

IFT2015, Été 2023

TP3 : Planification d'un réseau de panneaux publicitaires électroniques

(ARM algorithme Prim Jarnik)

Deanna Wung

## Auto évaluation

Le programme fonctionne correctement. Il produit le bon ARM et l'ordre des sommets et arêtes affichés respecte la consigne.

Les fichiers de sortie correspondent aux exemples de sortie en terme de l'ARM produit : toutes les mêmes arêtes et sommets sont retenus, mais les sommets de départ et d'arrivée peuvent être inversés dans l'appelation, par exemple : "rue9 a b 5" au lieu de "rue9 b a 5". Ainsi l'ordre de l'affichage ne sera pas identique aux exemples de sortie car les sommets de "départs" sont définis différemment.

## Analyse de complexité en notation grand O

Complexité globale :  $O(E \log E + V \log V + V)$ , où

$V$  = nombre de sommets (vertices)

$E$  = nombre d'arêtes (edges)

parce que...

### $O(E \log E)$

Le programme parcourt chacune des  $E$  arêtes (au maximum 2 fois, car dans 2 sens), et pour chacune, peut l'ajouter à la priority queue, donc  $= O(E \log E)$ . On peut avoir un excès d'arêtes dans la priority queue, donc on utilise  $E$  et non pas  $V$ .

Par la suite, on vide la priority queue jusqu'à ce qu'on arrive sur une arête qui évite les cycles, donc  $= O(E \log E)$

### + $O(V \log V)$

Ensuite on parcourt chacune des  $V$  arêtes retenues, et pour chacune on l'ajoute dans une TreeMap ( $O(\log V)$ ), donc au total :  $O(V * \log V)$ . Ici on dit  $V$  parce que le nombre d'arêtes retenues  $= V-1$ .

### + $O(V)$

Enfin, on parcourt toutes nos données pour les afficher (tous les  $V$  sommets visités dans un TreeSet, toutes les  $V$  arêtes visitées dans un TreeSet) pour créer le fichier final. (Encore une fois, le nombre d'arêtes retenues serait  $V-1$ ). Donc  $= O(V)$ .

### = $O(E \log E + V \log V + V)$

Voir les commentaires dans le code du programme pour plus de détails.