

Correction exercices du TD
Algorithmique et graphes

Exercice 10

0.1 Primitives usuelles sur le type graphe

Remarque :

Dans cet exercice on considère des graphes orientés simples.

Un graphe est un couple $G = (V, E)$ où V est un intervalle de la forme $[1, n]$ avec $n \geq 0$ ($[1, 0] =$ ensemble vide) et où E est inclus dans $V \times V$.

La fonction graphevide

Signature : graphevide : () \rightarrow graphe.

Exemple :

$\rightarrow (\emptyset, \emptyset)$

La fonction estvide

Signature : estvide : graphe \rightarrow booléen.

Exemples :

$(\emptyset, \emptyset) \rightarrow$ vrai

$(\{1\}, \{(1, 1)\}) \rightarrow$ faux

Spécification :

Entrée : le graphe $G = (V, E)$.

Sortie : le booléen $(V = \emptyset)$.

La fonction nbsommet

Signature : nbsommet : graphe \rightarrow entier.

Exemple :

$(\{1\}, \emptyset) \rightarrow 1$.

Spécification :
Entrée : le graphe $G=(V,E)$.
Sortie : le cardinal de V .

La fonction ajoutersommet

Signature : ajoutersommet : graphe \rightarrow graphe.

Exemple :
 $([1,2],\{(1,2)\}) \rightarrow ([1,3],\{(1,2)\})$.

Vu que dans le graphe l'ensemble des sommets doit être un intervalle de la forme $[1 n]$ alors le seul sommet qu'on peut ajouter est l'élément $\{n+1\}$, et si on décide de plus que ce n'est pas intéressant d'ajouter un sommet déjà existant alors la signature de l'opération sera :

ajoutersommet : graphe $\rightarrow ([1],\emptyset)$ si le graphe est vide et $(V \cup \{max(V)+1\},E)$ sinon.

La fonction enleversommet

Signature : enleversommet : graphe \dashrightarrow graphe.

Exemple :
 $([1,3],\{(1,3)\}) \rightarrow ([1,2],\emptyset)$.

Si on décide que supprimer un sommet à un graphe qui n'en possède pas n'a pas de sens alors on retient pour cette opération la spécification suivante :
Entrée : un graphe non vide $G=(V,E)$.
Sortie : le graphe $G'=(V - max(V), E \cap (V - max(V)) \times (V - max(V)))$.

La fonction estisolé

Signature : estisolé : (graphe x sommet) \rightarrow booléen.

Spécification :
Entrée : un graphe $G=(V,E)$ et un sommet s avec $(\{s\} \in V)$.
Sortie : le booléen $(E \cap (V - \{s\}) \times (V - \{s\}))$.

La fonction existarc

Signature : existarc : (graphe x sommet x sommet) \rightarrow booléen.

Spécification :
Entrée : un graphe $G=(V,E)$ et deux sommets s et t .

Sortie : le booléen $((s, t) \in E)$.

La fonction ajouterarc

Signature : ajouterarc : (graphe x sommet x sommet) \rightarrow graphe.

Spécification :

Entrée : un graphe $G=(V,E)$ et deux sommets s et t avec $(s, t) \in V \times V$.

Sortie : le graphe $G'=(V, E \cup \{(s, t)\})$.

On considère qu'il est inutile d'ajouter un arc qui existe déjà d'où la condition en entrée de la fonction.

La fonction supprimerarc

Signature : supprimerarc : (graphe x sommet x sommet) \rightarrow graphe.

Spécification :

Entrée : un graphe $G=(V,E)$ et deux sommets s et t avec $(s, t) \in V \times V$ et $(s, t) \in E$.

Sortie : le graphe $G'=(V, E - (s, t))$.

On considère qu'il est inutile de supprimer un arc qui n'existe pas d'où la deuxième condition en entrée de la fonction.

La fonction echanger

Signature : echanger : (graphe x sommet) \rightarrow graphe.

On suppose que l'échange s'effectue avec le dernier élément de V .

Spécification :

Entrée : un graphe $G=(V,E)$ et un sommet s avec $s \in V$ et $s \neq \max(V)$.

