

ALGORITHMES ET STRUCTURE DE DONNÉES

CONTENEURS

09/11/2022

Conteneur: Objet abstrait dont la principale caractéristique est de stocker des objets

On distingue les conteneurs ...

- qui mémorisent l'ordre d'arrivée des objets
ex : pile et files ;
- qui mémorisent des valeurs (ensembles, "sac", "map" (ou dictionnaires))

Conteneur de valeurs (**ensembles**) : ajouter de nouvelles valeurs, en retirer...

Exemples :

```
int ajouterEn(ensemble Entier_t* e, int v);  
// renvoie 0 si l'objet était déjà présent, 1 sinon  
// ajoute un entier à l'ensemble
```

```
int retirerEn(ensemble Entier_t* e, int v);  
// retire la valeur v à l'ensemble  
// renvoie 1 si l'entier était présent, sinon 0
```

```
int estDansEn(ensemble Entier_t* e, int v);  
// renvoie 1 si v est présent dans l'ensemble, 0 sinon
```

Et éventuellement des fonctions...

```
void initialiserEn(ensembleEntier_t *a);  
void détruire(ensembleEntier_t *a);
```

Utilisation :

```
ensembleEntier_t ens;  
initialiserEn(&ens); // l'ensemble est vide { }  
  
ajouterEn(&ens, 3); // {3}  
ajouterEn(&ens, 5); // {3, 5}  
ajouterEn(&ens, 3); // {3, 5}  
  
estDans(&ens, 3); // renvoie 1. l'ensemble est toujours {3, 5}  
estDans(&ens, 1); // renvoie 0.  
  
retirerEn(&ens, 3); // {5}  
retirerEn(&ens, 3); // {5}  
  
estDans(&ens, 3); // renvoie 0  
  
destruireEns(&ens);
```

Sac : Comme un ensemble mais peut contenir plusieurs fois la même valeur

Exemple :

```
ajouterSac(&sac, 5) // {5}  
ajouterSac(&sac, 3) // {5, 3}  
ajouterSac(&sac, 5) // {5, 5, 3}  
retirerSac(&sac, 5) // {5, 3}  
estDansSac(&sac, 5) // renvoie 1
```

Map (ou dictionnaire) : Comme un ensemble, cependant chaque valeur est associée à un objet

Exemple :

```
// la map est vide

ajouterMap(&map, 3, "Bonjour");
// la map contient {3 -> "Bonjour"}

ajouterMap(&map, 4, "Au revoir");
// la map contient {4 -> "Au revoir"};

ajouterMap(&map, 3, "Salut");
// la map contient {3 -> "Salut"};

valeurMap(&map, 4);
// renvoie l'adresse de la chaîne qui contient "Au revoir"
```

Remarques :

- On essaye d'implanter ces conteneurs et les opérations associées de manière efficace : bonne complexité en temps et en espace.
Exemple : arbre binaire de recherche, table de hachage
- Les interfaces de ces conteneurs peuvent être plus fournies
 - estVide()
 - Vider()
 - retirerauhasard() / pop()

Conteneur avec ordre : **Piles et files**

Une pile ou une file est un conteneur qui mémorise l'ordre dans lequel les objets ont été ajoutés.

Pile = First in, Last Out ; Last in, First Out

File = First in, First Out ; Last in, Last Out

Exemple d'une pile :

```
Pile p;  
int a;  
  
initialiser PI(&p);  
  
empilerI(&p, 1); // {1}  
empilerI(&p, 2); // {2, 1}  
  
a = depiler(&p); // {1}  
a = depiler(&p); // {}  
  
détruire(&p);
```

```
File f;  
int a;  
  
initialiser FI(&f);  
  
enfilerFI(&f, 1); // {1}  
enfilerFI(&f, 2); // {1, 2}  
  
a = defiler(&f); // {2}  
a = defiler(&f); // {}  
  
détruire(&f);
```