



Concours national d'informatique
Épreuve écrite d'algorithmique
Paris I

28 Janvier 2017

MANY WHELPS ! NOW HANDLE IT !



Alright chums, let's do this.

1 Préambule

Bienvenue à **Prologin**. Ce sujet est l'épreuve écrite d'algorithmique et constitue la première des trois parties de votre épreuve régionale. Sa durée est de 3 heures. Par la suite, vous passerez un entretien (20 minutes) et une épreuve de programmation sur machine (4 heures).

Conseils

- Lisez bien tout le sujet avant de commencer.
- **Soignez la présentation** de votre copie.
- N'hésitez pas à poser des questions.
- Si vous avez fini en avance, relisez bien.
- N'oubliez pas de passer une bonne journée.

Remarques

- Le barème est donné à titre indicatif uniquement.
- Indiquez lisiblement vos nom et prénom, la ville où vous passez l'épreuve et la date en haut de votre copie.
- Tous les langages sont autorisés, veuillez néanmoins préciser celui que vous utilisez.
- Ce sont des humains qui lisent vos copies : laissez une marge, aérez votre code, ajoutez des commentaires (**seulement** lorsqu'ils sont nécessaires) et évitez au maximum les fautes d'orthographe.
- Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
- Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le-nous, nous pouvons ajouter des questions plus difficiles.

2 Sujet

Introduction

LONIORPG — Les Ornythorinques Nains Irréductibles Online Role Playing Game™ — est le jeu de rôle en ligne le plus révolutionnaire des vingt dernières années. Les plus de soixante-dix continents à explorer, cent onze classes, dix-sept mille métiers et trois races jouables, ainsi que la possibilité d'incarner un véritable ornithorynque nain ont participé à l'immense succès qu'a rencontré ce jeu dernière génération.

Fasciné par ce jeu¹, vous avez décidé d'aller encore plus loin en postulant chez AcTTYvision, la société qui l'édite. Vous avez été chargé d'une tâche de la plus haute importance : créer un outil de recherche de groupe.

Dans LONIORPG, certaines quêtes demandent d'avoir des amis avec qui partir à l'aventure, car les dangers à braver sont bien trop importants pour des joueurs seuls². Pour trouver des amis dans le jeu, il n'y a malheureusement pas beaucoup de solutions³. La plus simple est de faire partie d'une guilde, mais celles-ci ne prennent que des joueurs qui ont déjà de l'expérience, il est donc difficile de « rentrer dans le système ». L'autre solution est de demander sur les salons de discussion à disposition des joueurs, mais le dialogue qu'on y entend se résume souvent à ceci :

```
<hunter2> hpal lf dk/war 870+ push 2v 3v 2k+ cr req  
<dives> "wipe club" 7/7hm 3/7mm seek roster rdd/heal w/ exp 2 leg BiS  
<leroy> Bonjour ! Je cherche de valeureux aventuriers pour m'aider dans ma  
         quête de l'inestimable essence du pouvoir !  
<angel42> leroy: no curve achiev? lol naab go pex
```

Frustrés par l'hermétisme de cette communauté, les nouveaux joueurs réclament une meilleure façon de trouver des compagnons pour progresser dans leurs aventures. C'est là que vous rentrez en jeu⁴ !

Votre travail sera de concevoir un outil qui permet de créer automatiquement des groupes d'aventuriers, aventuriers qui s'inscriraient ainsi dans une file d'attente pour être ensuite placés dans des groupes. Cependant, certains aventuriers ont déjà des amis⁵ avec eux lorsqu'ils s'inscrivent dans la file d'attente ! Il faut donc laisser la possibilité d'inscrire plusieurs amis ensemble sans qu'ils puissent être séparés dans des groupes différents : ces groupes d'amis sont appelés des **cliques**.

Partie I : Donjon !

Les donjons nécessitent des groupes de 5 personnes pour être complétés. Le principe est donc de créer des groupes complets de 5 personnes à partir des cliques qui s'inscrivent.

