

## 1 Complexité

**Q 1.** Montrer que  $3n^2 + 2n \in O(n^2)$  (donner les constantes  $c$  et  $n_0$ ).

**Q 2.** Montrer que  $2^{n+1} \in O(2^n)$  (donner les constantes  $c$  et  $n_0$ ).

**Q 3.** Montrer que si  $f(n)$  et  $g(n)$  sont positives alors  $\max(f(n), g(n)) \in O(f(n) + g(n))$  (donner les constantes  $c$  et  $n_0$ ).

**Q 4.** Montrer que quelles que soient les constantes  $a, b > 0$  on a  $(n+a)^b \in O(n^b)$  (donner les constantes  $c$  et  $n_0$ ).

Dans les exercices qui suivent, on considère un graphe à  $n$  sommets et  $m$  arcs. La complexité d'un algorithme ou d'une structure de données est une fonction  $f$  des deux variables  $n$  et  $m$ . On a donc besoin de généraliser la définition du « grand  $O$  » pour de telles fonctions.

**Q 5.** Quel rapport y a-t-il entre  $O(m)$  et  $O(n^2)$  ?

**Q 6.** Si un graphe est connexe alors  $m \geq n - 1$ . On considère deux algorithmes résolvant le même problème, applicables à des graphes connexes uniquement. L'un a une complexité en temps, dans le pire des cas en  $O(m)$ . L'autre a une complexité en  $O(n + m)$ . Comparer ces deux algorithmes.

**Q 7.** Quelle est la complexité en temps dans le pire des cas du pseudo code suivant dans le cas où le graphe est représenté par des listes de successeurs ?

```
for tout sommet  $x$  do
  for tout successeur  $y$  de  $x$  do
    instruction ayant un coût constant
  od
od
```

**Q 8.** Même question pour le pseudo code suivant.

```
for tout sommet  $x$  do
  instruction ayant un coût constant
od
for tout arc  $a$  do
  instruction ayant un coût constant
od
```

## 2 Graphes eulériens

On rappelle qu'un graphe connexe non orienté est *eulérien* si tous ses sommets sont de degré pair sauf éventuellement deux (le degré d'un sommet  $x$  étant le nombre d'arêtes ayant  $x$  pour extrémité).

**Q1.** À quel célèbre problème historique mentionné en cours la notion de graphe eulérien s'applique-t-elle? Quel mathématicien a résolu ce problème?

La fonction suivante détermine si un graphe non orienté  $G$  est eulérien. On suppose que le graphe comporte  $n$  sommets,  $m$  arêtes et qu'il est représenté par des listes d'adjacence.

```

fonction graphe_eulérien_? (G)
begin
  nombre_de_sommets_de_degré_impair := 0
  for tout sommet x de G do
    degré := 0
    for tout voisin de x do
      degré := degré + 1
    od
    if degré est impair then
      nombre_de_sommets_de_degré_impair := nombre_de_sommets_de_degré_impair + 1
    fi
  od
  return nombre_de_sommets_de_degré_impair vaut 0 ou 2
end

```

**Q2.** Donner pour chaque affectation une estimation asymptotique (notation « O ») du nombre total de fois où elle est exécutée.

**Q3.** En déduire la complexité en temps dans le pire des cas de la fonction.

### 3 Parcours de graphe

La procédure en pseudo-code suivante est paramétrée par un graphe orienté  $G$  et un sommet distingué  $s$  de  $G$ . On note  $n$  le nombre de sommets et  $m$  le nombre d'arcs de  $G$ . La variable locale  $E$  est un ensemble de sommets.

```

procédure chmorglurs (G, s)
begin
  colorier tous les sommets en bleu, sauf s
  colorier s en vert
  E := {s}
  while E ≠ ∅ do
    choisir un sommet x ∈ E au hasard et le retirer de E
    for tous les successeurs y de x do
      if y est bleu then
        colorier y en vert
        E := E ∪ {y}
      fi
    od
    colorier x en rouge
  od
end

```

*boucle extérieure*

*boucle intérieure*

Les opérations d'ensembles utilisées dans la procédure sont :

1. l'initialisation :  $E := \{s\}$
2. le test d'égalité avec l'ensemble vide
3. la sélection et le retrait d'un élément  $x \in E$  pris au hasard
4. l'union avec un singleton :  $E := E \cup \{x\}$

Chacune de ces opérations a une complexité en temps, dans le pire des cas, en  $O(n)$  (on suppose la structure de données implantée naïvement). Chaque opération de coloriage d'un sommet a une complexité en temps, dans le pire des cas, en  $O(1)$ .

- Q 1.** Que fait la procédure ? Que peut-on dire au sujet des sommets rouges, à la fin de l'exécution de la procédure.
- Q 2.** Donner un invariant de la boucle extérieure concernant les sommets verts et l'ensemble  $E$ .
- Q 3.** Donner une estimation de la complexité en temps, dans le pire des cas, de la séquence des trois premières instructions (initialisations).
- Q 4.** Combien de fois un sommet peut-il, au plus, être inséré dans  $E$  ? Justifier en une phrase ou deux. En déduire une estimation (majoration) du nombre d'itérations effectuées par la boucle extérieure.
- Q 5.** Déduire de la question précédente une estimation de la complexité en temps, dans le pire des cas, de l'ensemble de toutes les exécutions de la première instruction de la boucle extérieure (sélection et retrait d'un élément  $x \in E$ ).
- Q 6.** En supposant le graphe implanté par des listes de successeurs, donner une estimation du nombre total d'itérations effectuées par la boucle intérieure, c'est-à-dire le nombre total de fois où la condition du if est évaluée. Justifier en une phrase.
- Q 7.** En déduire une estimation de la complexité en temps, dans le pire des cas, de l'ensemble de toutes les exécutions du corps de la boucle intérieure.
- Q 8.** En déduire une estimation de la complexité en temps, dans le pire des cas, de la procédure *chmorganlurs*.