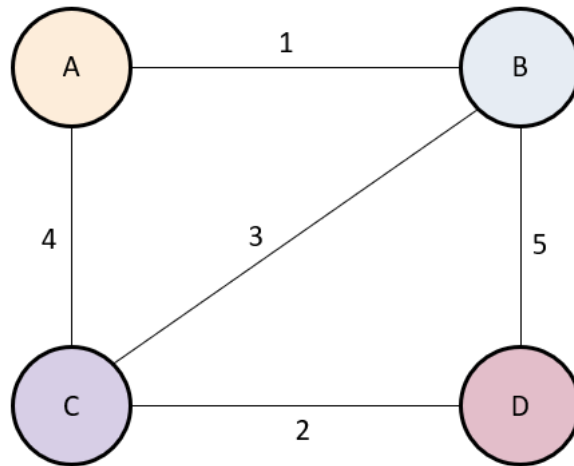


Algorithme GHS83 – dérouler l’algorithme – p.14



Déroulement pour 4 sommets → environ 1 heure.



	A		B			C			D	
Canal[j] tableau de taille Vois à valeur dans {basic, branch, reject}	B	C	A	C	D	A	B	D	B	C
	Basic Branch	Basic Basic Reject	Basic Branch Branch	Basic Basic Branch	Basic Basic Basic Reject	Basic Basic Basic Reject	Basic Basic Branch	Basic Branch	Basic Basic Reject	Basic Branch
Niv Entier	0 1 2		0 1 2			0 1 2			0 1 2	
Etat A valeur dans {find, found}	Found Find Found Find Found		Found Find Found Find Found			Found Find Found Find Found			Found Find Found Find Found	
Reçu Entier	0 0 0		0 0 0 1			0 0 0 1			0 0 0	
Nom Réel	1 3		1 3			2 3			2 3	
Père Identifiant	B B		A C			D B			C C	
Mcan Identifiant	x C x		x C x			x B x			x B x	
Testcan Identifiant	C x C x		C x D x			B x A x			B x B x	
Mpoids	∞		∞			∞			∞	

Réel	4 ∞	3 ∞	3 ∞	5 ∞
------	---------------	---------------	---------------	---------------

1. BLOC 1 : initialisation → Couleur

- A envoi (*Connect, 0*) à B.
- B envoi (*Connect, 0*) à A.
- C envoi (*Connect, 0*) à D.
- D envoi (*Connect, 0*) à C

2. BLOC 2 : réception de (*Connect, L*)

- Sur A :
 - Reçoit (*Connect, L=0*) de j=B
 - $L=0 == \text{Niv}=0$ alors :
 - $\text{canal}[j=B] = \text{Branch}$ alors :
 - Envoi (*initiate, 1, 1, find*) à B
- Sur B :
 - Reçoit (*Connect, L=0*) de j=A
 - $L=0 == \text{Niv}=0$ alors :
 - $\text{canal}[j=A] = \text{Branch}$ alors :
 - Envoi (*initiate, 1, 1, find*) à A
- Sur C :
 - Reçoit (*Connect, L=0*) de j=D
 - $L=0 == \text{Niv}=0$ alors :
 - $\text{canal}[j=D] = \text{Branch}$ alors :
 - Envoi (*initiate, 1, 2, find*) à D
- Sur D :
 - Reçoit (*Connect, L=0*) de j=C
 - $L=0 == \text{Niv}=0$ alors :
 - $\text{canal}[j=C] = \text{Branch}$ alors :
 - Envoi (*initiate, 1, 2, find*) à C

3. BLOC 3 : réception de (*initiate, L, F, S*) → Couleur

- Sur A :
 - Reçoit (*initiate, L=1, F=1, S=find*) de j=B
 - Attribution des nouvelles valeurs (cf. tableau)
 - Ne rentre pas dans la boucle pour tout
 - $\text{Etat} == \text{Find}$ alors Reçu=0 et appel à la procédure TEST.
- Sur B :
 - Reçoit (*initiate, L=1, F=1, S=find*) de j=A
 - Attribution des nouvelles valeurs (cf. tableau)
 - Ne rentre pas dans la boucle pour tout
 - $\text{Etat} = \text{Find}$ alors Reçu=0 et appel à la procédure TEST.
- Sur C :
 - Reçoit (*initiate, L=1, F=2, S=find*) de j=D
 - Attribution des nouvelles valeurs (cf. tableau)
 - Ne rentre pas dans la boucle pour tout
 - $\text{Etat} == \text{Find}$ alors Reçu=0 et appel à la procédure TEST.
- Sur D :
 - Reçoit (*initiate, L=1, F=2, S=find*) de j=C
 - Attribution des nouvelles valeurs (cf. tableau)

- Ne rentre pas dans la boucle pour tout
- Etat == Find alors Reçu=0 et appel à la procédure TEST.

