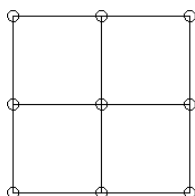


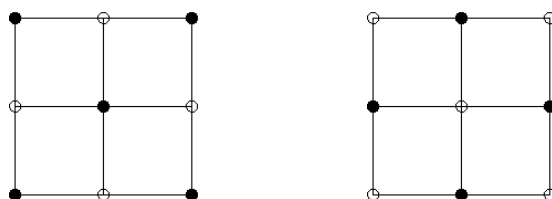
Série 6

1 Organiser la surveillance d'un musée

Vous devez organiser la surveillance d'un grand musée comme le Louvre. Un tel musée possède un grand nombre de corridors connectés les uns aux autres, qu'on peut modéliser schématiquement par un graphe:



Votre but est de placer un nombre minimum de gardiens aux intersections (=sommets) de ce graphe pour surveiller tous les corridors (=arêtes) du musée. On suppose qu'un gardien placé à une intersection donnée peut avoir un oeil sur tous les corridors directement reliés à cette intersection. Deux solutions possibles pour le graphe ci-dessus sont:



mais seule la solution de droite est optimale, car elle utilise un nombre minimum d'intersections.

Une telle solution optimale peut bien sûr être obtenue en essayant tous les placements possibles: le nombre de possibilités à tester est alors de 2^n si n est le nombre d'intersections (car en chaque intersection, on peut décider de placer un gardien ou pas). Mais personne ne sait actuellement s'il existe un algorithme qui permette de trouver une solution optimale en temps polynomial en n , et ça n'est donc pas ce qui vous est demandé dans cet exercice!

En revanche, il est possible de trouver une solution sous-optimale mais relativement proche de la solution optimale en temps polynomial en n . C'est ce qui vous est demandé ici. Plus précisément: supposons que pour un graphe donné (quelconque), la solution optimale nécessite le placement de k gardiens. Proposez alors un algorithme

- dont le temps d'exécution soit polynomial en n ;
- dont la sortie soit un placement qui assure une surveillance complète des corridors;
- et tout ceci avec au plus $2k$ gardiens à disposition.

2 Recherche de la médiane en place

Dans l'algorithme de recherche de la médiane, proposez un algorithme pour effectuer la subdivision de la liste L en trois sous-listes L_1 , L_2 et L_3 en place, c'est-à-dire sans créer de nouvelles sous-listes, juste en modifiant la liste L au fur et à mesure. Procéder ainsi permet d'économiser de la mémoire (surtout lorsqu'on utilise un algorithme récursif)!

Votre algorithme prendra en entrée la liste L de taille n , ainsi que la valeur du pivot p , et sortira la liste L modifiée et composée des trois sous-listes $L = (L_1, L_2, L_3)$, ainsi que les deux nombres entiers k, ℓ tels que:

- k est la taille de la liste L_1 (i.e., le nombre d'éléments de L strictement plus petits que p);
- ℓ est la taille de L_1 et L_2 réunies (i.e., le nombre d'éléments de L plus petits ou égaux à p).

Note: Pour rappel, le pivot p est une valeur choisie au hasard dans la liste L , et les trois sous-listes L_1, L_2, L_3 doivent satisfaire aux conditions suivantes:

- les éléments de L_1 doivent être strictement plus petits que p ;
- les éléments de L_2 doivent être égaux à p ;
- les éléments de L_3 doivent être strictement plus grands que p .

3 Pour le plaisir: Un tour de cartes pour les vacances*

Alice et Bob, deux suspects connus de nos services, réalisent un tour de cartes : Alice fait mélanger un paquet standard de 52 cartes à une personne dans l'audience. Puis elle tire au hasard 5 cartes du paquet (qu'elle regarde sans les montrer à personne). Elle présente ensuite une carte à Bob (en même temps qu'à toute l'audience), puis une deuxième, puis une troisième, puis une quatrième, et s'arrête là. C'est alors que Bob annonce fièrement la valeur de la cinquième carte! Comment ont-ils fait?

Indication: Voici un exemple de déroulement d'un tel tour:

La première carte révélée par Alice est un valet de pique.

La deuxième est un as de carreau.

La troisième est un 8 de trèfle.

La quatrième est un 10 de cœur.

Et Bob annonce alors "roi de pique".