Solutions de quelques exercices de la série N° 1

Exercice 8.

1. Soit G un *graphe simple biparti* d'ordre n, montrer que le nombre d'arêtes $m \leq n^2/4$.

G est biparti $\Rightarrow X = X_1 \cup X_2$ tel que $X_1 \cap X_2 = \varnothing$ et $\forall \{x,y\} \in E$, $(x \in X_1$ et $y \in X_2)$ ou $(x \in X_2$ et $y \in X_1)$. $\Rightarrow$

- $\forall x \in X_1$, $V(x) \subseteq X_2$… **(1)**
- $\forall y \in X_2$, $V(y) \subseteq X_1$… **(2)**

G est simple $\Rightarrow$ $\forall z \in X$, $d_G(z) = |V(z)|$… **(3)**

On pose $|X_1| = p$ et $|X_2| = q \Rightarrow p+q=n$… **(4)**

De (1) et (3) $\Rightarrow$ $\forall x \in X_1$, $d_G(x) \leq q$… **(5)**

De (2) et (3) $\Rightarrow$ $\forall y \in X_2$, $d_G(y) \leq p$… **(6)**

Calculons la somme des degrés des sommets de G :

$$\sum_{\forall z \in X} d_G(z) = \sum_{\forall x \in X_1} d_G(x) + \sum_{\forall y \in X_2} d_G(y) \text{ … } \mathbf{(7)}$$

De (5) $\Rightarrow \sum_{\forall x \in X_1} d_G(x) \leq q + q + \cdots + q = pq$

$\Rightarrow \sum_{\forall x \in X_1} d_G(x) \leq pq$ … **(8)**

De (6) $\Rightarrow \sum_{\forall y \in X_2} d_G(y) \leq p + p + \cdots + p = pq$

$\Rightarrow \sum_{\forall y \in X_2} d_G(y) \leq pq$ … **(9)**

De (7), (8) et (9) $\Rightarrow \sum_{\forall z \in X} d_G(z) \leq 2pq$ … **(10)**

On a $|E| = m$ et selon la formule des degrés, on a $\sum_{\forall z \in X} d_G(z) \leq 2m$ … **(11)**

De (4), (10) et (11) $\Rightarrow 2m \leq 2p(n-p) \Rightarrow m \leq p(n-p)$ … **(12)**

Par ailleurs, on sait que $(p-q)^2 \geq 0$ et de (4) $\Rightarrow (p-(n-p))^2 \geq 0$

$\Rightarrow (2p-n)^2 \geq 0 \Rightarrow 4p^2 - 4np - n^2 \geq 0 \Rightarrow 4p(n-p) \leq n^2$

$\Rightarrow p(n-p) \leq n^2/4$ … **(13)**

De (12) et (13) et par transitivité, nous avons : $m \leq n^2/4$

2. En déduire qu'il existe un sommet x tel que $d_G(x) \leq n/2$.

Si $\forall x \in X$, $d_G(x) > n/2 \Rightarrow \sum_{\forall x \in X} d_G(x) > n.\left(\frac{n}{2}\right) \Rightarrow 2m > n.\left(\frac{n}{2}\right) \Rightarrow 2m > n^2/4$. Ce qui est en contradiction avec ce que nous avons démontré dans la question 1.

$\Rightarrow \forall x \in X$, $d_G(x) \leq n/2$.

3. Montrer qu'un graphe régulier d'ordre impair ne peut être biparti.

Pour éviter la répétions, nous utilisons toutes les définitions, notations et formules (particulièrement 4 et 7) liées aux graphes bipartis vues dans la question 1.

Démontrons par l'absurde : On suppose qu'on a un graphe régulier d'ordre impair et biparti.

Nous avons : $\sum_{\forall z \in X} d_G(z) = \sum_{\forall x \in X_1} d_G(x) + \sum_{\forall y \in X_2} d_G(y)$

Et : G est régulier $\Rightarrow \forall x \in X$, $d_G(x) = k$.

$\Rightarrow \sum_{\forall z \in X} d_G(z) = kp + kq$

Par ailleurs, nous savons que dans un graphe biparti, toute arête a une extrémité dans X_1 et une extrémité dans $X_2 \Rightarrow \sum_{\forall x \in X_1} d_G(x) = \sum_{\forall y \in X_2} d_G(y) \Rightarrow kp = kq \Rightarrow p = q$

$n=p+q$ et $n=2l+1$ (impair) et $p=q \Rightarrow n = 2p = 2q = 2l+1 \Rightarrow p=q=l+1/2 \Rightarrow$ p et q ne sont pas des entiers. Or, le nombre d'éléments dans un ensemble (ici nombre de sommets) est toujours entier positif ou nul $\Rightarrow$ Contradiction $\Rightarrow$ Cqfd.

Exercice 10.

On veut organiser un examen comportant, outre les matières communes, 7 matières d'options : Langue (L), Electronique (E), Mécanique (M), Dessin industriel (D), Informatique (I), Génie Civil (G), Sport (S). Les profils des candidats à options multiples sont : L,E,M - M,D,S - L,S - I,L,E - D,G

Avant de répondre aux questions, il faut procéder à la modélisation.

On modélise le problème sous forme d'un graphe non orienté $G=(X, E)$.

Chaque sommet $x \in X$ représente une matière x. En d'autres termes $X = \{D, E, G, I, L, M, S\}$

Chaque arête $\{x, y\} \in E$ représente la relation « Les matières x et y ne peuvent pas être mises en parallèle ». C'est-à-dire « elles ont des étudiants en commun ».

