Algorithmique - Informatique deuxième année

Réunir, trouver

On manipule des partitions de l'ensemble $\{0, \ldots, N-1\}$.
Les sous-ensembles sont représentés par des arbres en utilisant la relation `a pour pere`. On convient que la racine est son propre père.

Pour évaluer les améliorations apportées par *la fusion par rang* et *la compression des chemins* on écrira des versions des fonctions qui n'utilisent pas ces améliorations et d'autres les utilisant.
On comparera la complexité en pratique des différentes améliorations en comptant le nombre d'étapes effectuées par les appels successifs à la fonction de recherche du représentant.
On utilise :

```
#define PERE 0
#define RANG 1
static unsigned int cherchePere =0;
```

La partition est représentée par un tableau `int peres[N][2];` où `peres[i][PERE]` est l'indice du père de l'élément i et `peres[i][RANG]` est le rang de l'ensemble si i est le représentant d'un ensemble (non mis à jour sinon).
Par exemple la partition $\{0,2\},\{3\},\{1,4,5\}$ est représentée par :

0	1	0	3	1	1
2	3	.	1	.	.

1. Ecrire la fonction `void Initialisation(int peres[][2]);` qui initialise le tableau `peres` afin qu'il représente N ensembles réduits à 1 élément.

2. Ecrire la fonction `int Trouve(int peres[][2], int x);`
qui renvoie la valeur du représentant de l'ensemble contenant x (et augmente `cherhePere` du nombre d'étapes). Les chemins ne seront pas compressés.

3. Ecrire la fonction `void Fusion(int peres[][2], int x,int y);`
qui effectue la fusion (sans utiliser le rang) de l'ensemble contenant x et de celui contenant y. Si x et y ne sont pas dans le même ensemble, le représentant de l'ensemble contenant x deviendra celui de l'ensemble fusionné.

4. Ecrire la fonction `int TrouveCompresse(int peres[][2], int x);`
qui renvoie la valeur du représentant de l'ensemble contenant x (et augmente `cherhePere` du nombre d'étapes). Les chemins seront compressés.

5. Ecrire la fonction `void FusionRang(int peres[][2], int x,int y);`
qui effectue la fusion de l'ensemble contenant x et de celui contenant y. Si x et y ne sont pas dans le même ensemble, le représentant de l'ensemble de plus grand rang deviendra celui de l'ensemble fusionné.

6. Ecrire la fonction `int main(void)`
permettent de tester l'efficacité de l'amélioration apportée par `TrouveCompresse`et `FusionRang`.
On pourra effectuer:

- M appels aléatoires aux fonctions `Trouve` et `Fusion`
  puis après réinitalisation,
- M appels aléatoires aux fonctions `TrouveCompresse` et `FusionRang`.

Les valeurs des entiers manipulés seront obtenues à l'aide des fonctions void srandom(unsigned int seed); et `long int random()`. On pourra tracer des graphiques à l'aide de `gnuplot`