

Recherche d'information

Objectifs de ce laboratoire

Dans ce laboratoire, nous allons utiliser les fichiers de description de graphes du tp2 pour faire un exercice de recherche dichotomique. L'objectif premier est de bien se familiariser avec le traitement des fichiers binaires du tp2, et avec la recherche dichotomique, pour que les exécutions du tp2 se passent bien.

Description de la tâche à réaliser

On vous fournit des fichiers binaires de description de graphes et la classe graphe de base suivante

```
class graphe{
private:
    struct noeud{ //description de toutes les composantes d'un noeud
    };
    map<uint32_t,noeud> lesNoeuds; //les noeuds deja lus
    uint32_t nbNOEUDS; //le nombre de noeuds
    ifstream DATA; //le flot d'entree
    uint32_t DEBUT; //debut de la partie fixe
    void lire_noeud(uint32_t);
    void lire(uint16_t&); //fonctions utilitaires de lecture binaire
    void lire(uint32_t&); //qui dependent de l'architecture
    void lire(float&);
    graphe(const graphe&)=delete; //copieur et
    graphe& operator=(const graphe&)=delete; //affectateur désactivés
public:
    graphe(string); //constructeur
    ~graphe(); //destructeur
    uint32_t size()const; //nombre de noeuds dans le graphe
    void afficher_noeud(uint32_t); //afficher toutes les infos sur un noeud
    uint32_t obtenir_noeud(string); //chercher un numero de noeud
    ...
};
```

C'est la fonction obtenir_noeud qui nous intéresse ici, elle n'était pas dans la description donnée à l'origine. Elle reçoit une chaîne de caractères et doit retourner comme résultat le numéro du noeud dont le nom est exactement cette chaîne, ou dont le nom vient immédiatement après cette chaîne. Un peu à l'image de la fonction `lower_bound` de C++98. Donc si le nom recherché vient après le dernier du graphe, cette fonction retourne un `uint32_t` donnant le nombre de noeuds, soit un de plus que le numéro du dernier.

Votre programme

Votre programme doit d'abord lire sur cin le nom du fichier qui contiendra le graphe, puis doit communiquer avec l'utilisateur pour obtenir sur cin une chaîne de caractères qui servira à la recherche. Suite à cette recherche (par un appel à obtenir_noeud, qui retourne un `uint32_t`), il doit afficher les informations concernant le noeud recherché.

J'utiliserai tel quel votre main, mais je lui fournirai mes propres données. Le programme devra lire sur cin le nom du fichier à ouvrir, puis des noms de noeuds à rechercher. Dans chaque cas, le programme doit simplement afficher le noeud résultant, ou indiquer que le numéro de noeud est égal au nombre de noeuds, donc n'existe pas.

Pour faire vos tests sous windows, utilisez les fichiers ibin que je vous ai déjà fournis. Pour les tests sous unix, je fournirai les fichiers en format `bid_endian`. Pour faire vos tests sur tarin, vous n'avez qu'à entrer le bon nom de fichier, en mettant par exemple ceci quand le programme demande un nom de fichier :

```
/home/public/cours/ift339/grUSA_32768.sbin
```

Remise du travail

Ce travail doit être complété à 23 h 59 jeudi le 6 novembre au plus tard. Soumettez-le à **turnin**, après vous être assurés qu'il fonctionne bien sur tarin. Ne soumettez pas un répertoire, et attention à la syntaxe! Une seule soumission par équipe! Soumettez votre programme principal dans un fichier `main.cpp`, et toutes les classes dans `graphe.h` et `graphe.cpp`.

```
turnin -c ift339 -p labo3 main.cpp graphe.h graphe.cpp
```