Pour l'instant, on part du principe que l'on a en avance la liste de toutes les cliques qui se sont inscrites. On propose l'algorithme suivant :

- On crée la liste de groupes, initialement vide.
- On prend la première clique et on l'ajoute dans le premier groupe de la liste qui peut la contenir.
- Si aucun groupe ne peut la contenir, on crée un groupe supplémentaire à la fin.
- On recommence avec la clique suivante jusqu'à avoir mis toutes les cliques dans la liste.

1. Et légèrement accro.

2. Si vous êtes un *PGM*, vous pouvez peut-être arriver à en *solo* quelques unes.

3. Pour la plupart des joueurs, c'est quand même plus facile qu'IRL.

4. Non, pas ce jeu là, un autre.

5. Quels privilégiés, ceux-là.

Exemple :

Numéro de la clique	1	2	3	4	5	6	7
Nombre de joueurs	3	4	2	1	1	1	3

Ici, 7 cliques se sont inscrites. On commence par créer un groupe qui contient les trois joueurs de la première clique. On ne peut pas mettre la deuxième clique dans ce groupe car on dépasserait la limite de 5 joueurs, on la met donc dans un deuxième groupe. La troisième clique, elle, ne contient que deux joueurs, on peut donc la mettre dans le premier groupe. En continuant l'algorithme, on obtient les groupes suivants :

Groupe	1	2	3
Cliques	1, 3	2, 4	5, 6, 7
Nombre de joueurs	3 + 2	4 + 1	1 + 1 + 3

Question 1

(2 points)

Cet algorithme est en fait incorrect : dans certains cas il n'arrivera pas à former des groupes complets alors qu'il y a assez de joueurs.

Donner une liste de cliques où cet algorithme ne donne pas la meilleure solution. Changer ensuite l'ordre des cliques pour que l'algorithme fonctionne correctement.

Question 2

(3 points)

Proposer un algorithme qui crée le nombre maximum de groupes de 5 personnes à partir de la liste de cliques inscrites, c'est à dire qui laisse le moins de joueurs possibles sans aucun groupe.

Question 3

(2 points)

Dans les conditions réelles du jeu, on n'a jamais en avance la liste des cliques inscrites. En effet, les cliques s'inscrivent une par une et on doit former des groupes « à la volée » le plus tôt possible après avoir formé un groupe complet. À présent, on se place donc dans le cas (plus utile) où l'on n'a pas en avance la liste des groupes, et on les reçoit un par un.

Pourquoi l'algorithme de la question précédente ne peut plus fonctionner ? Expliquer pourquoi, comme l'on n'a pas la liste des groupes en avance, on risque d'être parfois obligés de constituer des groupes de façon non-optimale.

Question 4

(2 points)

Bien que nous ne puissions plus constituer systématiquement des groupes de façon optimale, nous n'en sommes pas pour autant réduits à utiliser l'algorithme naïf de la question 1. En effet, imaginons que nous recevions des cliques contenant les nombres de joueurs suivants (dans l'ordre d'apparition dans la file) : **3, 1, 4**. L'algorithme de la question 1 va créer deux groupes de 4 joueurs, alors qu'il est possible de constituer un groupe de 5 joueurs et ainsi les faire sortir de la file d'attente. Tant que l'on n'a pas fait partir des groupes, on peut encore les réordonner.

Proposer un algorithme qui permet de constituer dès que possible un nombre maximum de groupes complets. Quel lien peut-on faire avec l'algorithme de la question 2 ?

Partie II : Classes et rôles

He needs those devout shoulders ? Doesn't... isn't he a paladin ?

Votre outil de constitution de groupes de donjons fonctionne à présent plutôt bien⁶, mais il a un défaut. Dans LONIORPG™, chaque classe peut assumer un ou plusieurs rôles parmi les trois suivants : **Tank**, **Soins** et **Dégats**. Un groupe équilibré doit être constitué d'un Tank, d'un Soins et de trois Dégats.