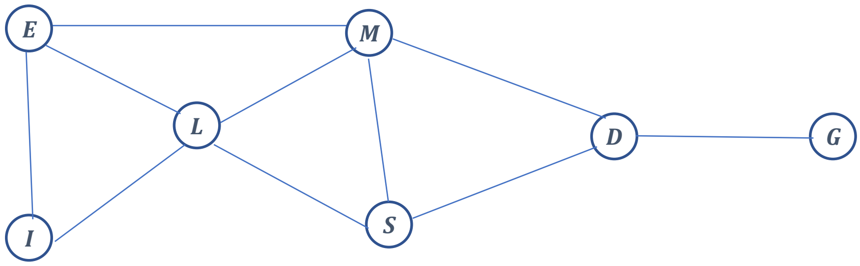
1. Quel est le nombre maximum d'épreuves qu'on peut mettre en parallèle ?

Identification du problème :

Nombre maximal de matières qu'on eut mettre en parallèle :

- 2 matières en parallèle $\Rightarrow$ 2 sommets non reliés $\Rightarrow$ Stable de 2 éléments.
- k matières en parallèle $\Rightarrow$ k sommets non reliés $\Rightarrow$ Stable de k éléments.

Donc, la solution revient à chercher le plus grand stable dans le graphe G.
Dessinons le graphe et cherchons le plus grand stable :



Soit $S_1=\{I, S, G\}$ un stable de 3 éléments.
C'est le plus grand stable (Mais pas le seul, on a par exemple : $S_2=\{I, M, G\}$ ou $S_3=\{E, S, G\}$).
Donc, on peut mettre en parallèle au maximum 3 matières.

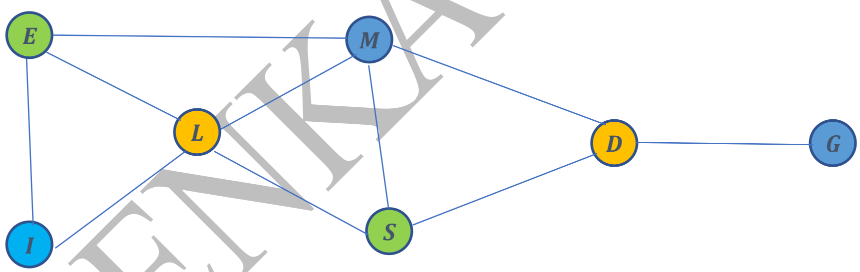
2. Une épreuve occupe une demi-journée ; quel est le temps minimal nécessaire pour ces options ?

La durée minimale des examens :

2 créneaux $\Rightarrow$ 2 matières qui ont des candidats en commun $\Rightarrow$ 2 sommets reliés $\Rightarrow$ 2 couleurs minimum.
k créneaux $\Rightarrow$ k matières qui ont des candidats en commun $\Rightarrow$ k sommets complètement reliés (2 à 2) $\Rightarrow$ k couleurs minimum.

Donc, la solution revient à chercher $\chi(G)$ le nombre chromatique de G.
Appliquons l'algorithme de *Welsh & Powell* pour effectuer une k-coloration des sommets de G.

Sommet $x \in X$	$d_G(x)$	**Couleur** $\varphi(x)$
L	4	1
M	4	2
D	3	1
E	3	3
S	3	3
I	2	2
G	1	2



Nous avons obtenu une 3-coloration. Est-ce que 3 est le nombre chromatique ?
Vérification :
On a $|C| \leq \chi(G) \leq \Delta(G)+1$ où C est la plus grande clique dans le graphe G.
$\Rightarrow 3 \leq \chi(G) \leq 5$
$\Rightarrow$ 3 est le minimum et on a obtenu 3 couleurs $\Rightarrow$ 3 est le nombre chromatique.
La durée est 3 demi-journées (1,5 journées).

Exercice 12

Montrez que dans un groupe de six (6) personnes, il y en a nécessairement trois (3) qui se connaissent mutuellement ou trois (3) qui ne se connaissent pas (on suppose que si A connaît B, B connaît également A).

On modélise le problème par un graphe non orienté $G=(X, E)$.
Chaque personne i sera représentée par un sommet $i \in X$.
Les arêtes vont représenter la relation « se connaissent ».
$\{i, j\} \in E$ correspond à : les personnes i et j « se connaissent ».
Nous pouvons reformuler le problème sous la forme suivante : Il faut montrer que le graphe G contient une clique de 3 éléments (3 qui se connaissent mutuellement) ou un stable de 3 éléments (3 qui ne se connaissent pas).
Le graphe G est simple car :

- Une boucle $\{i, i\}$ correspond à la relation i et i se connaissent qui n'a pas de sens $\Rightarrow$ Pas de boucles dans G.
- Si i et j se connaissent, ça sera représenté par une seule arête $\Rightarrow$ Pas d'arêtes parallèles dans G.

G simple $\Rightarrow 0 \leq d_G(x) \leq 5$
Pour un sommet quelconque, nous avons deux cas :

- Soit $d_G(x) \in \{0, 1, 2\}$
- Soit $d_G(x) \in \{3, 4, 5\}$

a) 1er cas : $d_G(x) \in \{0, 1, 2\}$:
Il y a au moins 3 sommets qui ne sont pas reliés avec *x*.
On a 2 possibilités :

- Soit ces 3 sommets sont reliés complètement entre eux et ils forment une clique de 3 éléments.
- Soit il y a au moins 2 sommets non reliés entre eux et ils forment avec *x* un stable de 3 éléments.

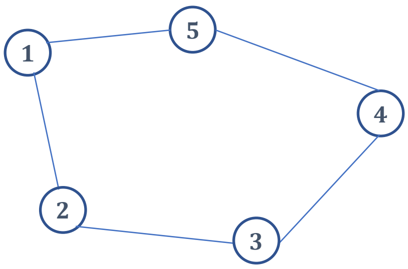
b) 2ème cas : $d_G(x) \in \{3, 4, 5\}$:
Il y a au moins 3 sommets qui sont reliés à *x*.
On a 2 possibilités :

- Soit ces 3 sommets ne sont pas du tout reliés entre eux et ils forment une stable de 3 éléments.
- Soit il y a au moins 2 sommets reliés entre eux et ils forment avec *x* une clique de 3 éléments.

