

# Raffinage PageRank

Julien Blanchon (1SN Groupe B)

## 0.1 Retour KJR

- Si vous pouviez mettre des virgules dans la liste des flots de données, ce serait plus facile à lire. Aussi pour les flots, le fait de les écrire au bout de chaque étape permet de mieux voir comment des données circulent d'une étape à une autre.
- Quand vous déterminez une exception, vous pourriez simplement commencer le calcul avec le paramètre par défaut (`alpha` ou `max_iter`), et indiquer avant le lancement les paramètres réellement pris en compte.
- Il manque le raffinement R3 des calculs de  $G$ , du produit matriciel.

## 1 Raffinages:

### 1.1 Consigne :

1. L'exécutable `./pagerank` doit comporter 4 arguments dont 3 optionnels:
  - `fichier.net` est le chemin vers le fichier décrivant le réseau. Ex: 

```
6
0 1      0 2      2 0      2 1      2 4      3 4      3 5      4 3
4 5      5 3
```
  - `-P` optionnel: Permet d'utiliser l'implantation `Google_Naive`. Par défaut (sans `-P`), on lance l'implantation `Google_Creuse`.
  - `-I <int>` optionnel: Permet de spécifier le nombre maximal d'itérations. Par défaut 150.
  - `-A <float>` optionnel: Permet de spécifier la valeur d'`\alpha`. Par défaut 0.85.
  - On adopte une programmation défensive pour l'appel de `./pagerank`.
  - `./pagerank -P -I 150 -A 0.90 exemple_sujet.net`
2. Les résultats du programme seront écrits dans des fichiers d'extensions `.p` (pour le fichier pagerank) et `.ord` (pour le fichier poids) avec le même préfixe que le fichier `.net`.
  - `*.p` Liste les poids des nœuds par ordre décroissant.
  - `*.ord` Liste l'identifiant des nœuds par poids décroissant (PageRank croissant).

## 1.2 Raffinage 0 : Programme Principale

---

**Raffinage 0:** Programme Principale

---

**Input** : stdin("-P -I 150 -A 0.90 exemple\_sujet.net")

**Output:** exemple\_sujet.p:Fichier exemple\_sujet.ord:Fichier

```
1 Vérifier l'intégrité des arguments et les chargés en mémoire
2 if Integre then
3   | Calculer la matrice de Google  $G$ 
4   | Calculer le vecteur de poids  $\pi$ 
5   | Trier  $\pi$  et déterminer le PageRank  $Pk$ 
6   | Écrire les sorties
7 else
8   | Afficher la documentation
9 end
```

---

## 1.3 Raffinage 1 : Vérifier l'intégrité des arguments et les chargés en mémoire

---

**Raffinage 1:** Vérifier l'intégrité des arguments et les chargés en mémoire

---

**Input** : stdin("-P -I 150 -A 0.90 exemple\_sujet.net")

**Output:** type\_matrice:Integer max\_iter:Integer  $\alpha$ :Float integre:Boolean

```
1 Initialiser les paramètres avec les valeurs par défauts
2 begin
3   | Mettre à jour les paramètres
4   exception
5     | Max_Iter_Arg_Exception =>
6     |   integre := False
7     |   Afficher("-I max_iter")
8     | Alpha_Arg_Exception =>
9     |   integre := False
10    |   Afficher("-A alpha")
11    | Net_Arg_Exception =>
12    |   integre := False
13    |   Afficher("chemin/fichier.net")
14  end
15 end
```

---

## 1.4 Raffinage 2 : Initialiser les paramètres avec les valeurs par défauts

---

**Raffinage 2:** Initialiser les paramètres avec les valeurs par défauts

---

**Input :**

**Output:** type\_matrice max\_iter  $\alpha$  Integre

```
1 type_matrice := 0
2 max_iter := 150
3  $\alpha$  := 0.85
4 integre := True
```

---

## 1.5 Raffinage 2 : Mettre à jour les paramètres

---

**Raffinage 1:** Mettre à jour les paramètres

---

**Input :** Argument:String Argument\_Count:Integer

**Output:** type\_matrice:Integer max\_iter:Integer  $\alpha$ :Float integre:Boolean  $A$

```
1 if not( $1 \leq \text{Argument\_Count} \leq 6$ ) then
2   | Raise Argument_Exception
3 end
4 I := 1
5 while I  $\neq$  Argument_Count do
6   | if Argument(I) := "-P" then
7     | type_matrice := 1
8   | else if Argument(I) = "-I" then
9     | max_iter := Argument(I+1)
10    | I := I+1
11  | else if Argument(I) = "-A" then
12    |  $\alpha$  := Argument(I+1)
13    | I := I+1
14  | else
15    | Ouvrir le fichier
16    | Construire la matrice adjacente  $A$ 
17  | end
18  | I:=I+1
19 end
```

---