

Allocation, chargement et libération des graphes

Avant d'être utilisé, un graphe doit être alloué par la fonction `grapheAlloue`. [Plus de détails...](#)

Fonctions

| | | |
|----------------|---|--|
| tGraphe | grapheAlloue () | Initialisation d'un graphe. |
| void | grapheLibere (tGraphe graphe) | Libère la mémoire occupée par un graphe. |
| void | grapheChangeType (tGraphe graphe, int oriente) | Définit si un graphe est orienté ou pas. |
| void | grapheAleatoire (tGraphe graphe, int nbSommets, int estOriente, double probaArc) | Crée un graphe aléatoire. |
| int | grapheChargeFichier (tGraphe graphe, char *fichier) | Charge un graphe depuis un fichier. |

Description détaillée

Avant d'être utilisé, un graphe doit être alloué par la fonction `grapheAlloue`.

Une fois utilisé, il est libéré par la fonction `grapheLibere`. Dans la majorité des cas, vous chargerez un graphe avec la fonction `grapheChargeFichier`.

Exemple typique d'utilisation:

```
tGraphe graphe;  
graphe = grapheAlloue();  
  
grapheChargeFichier(graphe, "fichier.grp");  
  
... code ...  
  
grapheLibere(graphe);
```

Documentation des fonctions

tGraphe **grapheAlloue** ()

Initialisation d'un graphe.

Exemple d'utilisation :

```
tGraphe graphe;  
graphe = grapheAlloue();
```

void **grapheLibere** (tGraphe **graphe**)

Libère la mémoire occupée par un graphe.

Paramètres:

graphe : un graphe précédemment obtenu par la fonction **grapheAlloue()**

```
tGraphe graphe;  
graphe = grapheAlloue();  
  
... code ...  
  
grapheLibere(graphe);
```

```
void grapheChangeType ( tGraphe graphe,  
                        int      oriente  
                        )
```

Définit si un graphe est orienté ou pas.

Paramètres:

graphe : un graphe (obtenu par grapheAlloue)

oriente : 0 (=non-orienté) ou 1 (=orienté) Le graphe doit être vide, donc fraîchement obtenu par grapheAlloue

```
void grapheAleatoire ( tGraphe graphe,  
                      int      nbSommets,  
                      int      estOriente,  
                      double   probaArc  
                      )
```

Crée un graphe aléatoire.

Paramètres:

graphe : un graphe (obtenu par grapheAlloue)

nbSommets : le nombre de sommets estOriente : 1 si on veut un graphe orienté, 0 sinon

probaArc : la probabilité qu'il y ait un arc entre deux sommets quelconques

```
int grapheChargeFichier ( tGraphe graphe,  
                          char *   fichier  
                          )
```

Charge un graphe depuis un fichier.

Paramètres:

graphe : un graphe (obtenu par grapheAlloue)

fichier : un nom de fichier contenant un graphe

Exemple d'utilisation :

```
tGraphe graphe;  
graphe = grapheAlloue();
```

```
grapheChargeFichier(graphe, "fichier.grp");
```

```
... code ...
```

```
grapheLibere(graphe);
```

Généré le Fri Nov 26 15:56:16 2010 pour Bibliothèque de manipulation les graphes par



1.5.8