Donc, dans toutes les situations, il y a 3 personnes qui se connaissent mutuellement ou 3 personnes qui ne se connaissent pas.

Cela est-il nécessairement vrai dans un groupe de cinq (5) personnes ?

Pour 5 sommets, ça ne marche pas toujours.
Prenons le contre-exemple suivant. Soit l'exemple d'un graphe simple d'ordre 5 qui est 2-régulier comme le montre le dessin ci-dessous :



La plus grande clique : 2 éléments.
Le plus grand stable : 2 éléments.

Exercice 13

Soit $G=(X, E)$ un graphe simple, d'ordre n. Si G est k-régulier, dans quelles conditions il est isomorphe à son complémentaire ?

Si G et $\bar{G}$ sont isomorphes alors :

- $G = (X, E)$ non orienté simple, d'ordre n, k-régulier.
- $\bar{G} = (X, E')$ non orienté simple, d'ordre n, k-régulier.

$\Rightarrow \forall x \in X$, $d_G(x) = k$ et $d_{\bar{G}}(x) = k$ … (1)
Vu que G est simple, on sait que chaque sommet x dans le complément est relié à tous les sommets (n) sauf lui-même (-1) et les sommets auxquels il était relié dans G ($-d_G(x)$).
$\Rightarrow d_{\bar{G}}(x) = n - d_G(x) - 1$ … (2)
De (1) et (2) $\Rightarrow k=n-1-k$.
$\Rightarrow n=2k+1$, c'est-à-dire le graphe doit être d'ordre impair
$\Rightarrow k=(n-1)/2$, c'est-à-dire les degrés des sommets doivent être égaux à la partie entière de la moitié de n.

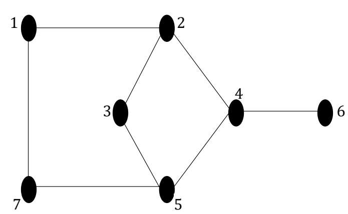
Chapitre 2 : Cheminement dans les Graphes. Série d'exercices de TD.

Présenté par :
H. BENKAOUHA
Bureau 222, Faculté d'Informatique, USTHB
haroun.benkaouha@usthb.edu.dz
haroun.benkaouha@gmail.com

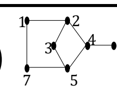
Exercice 1

- Soit $G = (X, E)$ un graphe non orienté, simple et connexe sur $\boldsymbol{n}$ sommets.
  - On note la **longueur** d'une chaîne $\boldsymbol{\mu(x, y)}$ joignant x et y, $|\boldsymbol{\mu(x, y)}|$.
  - L'**écart** entre x et y : $\boldsymbol{e(x, y)}$ est la longueur d'une plus courte chaîne joignant x et y :
    $e(x, y) = \min\limits_{\mu(x,y) \in G} \{|\mu(x,y)|\}$;
    $e(x, x) = 0.$
  - **Ecartement** d'un sommet x, le nombre $E(x) = \max\limits_{y \in X} \{ e(x, y) \}$
  - **Diamètre** de G, le nombre $e(G) = \max\limits_{x, y \in X} \{ e(x, y) \}$
  - **Rayon** de G, le nombre $r(G) = \min\limits_{x \in X} \{E(x)\}$
  - **Centre** de G, un sommet $s \in X$ tel que : $E(s) = r(G)$
- Déterminer le diamètre, le rayon et le ou les centres du graphe suivant.

Exercice 1 - Suite



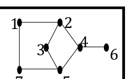
Exercice 1 – Solution (1/3)



- L'écart entre toute paire de sommets :

	1	2	3	4	5	6	7
1	0						
2		0					
3			0				
4				0			
5					0		
6						0	
7							0

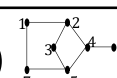
Exercice 1 – Solution (1/3)



- L'écart entre toute paire de sommets :

	1	2	3	4	5	6	7
1	0	1					1
2	1	0	1	1			
3		1	0		1		
4		1		0	1	1	
5			1	1	0		1
6				1		0	
7	1				1		0

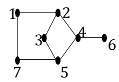
Exercice 1 – Solution (1/3)



- L'écart entre toute paire de sommets :

	1	2	3	4	5	6	7
1	0	1	2	2	2		1
2	1	0	1	1			
3	2	1	0		1		
4	2	1		0	1	1	
5	2		1	1	0		1
6				1		0	
7	1				1		0

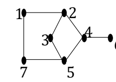
Exercice 1 – Solution (1/3)



- L'écart entre toute paire de sommets :

	1	2	3	4	5	6	7
1	0	1	2	2	2		1
2	1	0	1	1	2	2	2
3	2	1	0		1		
4	2	1		0	1	1	
5	2	2	1	1	0		1
6		2		1		0	
7	1	2			1		0

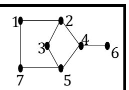
Exercice 1 – Solution (1/3)



- L'écart entre toute paire de sommets :

	1	2	3	4	5	6	7
1	0	1	2	2	2		1
2	1	0	1	1	2	2	2
3	2	1	0	2	1		2
4	2	1	2	0	1	1	
5	2	2	1	1	0		1
6		2		1		0	
7	1	2	2		1		0

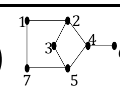
Exercice 1 – Solution (1/3)



- L'écart entre toute paire de sommets :

