Recherche Opérationnelle et Optimisation
TP7 : Algorithme de flot maximal

Responsable : Julien Ah-Pine

M1 Informatique 2017/2018

1 Quelques commandes utiles

1. Entrez et exécutez les commandes suivantes pour initialiser un ensemble de sommets et un réseau avec capacités (exemple du cours slide 168) :

```
> X=1:7
> A=rbind(c(0,5,8,0,0,0,0),c(0,0,0,4,2,0,0),c(0,0,0,0,5,2,0),
c(0,0,0,0,0,0,7),c(0,0,0,0,0,0,3),c(0,0,0,0,0,0,3),c(0,0,0,0,0,0,0))
> P=matrix(0,nrow=length(X),ncol=length(X))
```

2. Entrez et exécutez les commandes suivantes les unes après les autres :

```
> R1=A-P>0
> R2=t(P)>0
> C=R1|R2
```

   Que fait chacune de ces commandes ?

3. Entrez et exécutez les commandes suivantes les unes après les autres :

```
> S=c(1,2)
> Sb=setdiff(X,S)
> which(matrix(C[S,Sb]==TRUE,nrow=length(S),ncol=length(Sb)),arr.ind=TRUE)
```

   Que fait chacune de ces commandes ?

4. Entrez et exécutez les commandes suivantes :

```
> is.element(3,S)
> is.element(1,S)
```

   Que fait la commande `is.element` ?

2 Algorithme de Ford-Fulkerson

4. Ecrivez une fonction `Ford_Fulkerson` qui prend en entrée un ensemble de sommets `X`, un réseau de capacités (de poids non négatifs) `A`, un sommet source `s` et un sommet puits `p` de `X` et qui donne en sortie, le flot `P` de valeur maximale obtenus par l'algorithme de Ford-Fulkerson que nous rappelons ci-dessous :

Input : $G = [X, U, C]$, φ un flot réalisable
1 $m_s \leftarrow (\infty, +)$ et $S = \{s\}$
2 **Tant que** $\exists (j \in \overline{S}, i \in S) : (c_{ij} - \varphi_{ij} > 0) \vee (\varphi_{ji} > 0)$ **faire**
3   **Si** $c_{ij} - \varphi_{ij} > 0$ **faire**
4         $m_j \leftarrow (i, \alpha_j, +)$ avec $\alpha_j = \min\{\alpha_i, c_{ij} - \varphi_{ij}\}$
5   **Sinon Si** $\varphi_{ji} > 0$ **faire**
6         $m_j \leftarrow (i, \alpha_j, -)$ avec $\alpha_j = \min\{\alpha_i, \varphi_{ji}\}$
7   **Fin Si**
8     $S \leftarrow S \cup \{j\}$
9   **Si** $j = p$ **faire**
10         $V(\varphi) \leftarrow V(\varphi) + \alpha_p$
11   Aller en 14
12   **Fin Si**
13 **Fin Tant que**
14 **Si** $p \in S$ **faire**
15   **Tant que** $j \neq s$ **faire**
16   **Si** $m_j(3) = +$ **faire**
17             $\varphi_{m_j(1)j} \leftarrow \varphi_{m_j(1)j} + \alpha_p$
18   **Sinon Si** $m_j(3) = -$ **faire**
19             $\varphi_{jm_j(1)} \leftarrow \varphi_{jm_j(1)} - \alpha_p$
20   **Fin Si**
21         $j \leftarrow m_j(1)$
22   **Fin Tant que**
23   Aller en 1
24 **Sinon faire**
25   **Output :** φ
26 **Fin Si**

5. Testez votre fonction sur l'exemple du cours (slide 168)