Corrigé HC — Cours INF 431

Jean Berstel Jean-Jacques Lévy

Ecole polytechnique, 29 Avril 2003

Question 1 La fonction hauteur(g) prend un temps en $O(V + E)$.

```
static int hauteur (Graphe g) {
  int n = g.succ.length; int[ ] rang = new int[n];
  int h = -1;
  for (int x=0; x < n; ++x) rang[x] = -1;
  for (int x=0; x < n; ++x) {
    if (rang[x] == -1)
      calculerRangDe (g, x, rang);
    h = Math.max(h, rang[x]);
  }
  return h;
}

static void calculerRangDe (Graphe g, int x, int[ ] rang) {
  int r = 0;
  for (Liste ls = g.succ[x]; ls != null; ls = ls.suivant) {
    int y = ls.val;
    if (rang[y] == -1)
      calculerRangDe (g, y, rang);
    r = Math.max (r, 1 + rang[y]);
  }
  rang[x] = r;
}
```

Question 2

```
abstract class Ensemble { }

class Vide extends Ensemble { }

class NonVide extends Ensemble {
  int elt;
  Ensemble reste;
  NonVide (int x, Ensemble e) { elt = x; reste = e; }
}
```

Question 3

```
abstract class Ensemble {
  abstract int card ();
  abstract boolean contient (int x);
  abstract Ensemble ajouter (int x);
}

class Vide extends Ensemble {
  int card () { return 0; }
  boolean contient (int x) {return false; }
```

```
  Ensemble ajouter (int x) {return new NonVide (x, this); }
  public String toString () { return ""; }
}

class NonVide extends Ensemble {
  int elt;
  Ensemble reste;
  NonVide (int x, Ensemble e) { elt = x; reste = e; }

  int card () { return 1 + reste.card(); }

  boolean contient (int x) {
    return elt == x  reste.contient(x);
  }

  Ensemble ajouter (int x) {
    if ( contient(x) ) return this;
    else return new NonVide (x, this);
  }
}
```

Question 4

```
abstract class Ensemble {
  int prochain (int i, int k) {
    do ++i;
    while (i < k && contient(i));
    return i < k ? i : -1;
  }
}
```

Question 5

```
abstract class Ensemble {...}

class Vide extends Ensemble {
  public String toString () { return ""; }
}

class NonVide extends Ensemble {
  public String toString () { return elt + " " + reste; }
}
```

Question 6

Question 7 Si k est suffisamment grand, on peut prendre comme coloriage $c(x) = x$ pour tout sommet x. C'est un coloriage s'il n'y a pas de boucle.

Question 8

```
  static void colorier (Graphe g, int k) {
    int n = g.succ.length;
    int[ ] couleur = new int[n];
    for (int x=0; x < n; ++x) couleur[x] = -1;
    colorier1 (g, k, 0, couleur);
```

```
}

static void colorier1 (Graphe g, int k, int i, int[ ] couleur) {
  int n = g.succ.length;
  if (i == n) imprimerSolution (couleur);
  else {
    Ensemble e = new Vide();
    for (Liste ls = g.succ[i]; ls != null; ls = ls.suivant) {
      int y = ls.val;
      if (couleur[y] != -1)
        e = e.ajouter (couleur[y]);
    }
    for (int c = e.prochain(couleur[i], k); c != -1; c = e.prochain(c, k) ) {
      couleur[i] = c;
      colorier1 (g, k, i+1, couleur);
      couleur[i] = -1;
    }
  }
}
```

Cette dernière fonction peut aussi s'écrire sous la forme suivante plus succincte (mais moins claire) :

```
static void colorier1 (Graphe g, int k, int i, int[ ] couleur) {
  int n = g.succ.length;
  if (i == n) imprimerSolution (couleur);
  else {
    Ensemble e = new Vide();
    for (Liste ls = g.succ[i]; ls != null; ls = ls.suivant) {
      int y = ls.val;
      if (couleur[y] != -1)
        e = e.ajouter (couleur[y]);
    }
    while ((couleur[i] = e.prochain(couleur[i], k)) != -1)
      colorier1 (g, k, i+1, couleur);
  }
}
```