Andie SAMADOULOUGOU 300209487 **CSI2510** **Projet de Programmation 1 partie 2**

**Question 3**

1. **Algorithme**

/\*

L'idée de l'algorithme est la suivante :

prendre deux paires et échanger leurs éléments

Vérifier qu'il ne s'agit pas d'un meilleur mariage

le faire pour toutes les paires

\*/

**int [] echanger\_elements\_paires(**int tableau[],int indice\_premiere\_paire, int indice\_seconde\_paire**);**

//Par exemple (A,Y),(B,Z) devient (A,Z),(B,Y)

//cette fonction renvoie un tableau de taille n mais contenant seulement les paires échangées avec leurs indices respectés

//cela prend de l'espace certes mais au moins ce sera toujours plus simple de retrouver les bons indices

//une fonction qui determine si les paires sont plus heureuses

**boolean estPlusHeureux(**int employeur1, int employeur2, int etudiant1, int etudiant2, int A[][],int B[][]**)**

**{**

//trouver les rangs donnés respectivement par chancun aux autres

int rang\_emp1\_par\_etu1 = A[etudiant1][employeur1] ;

int rang\_emp1\_par\_etu2 = A[etudiant2][employeur1] ;

int rang\_emp2\_par\_etu2 = A[etudiant2][employeur2] ;

int rang\_emp2\_par\_etu1 = A[etudiant1][employeur1] ;

int rang\_etu1\_par\_emp1 = B[employeur1][etudiant1];

int rang\_etu1\_par\_emp2 = B[employeur2][etudiant1];

int rang\_etu2\_par\_emp1 = B[employeur1][etudiant2];

int rang\_etu2\_par\_emp2 = B[employeur2][etudiant2];

*//si le rang donné a l 'etudiant1 par l'employeur1 est plus petit que celui qu'il avait donné à //l'etudiant2 , alors il est plus satisfait*

*// Si le rang donné a l'employeur1 par l'etudiant1 est plus petit que celui qu'il a donné a l'etudiant2, //alors il est plus satisfait*

*// Meme principe pour la seconde paire*

return (rang\_etu1\_par\_emp1 < rang\_etu2\_par\_emp1 && rang\_emp1\_par\_etu1<rang\_emp2\_par\_etu1 && rang\_etu2\_par\_emp2<rang\_etu1\_par\_emp2 && rang\_emp2\_par\_etu2<rang\_emp1\_par\_etu2);

**}**

**boolean solution\_stable(**int employeurs[],int students[], int n ,int A[][] , int B[][]**)**

{

for(int i = 0 ;i < n;i++) // pour chaque paire

{

for(int j = i +1 ; j< n-1; j++) *// comparer avec les autres paire*s

{

*//echanger\_elements\_paires*

int []tableau = echanger\_elements\_paires(employeurs,i,j);

//tester si les 4 sont plus heureux dans la nouvelle distribution et retourner false si c'est le cas

if (estPlusHeureux(i,j,tableau[i],tableau[j],A,B)){**return false**;}

}

}

**return true**;

}

1. Complexité de l’algorithme

**int [] echanger\_elements\_paires(**int tableau[],int indice\_premiere\_paire, int indice\_seconde\_paire**); // O(1)**

**boolean estPlusHeureux(**int employeur1, int employeur2, int etudiant1, int etudiant2, int A[][],int B[][]**) // O(1)**

**boolean solution\_stable(**int employeurs[],int students[], int n ,int A[][] , int B[][]**)**

**complexité ici est O(1) \* (n-1) \* (n) soit O(n2)**

**L’algorithme a une complexité de O(n2) au pire cas**