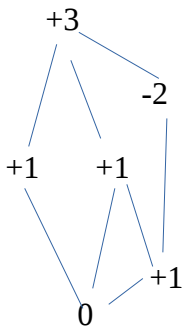
Compte Rendu TP2

Nom: KACI
Prénom : Sofiane
Groupe: 2
Q1)



Question 2:

S'il existe au moins un successeur avec une valeur négative,
prendre le successeur avec la plus grande valeur (On considère 0 comme un nombre négatif).
il n'y a aucun successeur avec une valeur négative,
prendre le successeur avec la plus grande valeur.

Question 3:

(10,7,7,3) met 5 secondes à s'exécuter
(10,7,5,3) met 11 secondes à s'exécuter

La complexité de cet algorithme(naïf) est exponentiel car à chaque récursion je fait 4 boucles et il faut résoudre (n+m)
sous problèmes qui exécutent autant de récursion que la taille de leur boucle

Question4 :

(100, 100, 50, 50) = -198
(100, 100, 48, 52) = 191

Question 5 :
les configurations qui vaut 127 comme résultat j'ai obtenu les configuration ci_dessous:

configuration: 127, 127, 126, 63 = 127
configuration: 127, 127, 63, 126 = 127
configuration: 127, 127, 63, 0 = 127

Question 6 :

le grandeur de la complexité est en O(n^4), car (m*m * n*n).

Question 7 :

logiquement sont les mémés,
admettant que la tablette du chocolat est une matrice à 2 Dimensions , on va mettre l'inverse pour chaque colonne de la matrice.

EXPLICATION:

Pour ce TP j'ai utilisé deux algorithme (naïf et dynamique) le but est de calculer le nombre deux coup pour qu'un jouer puisse perde ou gagner contre son adversaire sur un tablette de chocolat qui contient a carré de la mort. Mon programme a pour but de permettre au joueur de gagner le plus vite possible par exemple si y a une possibilité de gagner en 2 coups ou 6 coups on va choisir de gagner en 2 coup, ainsi on va essayer de perdre après plusieurs coups possibles exemple -1 ou -8 on choisir le -8 pour ne pas perdre facilement.

Implémentation:

Pour l'implémentation de deux algorithmes j'ai utilisé des configurations pour les découpages talques: avant la carre de la mort et après de la carre de la mort , à gauche de carre de la mort , à droite de carre de la mort Pour chaque algorithme.

Concernant l'algorithme naïf j'ai basé sur la récursivité afin de calculer toutes les configurations possibles du coup durant l'exécution on calculer certains valeurs plusieurs fois ce qui a pousser l'exécution a prendre du temps a chaque fois que la démissions de la tablette de chocolat soit augmenté.

Afin de résoudre le problèmes et l'optimisation du temps j'ai opté pour un algorithme dynamique ou j'ai utilisé une Hashmap qui prend deux paramétrés (configuration, un integer c'est le résultat de coup possibles). Le but de cette algorithme c'est que je stocke a chaque tour de boucle le résultat de chaque configuration afin de le vérifier pour le prochain tour de boucle, alors s'il est existe je vais pas faire l'appelle récursive je le stock directement dans la Hashmap Avec sa clé a coté bien sur.

Pour cela la complexité temporelle de l'algorithme ne sera pas élevé par rapport a l'algorithme naïf.