

Comptes-rendus de l'avancée du projet

Table des matières

0.1 Réunion du 22 février

Kevin a commencé par faire un résumé de l'avancement du projet. Plusieurs questions ont été soulevées concernant les bibliothèques disponibles pour générer des arbres en Latex. En effet, il existe des packages dédiés ([cliquer ici](#)) mais nous avons l'intention de créer une fonction en Python qui prend en argument un arbre et qui retourne directement le code d'affichage en Latex. Serge s'occupe de cette mission.

Nous avons également rencontré un problème concernant l'implémentation de la fonction suppression pour modifier une valeur en paramètre après l'appel de la fonction mais cela fonctionne sur des arbres générés aléatoirement grâce au module "random".

Kevin a travaillé sur les courbes de complexité en C qu'il devra essayer de faire en Python dans une idée d'homogénéité du projet bien que le langage de programmation ne soit pas imposé.

Concernant également les tests de complexité, vous nous avez conseillé d'établir d'autres statistiques comme la moyenne et l'écart-type en plus de la médiane. Aussi, la courbe doit être faite sur un même ordinateur, avec un même arbre et nous devons augmenter le nombre de tests car seulement une vingtaine peut biaiser les résultats. Pour cela, vous nous avez conseillé de répéter l'action 100 ou 1000 fois pour éviter la mesure de bruit, puis de diviser le temps par 100 ou 1000 ; en faisant attention aux fonctions suppression et insertion évidemment.

Vous nous avez également invité à utiliser pgfplot pour afficher les courbes. Il se trouve également que cela permet de tracer des arbres ([cliquer ici](#)). Nous avons ensuite évoqué des difficultés à utiliser Git : à propos de la gestion des conflits et la suppression de fichiers. Vous nous avez donc parlé des commandes "git rm", "gitignore" et "git mv". Vous nous avez aussi mis en garde sur les PDF à ne pas versionner car Git aura tendance à écraser entièrement la version antérieure au lieu de la modifier en ne sauvegardant que ce qui diffère de la version suivante.

Finalement, pour la prochaine fois, nous allons avancer sur la complexité et commencer à voir les quadrees.

0.2 Réunion du 8 mars

Lors de cette séance, Serge a présenté l'avancement du projet depuis la dernière réunion. En voici un bref résumé.

Charlotte a rédigé les définitions principales concernant les arbres binaires de recherche. On y retrouve notamment les notions déjà connues de nœud, feuille et hauteur ainsi que certaines nouvelles notions telles que le degré.

Serge nous a ensuite parlé des deux principaux types de parcours d'un ABR :

- Le parcours en largeur dans lequel on parcourt l'arbre par niveau
- Le parcours en profondeur dans lequel on parcourt tous les fils d'un nœud donné

La complexité de ces deux parcours est en $O(n)$ où n est le nombre de nœuds de l'arbre ce qui est cohérent car l'on doit passer une fois sur chaque nœud.

Nous avons ensuite vu le fonctionnement de la fonction de recherche, il s'agit d'une fonction récursive qui exploite la structure d'arbre binaire de recherche afin de savoir sur quel fils effectuer la récursion.

Quasiment toutes les fonctions principales (insertion, recherche...) utilisent ce principe, cependant, la fonction de suppression fonctionne un peu différemment car une fois un élément supprimé, il faut pouvoir "raccorder" son ou ses fils à l'arbre et il y a une distinction de cas à faire selon le nombre et la valeur des fils

Pour ces fonctions, le pire des cas est le cas où l'on explore l'arbre jusqu'à arriver à une feuille. Ainsi, ces fonctions sont linéaires en la hauteur de l'arbre dans le pire des cas.

J'ai rédigé quelques preuves sur des propriétés remarquables concernant les ABR, notamment le fait qu'un arbre parfait a une hauteur en $\theta(\log(n))$ où n est le nombre de nœuds de l'arbre.

Les résultats expérimentaux que j'ai obtenu sont donc cohérents car dans le cas d'arbres parfaits (ou presque), la complexité calculée expérimentalement des fonctions principales suit bien un modèle logarithmique.

Serge a aussi parlé de certains résultats intéressants bien que compliqués à démontrer, comme le fait qu'un ABR rempli aléatoirement aura en moyenne une hauteur logarithmique.

Nous avons aussi parlé des difficultés rencontrées dans la création d'arbres parfaits, les arbres que nous avons utilisés pour la complexité sont des arbres presque parfaits. C'est d'ailleurs pour cela qu'à la première réunion nos courbes de complexité étaient linéaires au lieu d'être logarithmiques. D'ailleurs, la création d'un arbre parfait de grande taille prend beaucoup plus de temps que les fonctions de recherche que nous souhaitons étudier. Ces dernières prennent un temps très négligeable par rapport à la durée nécessaire à la création d'un arbre parfait.

Thiziri a effectué une première implémentation du Quadtree. Nous avons découvert qu'il existait deux types de Quadtree : Le modèle Quadtree "point" qui correspond simplement à un ABR dont les nœuds possèdent 4 fils et le modèle Quadtree "région" que nous ne connaissions pas. C'est d'ailleurs ce modèle qui a été implémenté par Thiziri. Le Quadtree région n'est pas seulement un ABR à 4 fils, il faut plutôt voir cela comme un pavé du plan divisé en 4 parties de taille égales.