	1	2	3	4	5	6	7
1	0	1	2	2	2		1
2	1	0	1	1	2	2	2
3	2	1	0	2	1		2
4	2	1	2	0	1	1	2
5	2	2	1	1	0		1
6		2		1		0	
7	1	2	2	2	1		0

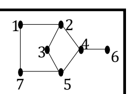
Exercice 1 – Solution (1/3)



- L'écart entre toute paire de sommets :

	1	2	3	4	5	6	7
1	0	1	2	2	2		1
2	1	0	1	1	2	2	2
3	2	1	0	2	1		2
4	2	1	2	0	1	1	2
5	2	2	1	1	0	2	1
6		2		1	2	0	
7	1	2	2	2	1		0

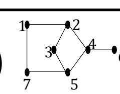
Exercice 1 – Solution (1/3)



- L'écart entre toute paire de sommets :

	1	2	3	4	5	6	7
1	0	1	2	2	2	3	1
2	1	0	1	1	2	2	2
3	2	1	0	2	1		2
4	2	1	2	0	1	1	2
5	2	2	1	1	0	2	1
6	3	2		1	2	0	
7	1	2	2	2	1		0

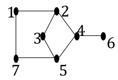
Exercice 1 – Solution (1/3)



- L'écart entre toute paire de sommets :

	1	2	3	4	5	6	7
1	0	1	2	2	2	3	1
2	1	0	1	1	2	2	2
3	2	1	0	2	1	3	2
4	2	1	2	0	1	1	2
5	2	2	1	1	0	2	1
6	3	2	3	1	2	0	
7	1	2	2	2	1		0

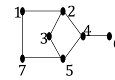
Exercice 1 – Solution (1/3)



- L'écart entre toute paire de sommets :

	1	2	3	4	5	6	7
1	0	1	2	2	2	3	1
2	1	0	1	1	2	2	2
3	2	1	0	2	1	3	2
4	2	1	2	0	1	1	2
5	2	2	1	1	0	2	1
6	3	2	3	1	2	0	3
7	1	2	2	2	1	3	0

Exercice 1 – Solution (2/3)

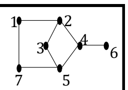


- L'écartement d'un sommet : max ligne ou colonne

	1	2	3	4	5	6	7
1	0	1	2	2	2	3	1
2	1	0	1	1	2	2	2
3	2	1	0	2	1	3	2
4	2	1	2	0	1	1	2
5	2	2	1	1	0	2	1
6	3	2	3	1	2	0	3
7	1	2	2	2	1	3	0

$E(x)$:	3	2	3	2	2	3	3

Exercice 1 – Solution (3/3)



$E(x)$:	3	2	3	2	2	3	3

- Le diamètre

  $e(G) = 3$
- Le rayon

  $r(G) = 2$
- Les centres

  2, 4 et 5

Exercice 2

- Dans un réseau téléphonique constitué de $2n$ centraux téléphoniques
- Disposés de telle façon que chaque central est relié par une ligne téléphonique directe avec au moins n autres centraux.
- Montrez qu'il est toujours possible d'établir une liaison entre deux centraux quelconques.

Exercice 2 – solution (1/7)

- Modélisation : Par un Graphe $G=(X, E)$
  - Chaque central téléphonique i représentée par un sommet i
  - Il y a $2n$ centraux téléphoniques $\Rightarrow$ Il y a $2n$ sommets $\Rightarrow$ Graphe d'ordre $2n$
  - Une arête $\{i,j\}$ « *Les centraux i et j sont reliés par une ligne directe* »
  - Chaque central est relié par une ligne téléphonique directe avec au moins n autres centraux
  - $\Rightarrow \forall x \in X, d_G(x) \geq n$

Exercice 2 - solution (2/7)

- Identification du problème
  - Toujours possible d'établir une liaison entre deux centraux quelconques.
  - $\Rightarrow$Il y a une chaîne reliant les sommets correspondants
  - $\Rightarrow$ G doit être connexe

Exercice 2 - solution (3/7)

- Identification du problème
  - Revient à montrer que
  - Si un graphe $G=(X, E)$ d'ordre $2n$ tel que $\forall i \in X$, $d_G(x) \geq n$
  - Alors G est connexe

Exercice 2 - solution (4/7)

- On démontre par l'absurde
  - On suppose qu'il existe un graphe $G=(X, E)$ d'ordre $2n$ tel que $\forall i \in X$, $d_G(i) \geq n$ et G n'est pas connexe.

Exercice 2 - solution (5/7)

$\Rightarrow$Il y a au moins 2 *CC* dans G

$\Rightarrow X=C_1 \cup C_2 \cup \ldots \cup C_k$ $(k \geq 2)$ et $\forall p \neq q$ $C_p \cap C_q = \varnothing$ où chaque C_l ($\forall 1 \leq l \leq k$) est une *CC* dans G

$\forall x \in X$, $d_G(x) \geq n$ *et* G *simple*

$\Rightarrow$ dans une *CC* (C_l $\forall 1 \leq l \leq k$), nous avons au moins $n+1$ sommets, c'est-à-dire le sommet x et tous ses voisins.

Exercice 2 - solution (6/7)

$\Rightarrow \forall$ $1 \leq l \leq k$, $|C_l| \geq n+1$

$\Rightarrow |C_1| + |C_2| + \ldots + |C_k| \geq k(n+1)$

Sachant que $k \geq 2$ alors $k(n+1) \geq 2(n+1) = 2n+2 > 2n$

$\Rightarrow |C_1| + |C_2| + \ldots + |C_k| > 2n = |X|$

Exercice 2 - solution (7/7)

Or, nous avons :

$\Rightarrow \forall$ $1 \leq l \leq k$, $|C_l| \leq |X|$

et $|C_1| + |C_2| + \ldots + |C_k| = |X| = 2n$

Car $X=C_1 \cup C_2 \cup \ldots \cup C_k$ $(k \geq 2)$ et $\forall p \neq q$ $C_p \cap C_q = \varnothing$

- *Contradiction*

Exercice 3

- Soit $G=(X, E)$ un graphe connexe d'ordre $n \geq 2$.
  - Montrons qu'il existe un sommet x
  - tel que :
  - le sous-graphe de G engendré par le sous ensemble de sommets X-{x} est toujours connexe.

