1ère Année ENSEIRB option Electronique

Algorithmique et structure de données

TD5 : Files et Tas

1 Recherche de plus courts chemins

On considère un réseau ferroviaire modélisé par une matrice `mat` de la façon suivante. Les n gares du réseau sont considérées comme des nombres entre 0 et $n-1$ et la valeur de `mat[i][j]` est égale à 1 s'il existe un train direct entre i et j ; elle est égale à 0 sinon.

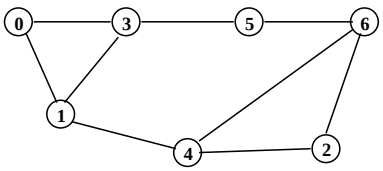

FIG. 1 – Un exemple de réseau

Par exemple le réseau dessiné ci-dessus est modélisé par la matrice :

$$\begin{array}{ccccccc} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}$$

On se propose de résoudre le problème du plus court chemin dans un tel réseau depuis une gare de départ. Il s'agit dans un premier temps de déterminer le nombre d'étapes nécessaires pour atteindre chacune des gares depuis cette gare de départ.

On supposera dans la suite que la gare de départ est 0 et on calculera le nombre d'étapes dans un tableau `dist`. Par exemple le tableau correspondant à l'exemple ci-dessus est :

```
0, 1, 3, 1, 2, 2, 3
```

1. Calculer le tableau `dist` sur l'exemple suivant de réseau :

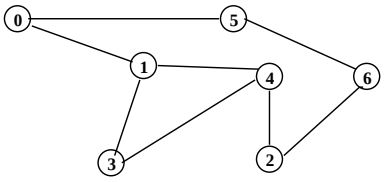

FIG. 2 – Deuxième exemple de réseau

2. Ecrire un algorithme

   `voisines (mat, n, i)`

   qui étant donné un réseau de `n` gares donné par un tableau de tableaux `mat` représentant la matrice décrivant le réseau et une gare particulière `i`, affiche toutes les gares atteignables depuis `i` en une seule étape.
   Sur le réseau de la Fig1 pour la gare `i = 3` l'affichage doit être : `0, 1, 5`.

3. On vous demande d'écrire l'algorithme qui calcule les nombres d'étapes depuis la gare 0, dans un tableau `dist`. Pour cela on utilisera une file F (de type FIFO) dans laquelle on mettra d'abord la gare 0, puis à chaque étape on determinera `dist[j]` pour les gares `j` voisines d'une gare `i` dont on connaît déjà la valeur de `dist`.
4. Afin de déterminer un chemin le plus court possible (en nombre d'étapes) entre la gare 0 et les autres gares on utilise un tableau `pred` tel que pour tout `i, pred[i]` est la gare qui précède `i` dans le chemin le plus court qui va de `0` à `i`. Par exemple, le tableau `pred` sur l'exemple de la Fig1 est le suivant (le -1 signifie qu'il n'y a pas de prédécesseur sur le chemin de 0 à 0.

   `-1, 0, 4, 0, 1, 3, 4`

   Comment modifier l'algorithme précédent pour qu'il cacule `pred` en même temps que `dist` ?
5. Donner l'algorithme qui affiche le plus court chemin pour atteindre `i`, une fois calculé `pred`.

2 Gestion des tas

On rappelle qu'un tas est modélisé par un tableau `t` tel que pour tout `i` on a

$$t[i] > t[\frac{i-1}{2}]$$

On a vu l'algorirthme d'ajout en cours. On s'interesse ici à un autre algorithme `rendreTas(t, n, j)` qui pourra être utilisé dans plusieurs cas, comme la suppression du plus petit élément par exemple. Il s'agit de transformer en un tas, par échanges de valeurs, un tableau `t` contenant `n+1` nombres qui satisfait la conditions des tas sauf peut être en un seul point `j` pour lequel on n'est pas assuré de la satisfaction des conditions :

`t[2j+1] > t[j]` et `t[2j+2] > t[j]`

.

1. Ecrire l'algorithme `rendreTas(t, n, j)` en considérant les trois cas
   – `2j+1 > n`

– `2j+1 = n`
– `2j+1 < n`
et en effectuant si nécessaire un appel récursif.

2. Comment utiliser `rendreTas` afin de supprimer le plus petit élment d'un tas
3. (Prolongement de l'exercice 1).
On s'interesse maintenant à des réseaux de type routiers avec des informations sur la distance en km entre deux villes. La matrice `mat` contient alors des entiers qui indiquent cette distance. Montrer comment la structure de tas peut être utilisée pour résoudre les plus courts chemin depuis une ville donnée. Vous vous aiderez de l'exemple suivant

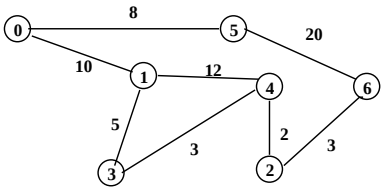

FIG. 3 – Un exemple de réseau routier