



Raffinages-PageRank-EF03

THEVENET Louis

MORISSEAU Albin

Table des matières

1. Introduction	3
2. Liste des modules	3
3. Raffinages	3
3.1. Programme_Principal	3
3.2. Module Lire Graphe	5
3.3. Module PageRank_Result	7
3.4. Module PageRank	9
3.5. Module PageRank_Pleine	10
3.6. Module PageRank_Creuse	11
4. Grille d'évaluation des raffinages	12
5. Tests	12
5.1. Traitement de la commande	13
5.2. Modules	13

1. Introduction

2. Liste des modules

- Programme_Principal
- Module Lire_Graphe
- Module PageRank_Result
- Module PageRank
 - Module PageRank_Pleine
 - Module PageRank_Creuse

3. Raffinages

3.1. Programme_Principal

3.1.1. Description

Le point d'entrée du programme, il traite les arguments et les transmet ensuite au Module PageRank. Il initialise les différentes variables à leurs valeurs par défaut :

$$\begin{aligned}\alpha &:= 0.85 \\ k &:= 150 \\ \varepsilon &:= 0.0\end{aligned}$$

Il traite ensuite les arguments en mettant à jour les variables si besoin.

Il vérifie finalement la conformité des valeurs à la spécification :

$$\begin{aligned}\alpha &\in [0, 1] \\ \varepsilon &\geq 0\end{aligned}$$

Un seul algorithme choisi (Creuse ou pleine)

Préfixe non vide

3.1.2. Raffinages

```
1  R0 : Répondre à l'appel au programme par l'utilisateur
2
3  R1 : Comment "Répondre à l'appel au programme" ?
4    Traiter les arguments
5        Arguments: in,
6        alpha : out,
7        k : out,
8        epsilon : out,
9        creuse : out,
10       pleine : out,
11       prefixe : out,
12       fichier_graphe : out
13
14   Appeler le module PageRank
15       alpha : in,
16       k : in,
17       epsilon : in,
18       creuse : in,
19       pleine : in,
20       prefixe : in,
```

```

21         fichier_graphe : in
22         indices : out, -- comment les indices ont été changés après le tri
23         resultat : out
24
25 R2 : Comment "Traiter les arguments" ?
26     Initialiser les variables
27         alpha : out,
28         k : out,
29         epsilon : out,
30         creuse : out,
31         pleine : out,
32         prefixe : out,
33         fichier_graphe : out
34
35     Récupérer le nom du fichier
36
37     Pour Argument allant de 1 à Nombre_Argument _ 1 Faire
38         Traiter argument
39             Nom_Argument : in out,
40             Argument : in out,
41             alpha : in out,
42             k : in out,
43             epsilon : in out,
44             creuse : in out,
45             pleine : in out,
46             prefixe : in out,
47             fichier_graphe : in out
48     Fin Pour
49
50
51 R3 : Comment "Initialiser les variables"
52     alpha := 0.85
53     k := 150
54     epsilon := 0.0
55     creuse := true
56     pleine := false
57     prefixe := "output"
58     fichier_graphe := ""
59
60 R3: Comment " Récupérer le nom du fichier " ?
61 Si Nombre_argument < 1 Faire
62     Lever Exception
63 Sinon
64     fichier_graphe := Argument(Nombre_argument)
65 Fin Si
66
67 R3 : Comment "Traiter argument" ?
68 Si Taille(Argument) = 2 et Argument(1) = '-' Alors
69     Nom_Argument = Argument(2) -- deuxième caractère
70     Selon Nom_Argument Dans
71         "A" => Si Argument_suivant > 0 et Argument_suivant < 1 Alors
72             alpha := argument
73         Sinon
74             Afficher("Vous devez respecter les conditions sur Alpha")
75             Lever l'exception Argument_error

