

Programmation efficace – Licence 3 Informatique

Trouver des équipes !

Problème

Dans ce problème, il y a n joueurs et un arbitre. On cherche à répartir efficacement les n joueurs dans p équipes ($n \geq p$).

Les joueurs d'une même équipe doivent communiquer entre eux : chaque joueur de l'équipe doit envoyer un message à chaque autre joueur de son équipe (un message différent pour chaque destinataire). Pour garantir la confidentialité des messages, ils sont cryptés de manière assez basique : le joueur x dispose d'une clé de cryptage personnelle k_x et lorsqu'il veut communiquer avec le joueur y , il code son message avec k_x , l'envoie à l'arbitre (qui dispose de toutes les clés) qui décode le message et le recode selon la clé k_y . Le message va donc de x à l'arbitre puis de celui-ci à y . Le réseau de communication est un graphe orienté pondéré : les joueurs et l'arbitre sont placés sur certains noeuds (au plus une personne par noeud) et le coût d'une transmission entre deux points correspond au poids du chemin emprunté dans le réseau.

Etant donnés un réseau de communication et la position des n joueurs et de l'arbitre dans le réseau, le but est de répartir les joueurs dans les p équipes afin d'avoir un coût de communication globale minimal.

Formats

Une instance du problème est donnée dans un fichier. La première ligne contient 4 entiers séparés par des espaces : le nombre de sommets m dans le réseau ($2 \leq m \leq 5000$), le nombre n de joueurs ($1 \leq n < m$), p le nombre d'équipes à constituer ($n \geq p \geq 1$), et le nombre d'arcs a dans le graphe.

Puis les a lignes suivantes contiennent la description des arcs sous la forme de trois entiers : $x y d$ où x et y sont des sommets et d est le poids de l'arc (x, y) .

On suppose que les noeuds $1, \dots, n$ correspondent à la position des n joueurs, et que l'arbitre est sur le sommet $n + 1$.

Par exemple, le premier exemple ci-dessus sera décrit avec :

```
5 4 2 10
5 2 1
2 5 1
3 5 5
4 5 0
1 5 1
2 3 1
3 2 5
2 4 5
2 1 1
3 4 2
```

Et le résultat attendu pour cet exemple est 13.

Et pour l'exemple ci-dessous, on doit obtenir 24 :

```
5 4 2 10
5 2 1
2 5 1
3 5 5
4 5 10
1 5 1
2 3 1
3 2 5
2 4 5
```

2 1 1
3 4 2