Exercice 3 – Solution (1/3)

- On démontre par l'absurde.
- On suppose qu'il existe un graphe $G=(X, E)$ d'ordre $n \geq 2$ mais $\forall\, x \in X$, G-$\{x\}$ est non connexe.
- En d'autre termes, la suppression de n'importe quel sommet va déconnecter le graphe.
- Soit $\mu = x_0 e_1 x_1 \ldots e_p x_p$
  - La plus longue chaîne dans G.
  - $|\mu| = p$ tel que $p \leq n$.

Exercice 3 – Solution (2/3)

- Selon l'hypothèse de l'absurde, la suppression de n'importe quel sommet va déconnecter le graphe.
- Prenons le sommet x_0.
- $G' = G$-$\{x_0\}$ est non connexe.
- $\mu' = x_1 e_2 x_2 \ldots e_p x_p$ est une chaîne dans G'.

$\Rightarrow C = \{x_1, x_2, \ldots, x_p\}$ font partie de la même CC.

Exercice 3 – Solution (3/3)

$\Rightarrow \exists y \in X$-$\{x_0\}$ et $y \notin C$ qui n'est pas dans la même CC que les sommets de C.

$\Rightarrow \exists\, e = \{y, x_0\} \in E$ tel que la suppression de x_0 a engendré la suppression de cette arête dans G'.

$\Rightarrow \mu'' = y\, e x_0 e_1 x_1 \ldots e_p x_p$ est une chaîne dans G de longueur $p+1$.

$\Rightarrow |\mu''| > |\mu|$

$\Rightarrow \mu$ n'est pas la plus grande chaîne dans G.

$\Rightarrow$ *Contradiction*

Exercice 3 – 2ème Solution(1/4)

- On démontre par récurrence sur l'ordre du graphe n.
- Cas de base :
  - $n=2$: On a 2 sommets reliés entre eux car G est connexe
  - On supprime l'un d'eux, on obtient un graphe avec un seul sommet
  - Ce genre de graphe est considéré comme connexe
  - C'est vérifié

Exercice 3 - 2ème Solution(2/4)

- On démontre par récurrence sur l'ordre du graphe n.
- Hypothèse de récurrence :
  - On suppose que pour un graphe d'ordre $n \leq p$ connexe, il existe un sommet x que si on le supprime le graphe reste connexe.
- Pas de récurrence
  - Démontrons pour $n=p+1$

Exercice 3 - 2ème Solution(3/4)

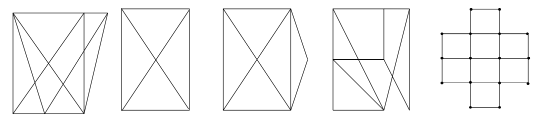
- Il s'agit d'un graphe d'ordre p auquel on a rajouté un sommet.
- On a 2 cas, soit le graphe d'ordre p est connexe, soit le graphe d'ordre p n'est pas connexe.
- Si le graphe d'ordre p est connexe, il suffit de supprimer le sommet qu'on a rajouté.

Exercice 3 - 2ème Solution(4/4)

– Si le graphe d'ordre p n'est pas connexe, alors le sommet qui a été rajouté permet de relier les différentes CC
– Dans ce cas, chaque sous graphe engendré par une CC vérifie l'hypothèse de récurrence car il est connexe et d'ordre $<p$
– Donc il existe un sommet qu'on eut supprimer sans déconnecter le graphe.

Exercice 4

- *Est-il possible de tracer les figures suivantes sans lever le crayon (et sans passer deux fois sur le même trait !...) ? Pourquoi ?*



Exercice 4 – Solution (1/10)

- Modélisation
  – On représente chacune des 5 figures par un graphe $G_i=(X_i, E_i)$ tel que i=1 à 5
  – Chaque point extrémité d'un trait est représenté par un sommet.
  – Chaque trait ou segment de trait reliant 2 points d'extrémités est représenté par une arête.

Exercice 4 - Solution (2/10)

- Identification du problème
  – Tracer une figure sans lever le crayon : parcourir tout le graphe (toutes les arêtes) en passant d'une arête à une autre qui lui est adjacente.
  – C'est-à-dire tracer une chaîne qui passe par toutes les arêtes du graphe.

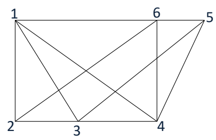
Exercice 4 - Solution (3/10)

- Identification du problème
  – Sans passer deux fois sur le même trait $\Rightarrow$ Ne pas passer par la même arête plus d'une fois $\Rightarrow$ Chaîne simple
  – Chaîne simple qui passe par toutes les arêtes $\Rightarrow$ Chaîne Eulérienne
  – Le problème revient à vérifier pour chacun des graphes G_i s'il admet une chaîne Eulérienne.

Exercice 4 - Solution (4/10)

- Résolution
  – Selon le théorème d'Euler, G_i doit être connexe (à des sommets isolés près) et doit avoir 0 ou 2 sommets de degrés impairs.
  – Tous les graphes sont connexes.
  – La solution revient à vérifier la parité des degrés des sommets de chacun des graphes.