```

```

76         Fin Si
77     "K" => Si Argument_suivant>0 Alors
78         k := argument
79     Sinon
80         Afficher("Vous devez respecter les conditions sur k")
81         Lever l'exception Argument_error
82     Fin Si
83     "E" => Si Argument_suivant>0 Alors
84         epsilon := argument
85     Sinon
86         Afficher("Vous devez respecter les conditions sur Epsilon")
87         Lever l'exception Argument_error
88     Fin Si
89     "P" => pleine := true
90     "C" => Si pleine=true Alors
91         Afficher ("Attention, vous ne pouvez pas activer à la fois
92 le mode Creuse et à la fois le mode Pleine");
93         Lever l'exception Argument_error
94     Fin Si
95     "R" => prefixe := argument
96     Autres => Lever l'exception Mauvais_Argument_Error
97 Fin Selon
98 Fin Si
99 R4 : Comment "Lever l'exception Argument_error"
100 Afficher la procédure d'aide
101 R4 : Comment "Lever l'exception Mauvais_Argument_Error"
102 Afficher(" Vous devez renseigner uniquement les arguments pris en charge par
103 le programme ")
104 Afficher la procédure d'aide

```

3.2. Module Lire Graphe

3.2.1. Description

3.2.2. Raffinages

```

1  R0 : Lire fichier_graphe
2
3  R1 : Comment "Lire le fichier_graphe" ?
4  taille_graphe := Lire Entier dans fichier_graphe
5  H := Initialiser_Matrice();
6
7  Remplir la matrice H
8  H : in out
9  Pondérer la matrice H
10 H : in out
11
12 R2 : Comment "Remplir la matrice H" ?
13 Pour chaque ligne de fichier_graphe Faire
14     A := Lire Entier dans fichier_graphe
15     B := Lire Entier dans fichier_graphe
16     H(A,B) := 1
17 Fin Pour
18

```

```
19 R2 : Comment "Pondérer la matrice H" ?
20   Pour i de 1 à taille_graphe Faire
21     total := 0
22     Pour j de 1 à taille_graphe Faire
23       total := total + H(i,j)
24     Fin Pour
25
26     Si total != 0 Alors
27       Pour j de 1 à taille_graphe Faire
28         H(i,j) := H(i,j) / total
29       Fin Pour
30     Sinon
31       Rien
32   Fin Pour
```

3.3. Module PageRank_Result

3.3.1. Description

Ce module permet d'interagir avec le résultat. Notamment le tri des pages selon leur poids.

```
1  N : Entier est générique
2
3  Type T_Resultat EST enregistrement
4    Taille : Entier;
5    Poids : tableau de flottants de taille N
6    Indices : tableau d'indices de taille N
7  end enregistrement
```

Il fournit ces opérations :

- Initialiser
- Norme_Au_Carre
- Combi_Lineaire
- Trier
- Enregistrer

3.3.2. Raffinages

```
1  R0 : Initialiser le résultat
2      Result : in T_Resultat
3
4  R1 : Comment "Initialiser le résultat" ?
5      Pour i allant de 1 à Result.Taille Faire
6          Result.Poids(i) := 0.0;
7          Result.Indices(i) := i;
8      fin Pour
```

```
1  R0 : Calculer la norme au carré
2      Res : in T_Resultat
3
4  R1 : Comment "Calculer la norme au carré" ?
5      Resultat := 0.0;
6      Pour i allant de 1 à Result.Taille Faire
7          Resultat := Resultat + Res.Poids(i)*Res.poids(i);
8      fin Pour
9      retour Resultat;
```

```
1  R0 : Calculer la combinaison linéaire
2      A,B : in tableau de flottants de taille N
3      lambda, mu : Flottants
4
5  R1 : Comment "Calculer la combinaison linéaire" ?
6      Resultat : tableau de flottants de taille N
7
8      Initialiser(Resultat);
9      Pour i allant de 1 à Result.Taille Faire
10         Resultat. :=
11
```

```
12   fin Pour
    retour Resultat;
```

```
1  R0 : Enregistrer le résultat
2
3  R1 : Comment "Enregistrer le résultat" ?
4      Produire le fichier PageRank
5          indices : in,
6          prefixe : in
7
8      Produire le fichier Poids
9          resultat : in,
10         prefixe : in,
11         taille_graphe : in,
12         k : in,
13         alpha : in
14
15  R2 : Comment "Produire le fichier PageRank" ?
16      Pour i de 1..taille_graphe Faire
17          Ecrire indices(i) dans le fichier prefixe.pr
18      Fin Pour
19
20  R2 : Comment "Produire le fichier Poids" ?
21      Ecrire taille_graphe alpha k dans le fichier prefixe.prw
22      Pour i de 1..taille_graphe Faire
23          Ecrire resultat(i) dans le fichier prefixe.prw
24      Fin Pour
```

```
1  R0 : Modifier un élément de la matrice M
2      M : in out T_Matrice
3      I : in Integer
4      J : in Integer
5      Nouveau : in Long_Float
6  R1 : Comment "Modifier une Matrice" ?
7      M.Mat(I, J) := Nouveau;
```

```
1  R0 : Obtenir un Élément de la Matrice M
2      M : in T_Matrice
3      I : in Integer
4      J : in Integer
5      Resultat : out Long_Float
6  R1 : Comment "Obtenir un Élément de la Matrice" ?
7      Resultat := M.Mat(I, J);
```