Sortie : le graphe $G'=(V, (f(a), f(b)) | (a, b) \in EXE)$

où $f(x)$ est défini par :

$$\begin{array}{ll} \max(V) & \text{si } x = s \\ s & \text{si } x = \max(V) \\ x & \text{sinon} \end{array}$$

La fonction successeur

Signature : successeur : (graphe x sommet) \rightarrow ensemble de sommets.

Spécification :

Entrée : un graphe $G=(V,E)$ et un sommet s .

Sortie : l'ensemble $\{t \mid (s, t) \in E\}$.

0.2 Implémentation des fonctions usuelles en utilisant une matrice de booléen

La structure graphe

On choisit de représenter un graphe comme une structure contenant deux champs :

Le premier contenant la matrice de booléens représentant le graphe et le second contenant le nombre de sommet nbsommet.

Structure Graphe :

- M = matrice de booléens de taille nbsommet x nbsommet
- nbsommet = entier donnant le nombre de sommet du graphe

La fonction graphevide

Algorithme 1 Fonction graphevide

ENTRÉES:

SORTIES: graphe

```
G ← Graphevide();  
G.matrice ← matricevide();  
G.nbsommet ← 0;  
retourner G
```

La fonction estvide

Algorithme 2 Fonction estvide

ENTRÉES: G : graphe

SORTIES: booléen

```
retourner (G.nbsommet=0)
```

La fonction nbsommet

Algorithme 3 Fonction nbsommet

ENTRÉES: G : graphe

SORTIES: entier

retourner G.nbsommet;

La fonction ajoutersommet

Algorithme 4 Fonction ajoutersommet

ENTRÉES: G : graphe

SORTIES: graphe

n ← G.nbsommet;

G' ← graphediscret(n+1);

Pour i de 1 à n **Faire**

Pour j de 1 à n **Faire**

 G'.matrice[i][j] ← G.matrice[i][j];

FinPour

FinPour

retourner G'

La fonction enleversommet

Algorithme 5 Fonction enleversommet

ENTRÉES: G : graphe

SORTIES: graphe

G.matrice ← enlevercolonne(G.matrice,G.nbsommet);

G.matrice ← enleverligne(G.matrice,G.nbsommet);

retourner G

La fonction estisolé

Algorithme 6 Fonction estisolé

ENTRÉES: G : graphe, k : sommet

SORTIES: booléen

Pour i de 1 à n **Faire**

Si existearc (G,j,k) **Alors**

 retourner faux ;

FinSi

FinPour

Pour j de 1 à n **Faire**

Si existearc (G,k,j) **Alors**

 retourner faux ;

FinSi

FinPour

retourner vrai

La fonction existearc

Algorithme 7 Fonction existearc

ENTRÉES: G : graphe, i,j : sommets

 retourner ($G.matrice[i][j]$) ;

La fonction supprimerarc

Algorithme 8 Fonction supprimerarc

ENTRÉES: G : graphe , s,t : sommets

SORTIES: graphe

G.matrice[s][t] \leftarrow 0;

retourner G;

La fonction ajouterarc

Algorithme 9 Fonction ajouterarc

ENTRÉES: G : graphe , s,t : sommets

SORTIES: graphe

G.matrice[s][t] \leftarrow 1;

retourner G;

La fonction successeur

Algorithme 10 Fonction successeur

ENTRÉES: G : graphe

SORTIES: ensemble de sommets

M \leftarrow G.matrice;

n \leftarrow G.nbsommet;

E \leftarrow ensemblevide();

Pour i de 1 à n **Faire**

Si matrice[s][i] **Alors**

 ajouter(i,E);

FinSi

FinPour

retourner E

La fonction echanger

Algorithme 11 Fonction echanger

ENTRÉES: G : graphe, s : sommet

SORTIES: graphe

n ← G.nbsommet ;

T ← constructTab(n)(1)(0) ;

Pour i de 1 à n **Faire**

 T[i] ← G.matrice[s][i] ;

 G.matrice[s][i] ← G.matrice[n][i] ;

 G.matrice[n][i] ← T[i] ;

 T[i] ← G.matrice[i][s] ;

 G.matrice[i][s] ← G.matrice[i][n] ;

 G.matrice[i][n] ← T[i] ;

FinPour

retourner G ;
