

Programmation : TP noté (11 mai 2004)

[C] Arbre binaire de recherche

Un arbre binaire de recherche est un arbre binaire étiqueté dont les nœuds «à gauche» sont plus petits que les nœuds «à droite» : pour un nœud étiqueté α avec un sous-arbre gauche γ et un sous-arbre droit δ , tous les nœuds de γ sont plus petits que α et tous les nœuds de δ sont plus grands que α .

1. Définir le type des arbres binaires de recherche. Chaque nœud sera étiqueté avec une clé de type `int` et une valeur de type `(void*)`.
2. Écrire une fonction d'affichage des clés d'un arbre binaire de recherche dans l'ordre croissant (en profondeur d'abord de gauche à droite).
3. Écrire une fonction d'insertion d'une valeur avec sa clé associée :
`void insertion(bin *tree, int key, void* val)`
4. Écrire une fonction de recherche d'une valeur à partir de sa clé :
`void *recherche(bin tree, int key)`
5. Tester avec la fonction `main()` suivante :

```
int main(int argc, char *argv[]) {
    int i;
    bin t = NULL;
    for(i = 1; i < argc; i++) {
        insertion(&t, atoi(argv[i]), (void*)argv[i]);
    }
    afficher(t);
}
```

[OCaml] Enveloppe convexe

L'algorithme de JARVIS calcule l'enveloppe convexe d'un ensemble de points du plan en partant du point le plus bas et en ajoutant les points un à un dans le sens trigonométrique.

- Soit P_1 le point d'ordonnée minimale; il fait partie de l'enveloppe convexe.
- Soient P_{k-1} et P_k les deux derniers points ajoutés à l'enveloppe convexe. Le point suivant P_{k+1} est choisi pour maximiser l'angle $\widehat{P_k P_{k+1}, P_k P_{k-1}}$.
- Pour débiter, on ajoute un point fictif P_0 de coordonnées $(x_1 - 1, y_1)$ où (x_1, y_1) sont les coordonnées de P_1 .
- On s'arrête quand on retombe sur P_1 .

1. Écrire une fonction qui calcule la liste des points de l'enveloppe convexe d'un ensemble de points :

```
val jarvis : (int * int) list -> (int * int) list
```

Indications :

- Exécuter l'algorithme sur un exemple sur un papier (avec 5 points).
 - Manipuler une liste de points. On pourra commencer par écrire la fonction `extremum` qui renvoie un couple formé du plus petit élément d'une liste (pour un critère donné) et du reste de la liste.
 - On utilisera le produit scalaire pour calculer l'angle de deux vecteurs ($\vec{a} \cdot \vec{b} = \cos(\widehat{\vec{a} \vec{b}}) \|\vec{a}\| \|\vec{b}\|$)
2. Afficher l'enveloppe avec la fenêtre graphique.
 3. Quelle est la complexité de l'algorithme de JARVIS dans le cas le pire ?