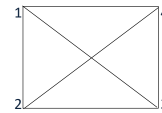
Exercice 4 - Solution (5/10)



- $d_G(1)$, $d_G(3)$, $d_G(4)$, $d_G(6)$ sont pairs et $d_G(2)$, $d_G(5)$ sont impairs

$\Rightarrow$ 2 sommets de degrés impairs $\Rightarrow$ Possible

Exercice 4 - Solution (6/10)

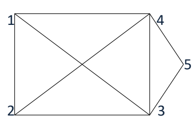


- $d_G(1)$, $d_G(2)$, $d_G(3)$, $d_G(4)$ sont impairs

$\Rightarrow$Plus de 2 sommets de degrés impairs

$\Rightarrow$ Impossible

Exercice 4 - Solution (7/10)

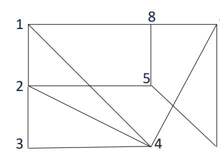


- $d_G(3)$, $d_G(4)$, $d_G(5)$ sont pairs et $d_G(1)$, $d_G(2)$ sont impairs

$\Rightarrow$2 sommets de degrés impairs

$\Rightarrow$ Possible

Exercice 4 - Solution (8/10)

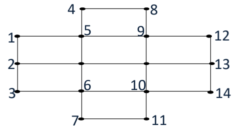


- $d_G(1)$, $d_G(5)$, $d_G(7)$, $d_G(8)$ sont impairs

$\Rightarrow$Plus de 2 sommets de degrés impairs

$\Rightarrow$ Impossible

Exercice 4 - Solution (9/10)



$\Rightarrow$2 sommets de degrés impairs (2 et 13)

$\Rightarrow$ Possible

- **Remarque 1. :** Au moins un sommet parmi les sommets 5, 6, 9 et 10 doit être pris en considération, sinon le graphe ne sera pas connexe.

Exercice 4 - Solution (10/10)

- **Remarque 2. :**
  - Les points d'intersection entre deux ou plusieurs traits, peut-on les prendre comme sommets? Si oui, qu'est ce qui va changer?
  - Dans le G_5, au moins un est obligatoire. (voir remarque 1)
  - En général c'est facultatif : On peut les prendre comme sommets.
  - Ça ne change rien à ce problème, car leurs degrés seront pairs.

Exercice 5

- Soit *G* un graphe non eulérien.
- Est-il toujours possible de rendre *G* eulérien en lui rajoutant une ou plusieurs arêtes ?

Exercice 5 – Solution (1/3)

- Oui, il est possible de rendre *G* eulérien en lui rajoutant une ou plusieurs arêtes comme suit :
  - **Etape 1 :** S'assurer que le graphe est connexe
  - **Etape 2 :** S'assurer que tous les degrés deviennent pairs car un graphe est Eulérien Ssi il admet un cycle Eulérien.

Exercice 5 – Solution (2/3)

- **Etape 1 :**
  - Toutes les composantes connexes (CC) on les relie avec une arête.
  - C'est-à-dire si on a *p* CC, on va ajouter $p - 1$ arêtes.
- **Etape 2 :**
  - Pour toute paire de sommets de degrés impairs on les relie avec une arête.
  - C'est-à-dire si on a *k* (qui est pair) sommets de degrés impairs, on va ajouter $k/2$ arêtes.

Exercice 5 – Solution (3/3)

```
Pour k de 2 à p
  Faire
    x ← choisir_sommet(CC_{k-1})
    y ← choisir_sommet(CC_k)
    Créer_arête({x,y})
  Fait
i← 1
Répéter
    Tant Que (i<=N) et (d_G(i) pair)
      Faire    i← i+1;    Fait
    Si (i<N) Alors
        x ← i;      i← i+1
        Tant Que (i<=N) et (d_G(i) pair)
          Faire    i← i+1;    Fait
        y ← i;      i← i+1
        Créer_arête({x,y})
    fSi
Jusqu'à (i=N);
```


Exercice 6

- Soit le graphe orienté *G=(X,U)* représenté dans le tableau ci-dessous par le dictionnaire des prédécesseurs :

x	1	2	3	4	5	6	7	8
Prédécesseurs de *x*	3, 7	4, 6	5	1	1	7, 8	5	2

1. Donner la matrice d'adjacence *M* du graphe *G*. Représenter sous forme de listes *LS* et *PS*.
2. *G* est-il connexe. Justifier.
3. *G* admet-il un parcours Eulerien ? Pourquoi ?
4. Donner la matrice de fermeture transitive du graphe *G*. *G* admet-il un circuit ?
5. Trouver les cfc de *G*. Donner le graphe réduit.

Exercice 6 – Solution (1/11)

- Matrice d'adjacence :

	1	2	3	4	5	6	7	8
1	0	0	0	1	1	0	0	0
2	0	0	0	0	0	0	0	1
3	1	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0
5	0	0	1	0	0	0	1	0
6	0	1	0	0	0	0	0	0
7	1	0	0	0	0	1	0	0
8	0	0	0	0	0	1	0	0

Remarque : Attention, liste prédécesseurs et non successeurs

Enseignant : Dr. H. BENKAOUHA (haroun.benkaouha@usthb.edu.dz)

Exercice 6 - Solution (2/11)

- Représentation par listes :

	1	2	3	4	5	6	7	8	9
PS	1	3	4	5	6	8	9	11	12

	1	2	3	4	5	6	7	8	9	10	11
LS	4	5	8	1	2	3	7	2	1	6	6

Exercice 6 - Solution (3/11)

- Connexité :
  - Algorithme de connexité :

  Sommet de départ : 1

  C={1}

  On rajoute les voisins de 1

  *V(1)={*3, 4, 5, 7*}*

  *C={*1, 3, 4, 5, 7*}*. On marque le sommet 1.

