

Algorithmique et Programmation

TD n° 3 : Arbres

École normale supérieure – Département d’informatique
algoL3@di.ens.fr

2014-2015

- Devoir à faire en binôme.
- Règles de collaboration : vous ne devez collaborer qu’avec votre binôme. Vous ne devez pas consulter internet ou d’autres sources pour essayer de trouver des idées de solution. Toutes vos questions doivent être adressées à vos enseignants.
- Devoir à rendre en cours le lundi 13 octobre à 15h30

Exercice 1. ARBRE AUGMENTÉ

Nous cherchons une structure de donnée qui nous permet les opérations suivantes. Cette structure garde en mémoire un ensemble d’intervalles qui est au départ vide.

- Insérer d’un intervalle $[g, d]$.
- Effacer un intervalle $[g, d]$.
- Trouver un intervalle contenant un entier p .

Cette dernière opération retourne une erreur si p est dans aucun intervalle. Si p est contenu dans plusieurs intervalles, n’importe quel intervalle contenant p peut être retournée.

Nous cherchons une structure de donnée qui permet de faire ces trois opérations en temps $O(\log(n))$ où n est le nombre d’intervalles présents au moment de l’opération.

Cette structure peut être obtenue à partir d’un arbre binaire de recherche en se rappelant de plus d’information à chaque noeud (et en mettant à jours ces informations à chaque opération). On dit alors que c’est un arbre binaire de recherche augmenté.

(Supposer que les opération de comparaison et arithmétique sur entiers délimitant les intervalles se font en temps constant. Ou en d’autre terme, on permet un facteur supplémentaire de \log du plus grand entier reçu à l’entrée.)

1. Trouver un arbre rouge-noir augmenté (sur les intervalles) pour ce problème et montrer que nous pouvons trouver un intervalle contenant p en temps $O(\log n)$.
Indice : utiliser le point gauche des intervalles comme clé de l’arbre.
2. Montrer que les opérations d’un arbre rouge-noir peuvent toujours être complétés dans le même ordre de temps si on met à jours les champs supplémentaires.