4. BLOC 4 : procédure TEST → *Couleur*

- Appel par A :
 - Il existe :
 - C tel que canal[C]= basic
 Choix = C
 Changement de la valeur de Testcan.
 Envoi (*test*, *Niv=1*, *Nom=1*) à Testcan=C
- Appel par B :
 - Il existe :
 - C tel que canal[C]= basic
 - D tel que canal[D]= basic
 Choix = C
 Changement de la valeur de Testcan.
 Envoi (*test*, *Niv=1*, *Nom=1*) à Testcan=C
- Appel par C :
 - Il existe :
 - A tel que canal[A]= basic
 - B tel que canal[B]= basic
 Choix = B
 Changement de la valeur de Testcan.
 Envoi (*test*, *Niv=1*, *Nom=2*) à Testcan=B
- Appel par D :
 - Il existe :
 - B tel que canal[B]= basic
 Choix = B
 Changement de la valeur de Testcan.
 Envoi (*test*, *Niv=1*, *Nom=2*) à Testcan=B

5. BLOC 5 : réception de (*test*, *L*, *F*)

- Réception sur C du :
 - Message (*test*, *L=1*, *F=1*) venant de A :
 - $L=1 = \text{Niv}=1$ donc sinon
 - $F=1 \neq \text{nom}=2$ donc envoi (accept) à A
 - Message (*test*, *L=1*, *F=1*) venant de B :
 - $L=1 = \text{Niv}=1$ donc sinon
 - $F=1 \neq \text{nom}=2$ donc envoi (accept) à B
- Réception sur B du :
 - Message (*test*, *L=1*, *F=2*) venant de C :
 - $L=1 = \text{Niv}=1$ donc sinon
 - $F=2 \neq \text{nom}=1$ donc envoi (accept) à C
 - Message (*test*, *L=1*, *F=2*) venant de D :
 - $L=1 = \text{Niv}=1$ donc sinon
 - $F=2 \neq \text{nom}=1$ donc envoi (accept) à D

6. BLOC 6 : réception de (*accept*) → *Couleur*

- Réception sur A de C :
 - Réinitialisation de la variable Testcan
 - $\text{Poids}(A, C)=4 < \text{Mpoids}=\infty$ alors

- Changements des valeurs des variables (Cf. tableau)
 - Appel procédure REPORT
- Réception sur B de C :
 - Réinitialisation de la variable Testcan
 - $Poids(B, C)=3 < Mpoids=\infty$ alors
 - Changements des valeurs des variables (Cf. tableau)
 - Appel procédure REPORT
- Réception sur C de B :
 - Réinitialisation de la variable Testcan
 - $Poids(B, C)=3 < Mpoids=\infty$ alors
 - Changements des valeurs des variables (Cf. tableau)
 - Appel procédure REPORT
- Réception sur D de B :
 - Réinitialisation de la variable Testcan
 - $Poids(B, D)=5 < Mpoids=\infty$ alors
 - Changements des valeurs des variables (Cf. tableau)
 - Appel procédure REPORT

7. BLOC 8 : procédure REPORT → Couleur

- Appel par A :
 - Rentre dans la condition car C répond aux attentes
 - Changement de la valeur de la variable Etat
 - Envoi (*report, mpoids=4*) à père=B
- Appel par B :
 - Rentre dans la condition car C et D répondent aux attentes
 - Changement de la valeur de la variable Etat
 - Envoi (*report, mpoids=3*) à père=A
- Appel par C :
 - Rentre dans la condition car A et B répondent aux attentes
 - Changement de la valeur de la variable Etat
 - Envoi (*report, mpoids=3*) à père=D
- Appel par D :
 - Rentre dans la condition car B répond aux attentes
 - Changement de la valeur de la variable Etat
 - Envoi (*report, mpoids=5*) à père=C