Exercice 6 – Solution (4/11)

*C={*1, 3, 4, 5, 7*}*

On choisit un sommet non marqué de *C* : 7

On rajoute les voisins de 7

*V(7)={*1, 5, 6*}*

*C={*1, 3, 4, 5, 6, 7*}*. On marque le sommet 7.

Exercice 6 - Solution (5/11)

*C={*1, 3, 4, 5, 6, 7*}*

On choisit un sommet non marqué de *C* : 6

On rajoute les voisins de 6

V(6)={2, 7, 8*}*

*C={*1, 2, 3, 4, 5, 6, 7 , 8*}*. On marque le sommet 6.

C=X $\Rightarrow$ Fin Algo.

G est connexe.

Exercice 6 - Solution (6/11)

- On vérifie si G admet un parcours Eulérien :
  - Calculons les degrés :

	1	2	3	4	5	6	7	8	d_G^+
1	0	0	0	1	1	0	0	0	2
2	0	0	0	0	0	0	0	1	1
3	1	0	0	0	0	0	0	0	1
4	0	1	0	0	0	0	0	0	1
5	0	0	1	0	0	0	1	0	2
6	0	1	0	0	0	0	0	0	1
7	1	0	0	0	0	1	0	0	2
8	0	0	0	0	0	1	0	0	1
d_G^-	2	2	1	1	1	2	1	1	

Exercice 6 - Solution (7/11)

- $d_G(2)=1+2=3$
- $d_G(5)=2+1=3$
- $d_G(6)=1+2=3$
- Plus de deux (2) sommets de degrés impairs
- $\Rightarrow$ *G* n'admet pas de chaine Eulérienne
- $\Rightarrow$ *G* n'admet aucun parcours Eulérien

Exercice 6 - Solution (8/11)

- Matrice de fermeture transitive :
  - Algorithme de Warshall

	1	2	3	4	5	6	7	8
1				1	1			
2								1
3	1							
4		1						
5			1				1	
6		1						
7	1					1		
8						1		

Exercice 6 – Solution (8/11)

- Matrice de fermeture transitive :
  - Algorithme de Warshall

	1	2	3	4	5	6	7	8
1				1	1			
2								1
3	1			1	1			
4		1						
5			1				1	
6		1						
7	1			1	1	1		
8						1		

Exercice 6 - Solution (8/11)

- Matrice de fermeture transitive :
  - Algorithme de Warshall

	1	2	3	4	5	6	7	8
1				1	1			
2								1
3	1			1	1			
4		1						1
5			1				1	
6		1						1
7	1			1	1	1		
8						1		

Exercice 6 – Solution (8/11)

- Matrice de fermeture transitive :
  - Algorithme de Warshall

	1	2	3	4	5	6	7	8
1				1	1			
2								1
3	1			1	1			
4		1						1
5	1		1	1	1		1	
6		1						1
7	1			1	1	1		
8						1		

Exercice 6 - Solution (8/11)

- Matrice de fermeture transitive :
  - Algorithme de Warshall

	1	2	3	4	5	6	7	8
1		1		1	1			1
2								1
3	1	1		1	1			1
4		1						1
5	1	1	1	1	1		1	1
6		1						1
7	1	1		1	1	1		1
8						1		

Exercice 6 – Solution (8/11)

- Matrice de fermeture transitive :
  - Algorithme de Warshall

	1	2	3	4	5	6	7	8
1	1	1	1	1	1		1	1
2								1
3	1	1	1	1	1		1	1
4		1						1
5	1	1	1	1	1		1	1
6		1						1
7	1	1	1	1	1	1	1	1
8						1		

Exercice 6 – Solution (8/11)

- Matrice de fermeture transitive :
  - Algorithme de Warshall

	1	2	3	4	5	6	7	8
1	1	1	1	1	1		1	1
2								1
3	1	1	1	1	1		1	1
4		1						1
5	1	1	1	1	1		1	1
6		1						1
7	1	1	1	1	1	1	1	1
8		1				1		1

Exercice 6 – Solution (8/11)

- Matrice de fermeture transitive :
  - Algorithme de Warshall

	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2								1
3	1	1	1	1	1	1	1	1
4		1						1
5	1	1	1	1	1	1	1	1
6		1						1
7	1	1	1	1	1	1	1	1
8		1				1		1

Exercice 6 – Solution (8/11)

- Matrice de fermeture transitive :
  - Algorithme de Warshall

	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2		1				1		1
3	1	1	1	1	1	1	1	1
4		1				1		1
5	1	1	1	1	1	1	1	1
6		1				1		1
7	1	1	1	1	1	1	1	1
8		1				1		1

Exercice 6 – Solution (9/11)

- Matrice de fermeture transitive :
  - Algorithme de Warshall

	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	0	1	0	0	0	1	0	1
3	1	1	1	1	1	1	1	1
4	0	1	0	0	0	1	0	1
5	1	1	1	1	1	1	1	1
6	0	1	0	0	0	1	0	1
7	1	1	1	1	1	1	1	1
8	0	1	0	0	0	1	0	1

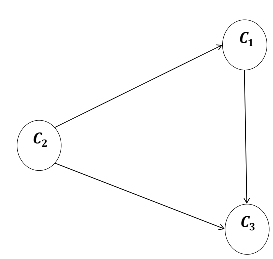
  - Oui, G admet un circuit car il y a des 1 sur la diagonale.

Exercice 6 – Solution (10/11)

- Les CFCs :
  - C_1={4} car 0 sur la diagonale
  - Lignes identiques (sauf 4) :
    - Lg_1={1, 3, 5, 7}
    - Lg_2={2, 6, 8}
  - Colonnes identiques (sauf 4) :
    - Cl_1={2, 6, 8}
    - Cl_2={1, 3, 5, 7}
    - $\Rightarrow$
  - C_2={1, 3, 5, 7} et C_3={2, 6, 8}

Exercice 6 – Solution (11/11)

- Le graphe réduit :



Exercice 7

- On considère des dominos dont les faces sont numérotées 1, 2, 3, 4 ou 5.

