

Algorithmique des graphes : TD2

Alain Cournier

Stéphane Devismes

1 Exercice 1

Algorithme 1 Fonction *TesteChemin* avec résultat booléen (version générale)

Entrées

- 1: G : graphe de n sommets
- 2: L : chemin sous la forme d'une liste

Variables

- 3: x : sommet

Programme

- 4: **Si** *TestListeVide*(L) *OuSinon* *TestListeVide*(*Suite*(L)) **alors**
 - 5: renvoyer vrai
 - 6: **Fin Si**
 - 7: $x \leftarrow$ *Premier*(L)
 - 8: renvoyer *Appartient*($G, x, \text{Premier}(\text{Suite}(L))$) \wedge *TesteChemin*($G, \text{Suite}(L)$)
-

Algorithme 2 Fonction *TesteCheminElémentaire* avec résultat booléen (version générale)

Entrées

- 1: G : graphe de n sommets
- 2: L : chemin sous la forme d'une liste

Programme

- 3: renvoyer *TesteChemin*(G, L) \wedge *SansDoublons*(*Trier*(L))
-

Algorithme 3 Fonction *TesteCheminHamiltonien* avec résultat booléen (version générale)

Entrées

- 1: G : graphe de n sommets
- 2: L : chemin sous la forme d'une liste

Programme

- 3: renvoyer *TesteCheminElémentaire*(G, L) \wedge *Longueur*(L) = n
-

2 Exercice 3

Algorithme 4 Procédure *CalculeDesc* (version générale)

Entrées

- 1: G : graphe de n sommets
- 2: x : sommet

Sorties

- 3: *Descendants* : ensemble de sommets

Variables

- 4: $TEmp[0..n-1]$, $TDep[0..n-1]$: tableaux de n entiers
- 5: $nEmp$, $nDep$: entiers

Programme

- 6: $Descendants \leftarrow \emptyset$
 - 7: $nEmp \leftarrow 1$
 - 8: $nDep \leftarrow 1$
 - 9: $VisitGraphProf(G, x, Descendants, nEmp, nDep, TEmp, TDep)$
-

3 Exercice 4

Algorithme 5 Procédure *InverseTableau*

Entrées-Sorties

- 1: $T[0..n-1]$: tableau de n entiers

Variables

- 2: i : entier

Programme

- 3: **Pour** i **de** 0 **à** $n-1$ **faire**
 - 4: $T[i] \leftarrow n+1 - T[i]$
 - 5: **Fin Pour**
-

Algorithme 6 Procédure *ExtLinéaire* (version générale)

Entrées

- 1: G : graphe de n sommets

Sorties

- 2: $E[0..n-1]$: tableau de n entiers

Variables

- 3: *Exploré* : ensemble de sommets
- 4: $TEmp[0..n-1]$: tableau de n entiers
- 5: $nEmp$, $nDep$: entiers
- 6: u : sommet

Programme

- 7: $nEmp \leftarrow 1$
 - 8: $nDep \leftarrow 1$
 - 9: $Exploré \leftarrow \emptyset$
 - 10: **Pour tout** $u \in X$ **faire**
 - 11: $VisitGraphProf(G, x, Exploré, nEmp, nDep, TEmp, E)$
 - 12: **Fin Pour**
 - 13: $InverseTableau(E, n)$
-

4 Exercice 5

Algorithme 7 Fonction *ChoisirSucc* avec résultat sommet

Entrées

- 1: x : sommet
- 2: $TEmp[0..n-1]$: tableau des numéros d'empilement
- 3: $TDep[0..n-1]$: tableau des numéros de dépilement

Variables

- 4: y, z : sommet

Programme

- 5: $y \leftarrow x$
 - 6: **Pour tout** $z \in Succ(x)$ **faire**
 - 7: **Si** $TDep[z] > TDep[x] \wedge TEmp[z] < TEmp[x] \wedge TDep[z] > TDep[y]$ **alors**
 - 8: $y \leftarrow z$
 - 9: **Fin Si**
 - 10: **Fin Pour**
 - 11: renvoyer y
-

Algorithme 8 Procédure *CFC*

Entrées

1: G : graphe de n sommets

Sorties

2: $T[0..n-1]$: tableau de n sommets

Variables

3: *Exploré* : ensemble de sommets

4: $TEmp[0..n-1]$: tableau de n entiers

5: $TDep[0..n-1]$: tableau de n entiers

6: $OrdreEmp[1..n]$: tableau de n sommets

7: $OrdreDep[1..n]$: tableau de n sommets

8: $nEmp, nDep, DepCour$: entiers

9: P, P' : piles de sommets

10: x, y : sommets

11: i : entier naturel

Programme

12: $nEmp \leftarrow 1$

13: $nDep \leftarrow 1$

14: *Exploré* $\leftarrow \emptyset$

15: **Pour tout** $x \in X$ **faire**

16: *VisitGraphProf*($G, x, \textit{Exploré}, nEmp, nDep, TEmp, TDep$)

17: $T[x] \leftarrow x$

18: **Fin Pour**

19: $DepCour \leftarrow 1$

20: $P \leftarrow \textit{PileVide}()$

21: **Pour tout** $x \in X$ **faire**

22: $OrdreEmp[TEmp[x]] \leftarrow x$

23: $OrdreDep[TDep[x]] \leftarrow x$

24: **Fin Pour**

25: **Pour** i **de** 1 **à** n **faire**

26: *Empiler*($OrdreEmp[i], P$)

27: **Tant que** $\text{NON } \textit{TestPileVide}(P)$ **EtAlors** $\textit{Sommet}(P) = \textit{OrdreDep}[DepCour]$ **faire**

28: $P' \leftarrow \textit{Copie}(P)$

29: $DepCour \leftarrow DepCour + 1$

30: $x \leftarrow \textit{Dépiler}(P)$

31: $y \leftarrow \textit{ChoisirSucc}(G, x, TEmp, TDep)$

32: **Tant que** $\textit{Sommet}(P') \neq y$ **faire**

33: $\textit{Union}(T, x, \textit{Dépiler}(P'))$

34: **Fin Tant que**

35: $\textit{Union}(T, x, y)$

36: **Fin Tant que**

37: **Fin Pour**

5 Exercice 6

Par cette méthode, on calcule aussi les composantes fortement connexe (Algorithme de Kosaraju).