3.4. Module PageRank

3.4.1. Description

Implémente l'algorithme PageRank en utilisant le Module PageRank_Pleine ou Module PageRank_Creuse. Ce module renvoie un vecteur de Poids des différentes pages web.

3.4.2. Raffinage

```
1  R0 : Répondre à l'appel du programme principal
2
3  R1 : Comment "Répondre à l'appel du programme principal" ?
4  Paramètres du module :
5      alpha : in,
6      k : in,
7      epsilon : in,
8      creuse : in,
9      pleine : in,
10     prefixe : in,
11
12
13
14  Si Pleine Alors
15      Lire fichier_graphe_plein (via module Fichier Graphe)
16          fichier_graphe : in,
17          H : out,
18          taille_graphe : out
19
20      Calculer la matrice de Google G
21          creuse : in,
22          pleine : in,
23          alpha : in,
24          H : in,
25          taille_graphe : in
26
27  Sinon
28      Lire fichier_graphe_plein (via module Fichier Graphe)
29          fichier_graphe : in,
30          H : out,
31          facteurs : out, -- représente le nombre de composantes non nulles
de chaque ligne
32          taille_graphe : out
33  Fin Si
34
35  Initialiser Pi_transpose
36      taille_graphe : in,
37      Pi_transpose : out
38
39  Si Pleine
40      Appliquer la relation de récurrence
41          Pi_transpose : in,
42          G : in,
43          taille_graphe : in
44  Sinon
45      Appliquer la relation de récurrence
46          Pi_transpose : in,
```

```

47         H : in,
48         Facteurs : in
49         taille_graphe : in
50
51     trier Pi_transpose
52         Pi_transpose : in,
53         resultat : out
54
55     Enregistrer le resultat (via module Engresitrer Résultat)
56         taille_graphe : in,
57         k : in,
58         alpha : in,
59         indices : in,
60         resultat : in,
61         prefixe : in
62
63     R2 : Comment "Initialiser Pi_transpose" ?
64     Pi_transpose = new tableau (1..taille_graphe) DE Double
65     Pour i allant de 1..taille_graphe Faire
66         Pi_transpose(i) := 1/taille_graphe
67     Fin Pour
68
69     R2 : Comment "Trier Pi_transpose"
70     resultat := new T_Resultat (Pi_transpose trié par ordre décroissant,
    Permutation des indices du tri)

```

3.5. Module PageRank_Pleine

3.5.1. Description

Ce module est appelé par le module **PageRank** et renvoie la Matrice de Google **G** obtenue à partir de la matrice d'adjacence pondérée **S** des différents référencements des pages web entre elles. On ne se préoccupe pas du fait que la machine soit creuse.

3.5.2. Raffinage

```

1  R0 : Calculer la matrice de Google G
2
3  R1 : Comment "Calculer la matrice de Google G" ?
4      Calculer la matrice S
5          alpha : in,
6          taille_graphe : in,
7          H : in,
8          S : out
9
10
11     Calculer la matrice G
12         alpha : in,
13         taille_graphe : in,
14         S : in,
15         G : out
16
17
18     R2 : Comment "Calculer la matrice S" ?
19     S := H

