1 Ensemble ordonné

1.1 Implémentation

Nous avons choisi de réprésenter la structure de ensemble ordonné par une table. Cette table contient un tableau, qui stocke les élements triés par ordre croissant. La structure contient aussi la capacité du tableau ainsi que le nombre d'élements dans le tableau. Il est important de remarquer que la capacité est le nombre maximum d'éléments qu'il peut stocker avant de faire une réallocation de mémoire, s'il n'y a plus assez de place dans le tableau.

```
typedef struct s_set
{
    int * elements;
    int max_elt;
    int n_elt;

} * OrderedSet;
```

1.2 Fonctionnement de la fonction intersection

1.3 Complexité

1.3.1 Insertion dans l'ensemble

1.3.2 Appartenance d'un élément

1.3.3 Intersection de 2 ensembles

2 Arbre binaire de recherche

2.1 Choix d'implémentations pour l'arbre binaire de recherche

2.2 Fonctionnement des fonctions getAverageDepth et isBalanced

2.3 Complexité de FindCooccurrences

2.3.1 Hauteur moyenne d'un noeud

2.3.2 Complexité dans le pire cas dans un arbre non équilibré

2.3.3 Complexité dans le pire cas dans un arbre équilibré