8. BLOC 9 : réception de (*report, poids*)

- Réception sur A du message (*report, poids=3*) de B :
 - B père de A
 - Ne remplit aucune condition sort sans rien faire
- Réception sur B du message (*report, poids=4*) de A :
 - A père de B
 - $Poids=4 > Mpoids=3$ alors appel à la procédure CHANGERROOT
- Réception sur C du message (*report, poids=5*) de D :
 - D père de C
 - $Poids=5 > Mpoids=3$ alors appel à la procédure CHANGERROOT
- Réception sur D du message (*report, poids=3*) de C :
 - C père de D
 - Ne remplit aucune condition sort sans rien faire

9. BLOC 10 : procédure CHANGERROOT → Couleur

- Appel par B :
 - Canal[Mcan=C]=basic != branch alors
 - Envoi (*connect*, *Niv=1*) à C
 - Changement valeur de Canal[Mcan=C]
- Appel par C :
 - Canal[Mcan=B]=basic != branch alors
 - Envoi (*connect*, *Niv=1*) à B
 - Changement valeur de Canal[Mcan=B]

10. BLOC 2 : réception (*connect*, *L*)

- Sur B :
 - Réception du message (*connect*, *L=1*) de C
L=1 = Niv=1 donc on rentre dans le sinon
 Canal[C]=Branch != Basic, donc on rentre dans le sinon
 Envoi (*initiate*, *Niv+1=2*, poids(B ,C)=3, *find*) à C
- Sur C :
 - Réception du message (*connect*, *L=1*) de B
L=1 = Niv=1 donc on rentre dans le sinon
 Canal[B]=Branch != Basic, donc on rentre dans le sinon
 Envoi (*initiate*, *Niv+1=2*, poids(B ,C)=3, *find*) à B

11. BLOC 3 : réception de (*initiate*) → Couleur

- Sur B :
 - Réception du message (*initiate*, *L=2*, *F=3*, *S=find*) de C
 - Changement des valeurs de variables (Cf. tableau)
 - On rentre dans la boucle « pour ».
 On a Canal[A] et Canal[C] = Branch mais j=C donc on envoie seulement à A :
 - Envoi (*initiate*, *L=2*, *F=3*, *S=find*) à A
 - Etat=find alors :
 - Changement de la valeur de la variable reçu.
 - Appel procédure TEST.
- Sur C :
 - Réception du message (*initiate*, *L=2*, *F=3*, *S=find*) de B
 - Changement des valeurs de variables (Cf. tableau)
 - On rentre dans la boucle « pour ».
 On a Canal[B] et Canal[D] = Branch mais j=B donc on envoie seulement à D :
 - Envoi (*initiate*, *L=2*, *F=3*, *S=find*) à D
 - Etat=find alors :
 - Changement de la valeur de la variable reçu.
 - Appel procédure TEST.
 -

12. BLOC 3 : réception de (*initiate*) → Couleur

- Sur A :
 - Réception du message (*initiate*, *L=2*, *F=3*, *S=find*) de B
 - Changement des valeurs de variables (Cf. tableau)
 - On ne rentre pas dans la boucle « pour »
 - Etat=Find donc :
 - Changement de la valeur de la variable Reçu.
 - Appel procédure TEST.
- Sur D :
 - Réception du message (*initiate*, *L=2*, *F=3*, *S=find*) de C

- Changement des valeurs de variables (Cf. tableau)
- On ne rentre pas dans la boucle « pour »
- Etat=Find donc :
 - Changement de la valeur de la variable Reçu.
 - Appel procédure TEST.

13. BLOC 4 : procédure TEST → *Couleur*

- Appel par B :
 - Canal[D]=Basic → donc choix = D
 - Changement de la valeur de la variable Testcan.
 - Envoi (*test, Niv=2, Nom=3*) à D
- Appel par C :
 - Canal[A]=Basic → donc choix = A
 - Changement de la valeur de la variable Testcan.
 - Envoi (*test, Niv=2, Nom=3*) à A
- Appel par A :
 - Canal[C]=Basic → donc choix = C
 - Changement de la valeur de la variable Testcan.
 - Envoi (*test, Niv=2, Nom=3*) à C
- Appel par D :
 - Canal[B]=Basic → donc choix = B
 - Changement de la valeur de la variable Testcan.
 - Envoi (*test, Niv=2, Nom=3*) à B