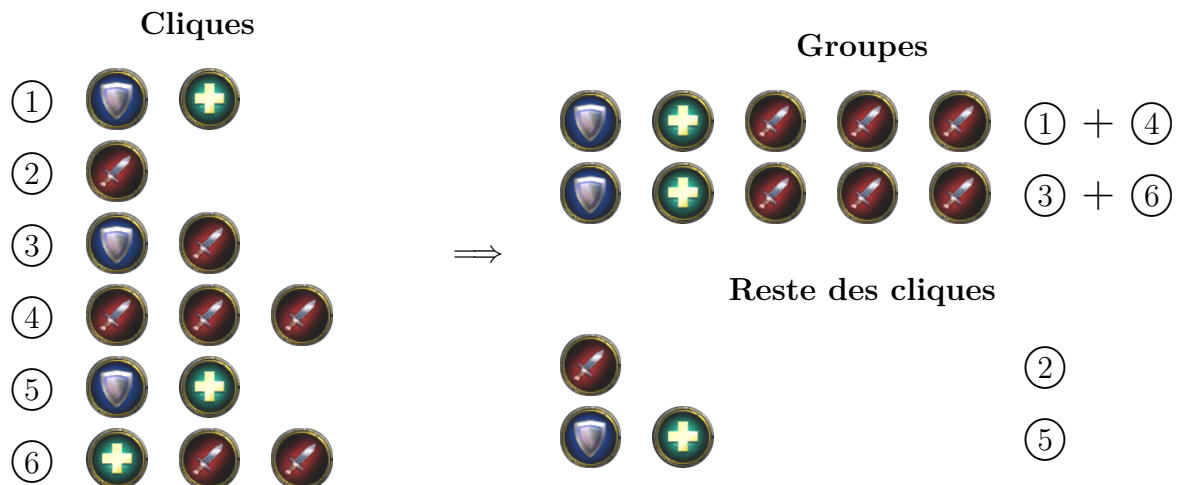
Question 5

(2 points)

Dans les cliques qui s'inscrivent, chaque joueur indique à présent le rôle qu'il souhaite accomplir. Proposer un algorithme qui constitue le plus de groupes équilibrés complets possible.

Note : On considèrera qu'il est impossible d'inscrire une clique comportant trop de joueurs pour un certain rôle, par exemple on ne pourra pas trouver dans la liste des cliques une clique avec deux Tanks.

Exemple :



Question 6

(2 points)

Certaines classes peuvent accomplir plusieurs rôles à la fois, ce sont des classes dites **polyvalentes**. Chaque joueur peut s'inscrire non plus avec un rôle mais avec une liste de rôles qu'il souhaite accomplir.

Comment adapter l'algorithme de la question précédente pour prendre en compte les classes polyvalentes ?

6. Enfin, j'espère pour vous, ce sera au correcteur de votre copie d'en décider.

Question 7

(3 points)

Intéressons-nous maintenant à l'ordre dans lequel les cliques sont assignés à des groupes. On remarque qu'il est impossible de faire passer toutes les cliques dans l'ordre de leur inscription, sans quoi la file pourrait se bloquer. Deux cliques de même taille et de même composition passent certes toujours dans leur ordre d'inscription, mais dans les autres cas :

- Une clique comportant moins de joueurs doit nécessairement pouvoir passer avant (sinon une clique de 4 joueurs suivie d'une clique de 3 joueurs bloqueraient toute la file, par exemple).
- Une clique comportant autant de joueurs doit nécessairement pouvoir passer avant si elle contient un arrangement de rôles différents (sinon une clique de 3 Dégats et un Soins suivie d'une clique d'un Dégat bloqueraient toute la file, par exemple).

On dit qu'une clique est **jalouse** si une autre clique passe avant elle sans remplir ces conditions (c'est à dire une clique qui s'est inscrite après elle et qui comporte plus de joueurs qu'elle).