Pour la prochaine fois, nous essayerons de faire les régressions linéaires directement sur python au lieu de passer par LibreOffice. Il faudrait aussi essayer de faire des grands arbres sans recommencer de 0 à chaque fois car c'est ce que nous faisons actuellement. Cela impliquerait peut-être d'utiliser des étiquettes flottantes et non plus des nombres entiers.

Nous nous renseignerons aussi plus en détail sur le modèle du Quadtree "région".

0.3 Réunion du 22 mars

Cette semaine on a pas put faire le débriefing du projet.

Charlotte a commencé par faire une présentation de ce qu'on déjà fait, elle vous a signalé que ces deux dernières semaine on a pas trop avancé dans le projet à cause des partiel, on a juste finalisé le compte rendu de mi-parcours, et qu'on a commencé une partie des quadtree. Vous nous avez suggéré de vous envoyer le lien du overleaf de notre compte rendu, afin que vous nous corrigiez les erreurs qu'on fait en latex .

Serge et Kevin on exposé aussi leur idée sur une application des quadtree qui est l'application des quadtree sur les velibs ou les trottinettes afin de trouver leurs positions, mais c'est pas évident de trouver les données nécessaires pour la modéliser.

Enfin pour les deux semaines à venir, on fera une étude théorique sur les quadtree points et région, et essayer de trouver un exemple application de quadtree où les données sont accessibles.

0.4 Réunion du 03 avril

La réunion a commencé par un retour sur le rapport de mi-parcours. Nous devons mieux mettre en valeur les définitions (notamment en utilisant l'environnement LaTeX dédié), et ne pas être trop synthétique au détriment de la clarté. Nous avons vous aussi vu un certain nombre d'erreur courante en LaTeX et de bonnes pratiques à adopter. Les commentaires détaillés reçus nous serviront pour l'amélioration des parties déjà rédigées du rapport et la suite du rapport.

Thiziri a ensuite présenté l'avancement du projet. Elle a commencé par les AVL, arbres binaires de recherches équilibrés inventés par Georgii Adelson-Velsky et Evguenii Landis, que Serge a commencé à implémenter. Elle a décrit le fonctionnement des rotations en déroulant l'algorithme sur un exemple au tableau. Nous avons discuté des subtilités dans l'implémentation avec la gestion de la hauteur, que nous devons éviter de recalculer à chaque fois. La solution sera d'en faire un attribut de chaque nœud.

Puis elle a présenté les Quadtree Point, structure très proche des ABR, que Serge a commencé à décrire dans un document overleaf. Ils consistent en effet à avoir un élément par nœud avec quatre fils, un par quadrant. Elle a expliqué que le calcul de la hauteur et de la taille ainsi la recherche et l'insertion d'éléments fonctionnent comme pour les ABR. Nous avons ensuite discuté de la suppression, plus délicate que pour les ABR, avec la gestion des fils de l'élément à supprimer. Nous avons ensuite discuté des méthodes d'analyse spatiale comme la recherche des k plus proches voisins, celle des éléments dans une zone donnée (circulaire ou rectangulaire), pas encore implémentées.

Ensuite Thiziri a introduit les Quadtree Region, structure qu'elle a commencé à implémenter. Elle a expliqué que cette structure consiste à subdiviser l'espace en 4 quadrant récursivement. La structure peut être utilisée pour stocker de données "continues", elle a montré l'exemple avec un dessin au tableau.

Elle a ensuite montré un exemple d'application à la compression d'image, dans lequel on subdivise l'espace tant qu'il y a plusieurs couleurs (ici uniquement noir ou blanc), et on s'arrête lorsque le quadrant est de couleur homogène. Sur l'exemple on passe d'une image de 64 pixels à un arbre à 21 noeuds.

Elle a ensuite présenté l'utilisation de tels Quadtree pour représenter des données de type point, qui sont stockés dans les feuilles uniquement, soit un point par feuille soit par une liste de point dans chaque feuille avec une taille maximum fixée. Elle a donné deux résultats sur la taille et la hauteur de tels arbres, ainsi qu'une relation entre longueur d'un quadrant, longueur de la région représentée et profondeur.

Sur les méthodes nous avons vu que la suppression était plus aisée que pour le quadtree point. Il faut quand même faire des distinction de cas en fonction de la présence d'autres éléments dans les frères du point à supprimer, en effet si l'élément à supprimer est le seul fils, on doit supprimer son père.

Finalement elle a discuté des avantages et des inconvénients des deux types de quadtree. Elle a expliqué à l'aide d'exemples au tableau que la structure d'un quadtree région ne change pas en fonction de l'ordre d'insertion des éléments, alors qu'elle change pour un quadtree point.

Charlotte a commencé à travailler sur l'implémentation de quadtree point et Kevin sur une application au traitement d'image.

Après la présentation nous avons vu les bonnes pratiques pour les slides, notamment : éviter les phrases complètes et les textes longs, ne pas mettre de ponctuation dans les bullet points, ne pas faire de bullet point inutile, ne pas mélanger le type d'éléments listés.