

Devoir 3

Structure de Données

Guillaume Poirier-Morency p1053380
Vincent Antaki p1038646

23 novembre 2014

0.a Résumé

Implémentation d'une file à priorités multiples et tests relatifs.

1

2

La classe Queue

Notre implémentation commence avec la classe Queue qui implémente une structure de liste chaînée simple et les fonctions enqueue() dequeue() et remove(). Les fonctions built-in python (_len_, _iter_, _bool_, _next_, _contains_, ...) ont été utilisées pour améliorer la rapidité d'exécution de notre programme et ainsi qu'en faciliter l'implémentation.

CasinoQueue

Les instances de la classe CasinoQueue sont des files à 3 priorités. Elles possèdent les fonctions de base des files : enqueue() et dequeue(). Elle possède 3 files chaînées de la classe Queue correspondant à ses trois priorités (table brisée, changement de table et le reste). Les éléments mis dans les files sont des tuplets de joueur, temps d'entrée dans la file et, dans le cas de la file pour changement de table, la table désirée.

Lorsque appelée, la fonction dequeue retire, en tout respect de l'énoncé, une personne de la file ou retourne une exception si cela est impossible. dequeue() possède un ordre constant lorsque il y a des éléments dans la queue pour table brisée et a un ordre linéaire par rapport au nombre de joueur dans la queue pour changement de table dans tous les autres cas.

Chaque appel de la fonction enqueue vérifie que le nom entrée n'est pas un doublon d'un nom existant déjà dans le casino. Si ce n'est pas le cas, les paramètres table et broken détermineront à quelle file sera enqueue le joueur.

Il nous aurait été possible d'implémenter la vérification des doublons par une itération à travers les 3 queues. Pour cause de mauvaise complexité, nous avons refusé cette option. Il nous aurait été possible de faire un arbre qui stocke les noms de tous les joueurs qui sont dans le casino et qui font des recherches en $O(\log n)$. Pour cause de flemmardise, nous avons refusé cette option. Nous avons implémenté _contains_ qui teste l'appartenance à un objet à self.players (un set !) lors de l'entrée d'un nouveau joueur dans le casino (ajout d'un joueur à la normal_queue).