Réécrire l'algorithme de constitution des groupes en faisant en sorte qu'il n'y ait jamais de cliques jalouses.

Question 8

(1 point)

Donner un exemple de liste de cliques où, en appliquant l'algorithme « avec jaloux », une clique « jalouse » passe plus tôt que si l'on applique l'algorithme « sans jaloux ». Qu'en déduire sur la pertinence de ce changement d'algorithme ?

Partie III : Raid !

Go away from the head Lee. Go. Away.

Les donjons, c'est marrant, mais hors de question de s'attaquer à Raxxor, le dieu maléfique et corrompu des ratons géants, avec un petit groupe de 5 personnes. Un **raid** est un groupe de plus de 5 personnes.

Question 9

(2 points)

Donner une liste de cliques et une taille de raid pour lesquels l'algorithme de la partie I (adapté pour cette taille) ne constitue plus des groupes de raid de façon optimale.

Question 10

(3 points)

Montrer que l'algorithme de la partie I fonctionne pour des tailles de groupes inférieures à 7 et ne fonctionne plus pour des tailles de groupes supérieures ou égales à 7.

Bonus : quelle est la *classe de complexité* de ce problème pour des tailles de groupes supérieures ou égales à 7 ?

Les questions suivantes sont des questions ouvertes et n'ont pas de réponse spécifique attendue, vous êtes libres de les traiter en les orientant comme bon vous semble. La précision et la pertinence de l'argumentation et des solutions proposées seront appréciées.

Question 11

(3 points)

On propose les deux algorithmes suivants pour constituer des groupes de raid de taille $N \geq 7$:

- “**first fit**” : Chaque fois qu’une nouvelle clique s’inscrit, on la met dans le premier groupe dans lequel elle rentre.
- “**best fit**” : Chaque fois qu’une nouvelle clique s’inscrit, on la met dans le groupe où il reste le moins de place parmi ceux dans lesquels elle rentre.

Discuter des avantages et des inconvénients de chacun de ces deux algorithmes. Lequel voudriez-vous utiliser pour faire l’outil de recherche de groupe, et pourquoi ?

Question 12

(3 points)

Certains raids peuvent-être **flexibles**, c’est à dire qu’ils acceptent autant de joueurs qu’ils le souhaitent (à partir de 10 joueurs), à condition que leurs compositions soient équilibrées. Pour qu’un groupe de raid soit équilibré, on a besoin de toujours avoir⁷ :

$$\text{🛡️} = 2 \quad \text{⚡} = \left\lfloor \frac{2 \times \text{🔪}}{7} \right\rfloor$$

Comment pourrait-on utiliser la flexibilité des raids pour faire partir des groupes équilibrés plus vite ? On pensera à prendre en considération la rareté de certains rôles (faire partir toujours des groupes de taille minimale peut par exemple être problématique s’il y a un faible nombre de Tanks qui rentrent dans la file d’attente, car leur nombre est fixe dans un groupe équilibré).

Vous ne pouvez attaquer les questions suivantes que si vous avez bu votre flasque d’intelligence, mangé votre pain au chocolat du destin, utilisé votre rune d’agilité et qu’il vous reste assez de mana.

Partie IV : Bonus

Question bonus 13

(3 points)

Proposer une façon d’estimer le plus précisément possible le temps d’attente estimé pour chaque clique avant que son groupe ne soit complet.

Question bonus 14

(1 point)

Décrivez dans un style *role-play* les caractéristiques de votre personnage et les tenants et aboutissants de ses quêtes dans LONIORPG (si vous en aviez un).

Le sujet comporte 7 pages (sans compter la page de garde) et 14 questions, parmi lesquelles 2 questions bonus. Les questions normales sont notées sur 28 points, et les questions bonus rapportent au total 4 points, plus 1 point de présentation.

7. $\lfloor x \rfloor$ signifie x arrondi à l’entier inférieur (ou « tronqué »).