

DUT MMI – IUT de Marne-la-Vallée

02/04/2019

M2202 – Algorithmique et programmation Javascript

Cours 5

Piles et files

Résumé de l'épisode précédent (cours 2)

- Tris

Pour ranger un ensemble d'éléments tous comparables deux à deux selon un certain ordre.

Cas de base : tri d'un **tableau d'entiers** pour l'ordre \leq

Plusieurs algorithmes possibles :

- tri à bulles (cours + TD)
- tri par sélection (cours)
- tri par insertion (TD)
- ...

*plus ou moins
rapide... compter
le nombre de
comparaisons,
d'échanges, etc.*

Sources

- Cours de Jean-François Berdjugin à l'IUT de Grenoble
<http://berdjugin.com/enseignements/inf/inf220/>
- Cours de Xavier Heurtebise à l'IUT de Provence
<http://x.heurtebise.free.fr>

Plan du cours 5 – Structures de données

- Intérêt des piles et files
- Définition
- Manipulation en Javascript

Intérêt des piles et des files

Le problème des tableaux :

- insertion “difficile” d'un élément
- suppression “difficile” d'un élément

L'avantage des tableaux :

- accès facile au i -ième élément : `tab[i-1]`

Intérêt des files et des piles

L'intérêt des files et des piles :

- insertion facile d'un élément (au début pour une pile, à la fin pour une file)
- suppression facile de l'élément élément au début

Mais :

- accès "difficile" au i -ième élément

Files et piles

Files

FIFO “First In First Out”

File d'attente à la Poste

Piles

LIFO “Last In First Out”

Pile de documents à travailler

Files et piles

Files

FIFO “First In First Out”

File d'attente à la Poste



Gestion des buffers



<http://www.dessinateur.biz/blog/tag/file-dattente/>
<http://img.clubic.com/02989454-photo-cisco-crs3.jpg>

Piles

LIFO “Last In First Out”

Pile de documents à travailler



<http://fnhostile420.tumblr.com/post/1586779953/aporianonymous-i-have-a-to-do-pile>
<http://imageshack.us/photo/my-images/268/magicthegatheringback.jpg/>

Files et piles

Files

FIFO "First In First Out"

File d'attente à la Poste

Quatre fonctions :

- **créer** : créer une file vide
- **enfiler** : ajouter un élément en fin de file
- **défiler** : lire la valeur de l'élément au début de la file en l'enlevant de la file
- **tester si vide** : savoir si la file est vide



Gestion des buffers



Piles

LIFO "Last In First Out"

Pile de documents à travailler

Quatre fonctions :

- **créer** : créer une pile vide
- **empiler** : ajouter un élément en haut de la pile
- **dépiler** : lire la valeur de l'élément en haut de la pile en l'enlevant de la pile
- **tester si vide** : savoir si la pile est vide



Manipulation en Javascript

En Javascript les tableaux peuvent être utilisés comme des piles ou des files !

Files

- **créer** : créer une file vide

```
var file = [];
```

- **enfiler** : ajouter un élément en fin de file (à la *fin* du tableau *file*)

```
file.push(2);
```

```
file.push(5);
```

- **défiler** : lire la valeur de l'élément au début de la file en l'enlevant de la file (du début du tableau *file*)

```
var debut = file.shift();
```

- **tester si vide** : savoir si la file est vide

```
if(file.length==0){...}
```

Piles

- **créer** : créer une pile vide

```
var pile = [];
```

- **empiler** : ajouter un élément en haut de la pile (à la *fin* du tableau *pile*)

```
pile.push(3);
```

```
pile.push(4);
```

- **dépiler** : lire la valeur de l'élément en haut de la pile en l'enlevant de la pile (de la *fin* du tableau *pile*)

```
var haut = pile.pop();
```

- **tester si vide** : savoir si la pile est vide

```
if(pile.length==0){...}
```