14. BLOC 5 : réception de (*test*) → *Couleur*

- Sur A :
 - Réception du message (*test, L=2, F=3*) de C
 - L=2 = Niv=2 donc rentre dans le « sinon »
 - F=3 = Nom=3 donc rentre dans le « si »
 - Canal[C]=Basic donc rentre dans le « si » :
 - Changement de la valeur de Canal[C]
 - j=C = Testcan=C donc ne rentre pas dans le « si »
 - Appel procédure TEST
- Sur B :
 - Réception du message (*test, L=2, F=3*) de D
 - L=2 = Niv=2 donc rentre dans le « sinon »
 - F=3 = Nom=3 donc rentre dans le « si »
 - Canal[D]=Basic donc rentre dans le « si » :
 - Changement de la valeur de Canal[D]
 - j=D = Testcan=D donc on ne rentre pas dans le « si »
 - Appel procédure TEST
- Sur C :
 - Réception du message (*test, L=2, F=3*) de A
 - L=2 = Niv=2 donc rentre dans le « sinon »
 - F=3 = Nom=3 donc rentre dans le « si »
 - Canal[A]=Basic donc rentre dans le « si » :
 - Changement de la valeur de Canal[A]
 - j=A = Testcan=A donc ne rentre pas dans le « si »
 - Appel procédure TEST
- Sur D (message qu'on devait traiter plus tard) :

- Réception du message (*test*, $L=2$, $F=3$) de B
- $L=2 = \text{Niv}=2$ donc on rentre dans le « si »
- $F=3 = \text{Nom}=3$ donc rentre dans le « si »
- $\text{Canal}[B]=\text{Basic}$ donc rentre dans le « si » :
 - Changement de la valeur de $\text{Canal}[B]$
- $j=B = \text{Testcan}=B$ donc on ne rentre pas dans le « si »
- Appel procédure TEST

15. BLOC 4 : procédure TEST → *Couleur*

- Sur tous les sites :
 - Changement valeur Testcan.
 - Appel procédure Report

16. BLOC 8 : procédure REPORT → *Couleur*

- Sur A :
 - $\text{reçu} = 0 = \#\{\dots\}$ donc on rentre dans le « si »
 - On change la valeur de la variable Etat.
 - Envoi (*report*, $\text{mpoids} = \infty$) à père B
- Sur D :
 - $\text{reçu} = 0 = \#\{\dots\}$ donc on rentre dans le « si »
 - On change la valeur de la variable Etat.
 - Envoi (*report*, $\text{mpoids} = \infty$) à père C
- Sur B et C :
 - Ne rentre pas dans le « si ».

17. BLOC 9 : réception de (*report*) → *Couleur*

- Sur B :
 - Réception du message (*report*, $\text{poids} = \infty$) de A
 - $j=A \neq \text{père}=C$ donc on rentre dans le « si »
 - $\text{poids}=\infty == \text{mpoids}=\infty$ donc on ne rentre pas dans le « si »
 - Changer la valeur de Reçu
 - Appel procédure REPORT
- Sur C :
 - Réception du message (*report*, $\text{poids} = \infty$) de D
 - $j=D \neq \text{père}=B$ donc on rentre dans le « si »
 - $\text{poids}=\infty == \text{mpoids}=\infty$ donc on ne rentre pas dans le « si »
 - Changer la valeur de Reçu
 - Appel procédure REPORT

18. BLOC 8 : procédure REPORT → *Couleur*

- Sur B :
 - $\text{reçu} = 1 == \#\{\dots\}=1$ (A satisfait la condition) donc on rentre dans le « si »
 - On change la valeur de la variable Etat.
 - Envoi (*report*, $\text{mpoids} = \infty$) à père C
- Sur C :
 - $\text{reçu} = 1 == \#\{\dots\}=1$ (D satisfait la condition) donc on rentre dans le « si »
 - On change la valeur de la variable Etat.
 - Envoi (*report*, $\text{mpoids} = \infty$) à père B

19. BLOC 9 : réception de (*report*) → *Couleur*

- Sur B :
 - Réception du message (*report*, $poids = \infty$) de C
 - $j=C == père=C$ donc on ne rentre pas dans le « si »
 - $etat=found != find$ on rentre dans le « sinon »
 - $poids=\infty == mpoids=\infty$ donc on rentre dans le « sinon » puis dans le « si »
 - **TERMINE**
- Sur C :
 - Réception du message (*report*, $poids = \infty$) de B
 - $j=B == père=B$ donc on ne rentre pas dans le « si »
 - $etat=found != find$ on rentre dans le « sinon »
 - $poids=\infty == mpoids=\infty$ donc on rentre dans le « sinon » puis dans le « si »
 - **TERMINE**

On obtient graphiquement :