```

```

20  Pour i de 1 à taille_graphe Faire
21      est_nul := true
22      Tant que est_nul Faire
23          est_nul := est_nul ET (S(i,j)=0)
24      Fin Tant que
25
26      Si est_nul Alors
27          Pour j de 1 à taille_graphe Faire
28              S(i,j) := 1/taille_graphe
29          Fin Pour
30      Fin Si
31
32  Fin Pour
33
34  R2 : Comment "Calculer la matrice G" ?
35  G = alpha * S
36  Pour i de 1 à taille_graphe Faire
37      Pour j de 1 à taille_graphe Faire
38          G(i,j) := G(i,j) + (1-alpha)/taille_graphe
39      Fin Pour
40  Fin Pour

```

```

1  R0 : Appliquer la relation de récurrence
2
3  R1 : Comment "Appliquer la relation de récurrence" ?
4  Pour i allant de 1..k Faire
5      Calculer Pi_transpose
6          Pi_transpose : in out
7          G : in
8  Fin Pour
9
10 R3 : Comment "Calculer Pi_transpose" ?
11 Pour j allant de 1..taille_graphe Faire
12     tmp :=0
13     Pour i allant de 1..taille_graphe Faire
14         tmp := tmp + Pi_transpose(i) * G(i, j)
15     Fin Pour
16     Pi_transpose(j) := tmp
17 Fin Pour

```

3.6. Module PageRank_Creuse

3.6.1. Description

Ce module est appelé par le module PageRank et , étant donné que la matrice de Google est très creuse, ne calcule pas explicitement celle ci. On calcule les poids directement sans passer par un calcul matriciel.

3.6.2. Raffinage

```

1  R0 : Appliquer la relation de récurrence
2
3  R1 : Comment "Appliquer la relation de récurrence" ?
4  Pour i allant de 1..k Faire

```

```

5      Calculer Pi_transpose
6          Pi_transpose : in out
7          G : in
8      Fin Pour
9
10 R3 : Comment "Calculer Pi_transpose" ?
11 beta := (1.0 - Alpha) / Taille
12 Pour j allant de 1..taille_graphe Faire
13     tmp := 0
14     Pour i allant de 1..taille_graphe Faire
15         tmp := tmp + Pi_transpose(i) * (G(i, j) * Facteurs(I) * Alpha + beta)
16     Fin Pour
17     Pi_transpose(j) := tmp
18 Fin Pour

```

4. Grille d'évaluation des raffinages

		Eval. étudiant	Justif./Comm.	Eval. enseignant
Forme	Respect de la syntaxe	TB		
	Ri : Comment « ... une action complexe ... » ? des actions combinées avec des structures de controle			
	Rj : ...			
	Verbe à l'infinitif pour les actions complexes	TB		
	Nom ou équivalent pour expressions complexes	TB		
	Tous les Ri sont écrits contre la marge et espacés	TB		
	Les flots de données sont définis	TB		
	Pas trop d'actions dans un raffinement (moins de 6)	B		
	Bonne présentation des structures de contrôle	TB		
Fond	Le vocabulaire est précis	TB		
	Le raffinement d'une action décrit complètement cette action	TB		
	Les flots de données sont cohérents	TB		
	Pas de structure de contrôle déguisée	TB		
	Qualité des actions complexes	TB		

5. Tests

5.1. Traitement de la commande

On utilise le fichier `exemple-fichier.txt`.

- $\alpha < 0$
`./programme_principal -P -A -0.90 -K 20 ./exemple-fichier.txt`
- $\alpha > 1$
`./programme_principal -P -A 1.90 -K 20 ./exemple-fichier.txt`
- $K < 0$
`./programme_principal -P -K -20 ./exemple-fichier.txt`
- $\varepsilon < 0$
`./programme_principal -P -E -20.0 ./exemple-fichier.txt`
- Creuse = Pleine
`./programme_principal -P -C ./exemple-fichier.txt`
- Pas de fichier d'entrée `./programme_principal -P -C`
- Mauvais fichier d'entrée `./programme_principal -P -C ./exemple-fichier-qui-existe-pas.txt`

Le programme affiche bien des erreurs dans tous ces cas.

5.2. Modules

On crée trois fichiers de tests unitaires :

- `test_vecteurs_creux.adb`
- `test_matrices_pleines.adb`
- `test_matrices_creuse.adb`