1. En excluant les dominos doubles, de combien de dominos dispose-t-on ?
2. Montrez que l'on peut arranger ces dominos de façon à former une boucle fermée (en utilisant la règle habituelle de contact entre les dominos).
3. Pourquoi n'est-il pas nécessaire de considérer les dominos doubles ?
4. Si l'on prend maintenant des dominos dont les faces sont numérotées de 1 à n, est-il possible de les arranger de façon à former une boucle fermée ?

Exercice 7 - Solution (1/7)

- Modélisation :
  - Graphe non orienté $G=(X, E)$ d'ordre n et de taille m
  - Chaque face (chaque numéro) i par un sommet i
  - Chaque domino (pièce) qui est constitué de deux faces (i et j) par une arête $\{i, j\}$
  - $n=5$

Exercice 7 - Solution (2/7)

1. Nombre de dominos
   - Correspond au nombre d'arêtes dans le graphe
   - Pas de dominos doubles ⇒Pas de boucles
   - Chaque pièce est unique (combinaison unique de deux numéros) ⇒Pas d'arêtes parallèles

   ⇒ G est <u>simple</u>
   - Dans un jeu de dominos, nous avons toutes les combinaisons de faces possibles
   - Tous les sommets sont reliés entre eux

   ⇒ G est <u>complet</u>

Exercice 7 - Solution (3/7)

1. Nombre de dominos
   - G simple et complet, $\forall i \in x$, $d_G(x)=n$-1

   ⇒ $n(n\text{-}1) = 2m$

   ⇒ $m= n(n\text{-}1)/2$

   ⇒ $m=5(5\text{-}1)/2$

   ⇒ $m=10$

   ⇒ Il y a 10 dominos

Exercice 7 - Solution (4/7)

2. On montre que l'on peut arranger ces dominos de façon à former une boucle fermée.
   - Règle de contact des dominos : contacter 2 dominos à travers les faces ayant le même numéro.
   - 2 arêtes adjacentes ⇒ former une chaîne
   - Les arranger en boucle fermée ⇒chaîne fermée
   - Vu qu'on utilise chaque domino une seule fois ⇒ Chaque arête est utilisée une seule fois dans la chaîne ⇒ Chaîne fermée et simple ⇒cycle
   - Utiliser tous les dominos ⇒ cycle Eulérien

Exercice 7 - Solution (5/7)

2. On montre que l'on peut arranger ces dominos de façon à former une boucle fermée.
   - Revient à montrer que G admet un cycle Eulérien
   - Chaque sommet est degré $n-1 = 5-1 = 4$
   - Tous les sommets sont degrés pairs ⇒ Pas de sommets de degrés impairs, de plus le graphe est connexe (car il est complet) ⇒ G admet une chaîne Eulérienne selon le théorème d'Euler et cette chaîne est un cycle (0 sommets de degrés impairs).

Exercice 7 - Solution (6/7)

3. Pourquoi n'est-il pas nécessaire de considérer les dominos doubles ?
   – Les dominos doubles sont représentés par des boucles.
   – Une boucle fait augmenter le degré d'un sommet de 2, ce qui ne change pas la parité de son degré.
   – De ce fait, le nombre de sommets de degrés impairs ne change pas.

Exercice 7 - Solution (7/7)

4. Si l'on prend maintenant des dominos dont les faces sont numérotées de 1 à n, est-il possible de les arranger de façon à former une boucle fermée ?
   – Chaque sommet est de degré $n-1$
   – Il faut que $n-1$ soit pair
   – C'à.d. $n-1=2k$ où k est un entier ≥ 0
   $\Rightarrow$ $n=2k+1$
   $\Rightarrow$ n doit être impair
   $\Rightarrow$ Le plus grand numéro des faces doit être impair

Exercice 8

- Soit un tournoi de volley-ball regroupant n clubs.
  – Chaque club affronte un autre exactement une (1) seule fois.
  – On veut représenter à travers un graphe les résultats de ce tournoi.
  – Sachez qu'en volley-ball il n'y a pas de score d'égalité.

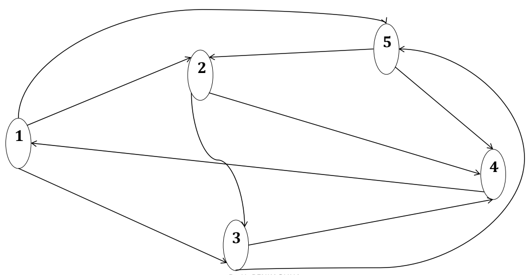
1. Modéliser (sans dessiner) le problème. Puis dessiner un exemple de graphe pour le cas de cinq (5) clubs.
2. Est-il possible de trouver une situation où le club C_1 a gagné contre C_2 et C_2 a battu C_3 et ainsi de suite jusqu'à C_{n-1} a gagné contre C_n ? Justifier. (C_i est un club quelconque)
3. Dans quels cas, peut-on avoir la même situation et de plus C_n a gagné contre C_1.

Exercice 8 - Solution (1/5)

1. Modéliser (sans dessiner) le problème.
   – Par un graphe orienté $G=(X, U)$
   – Chaque sommet représente un club
   – Chaque arc (i, j) représente le résultat du match entre les clubs i et j : i « a gagné contre » j.
   – Le graphe est complet car tous les clubs s'affrontent entre eux.
   – Le graphe est simple car tous les clubs s'affrontent une seule fois (pas d'arêtes parallèles) et un club ne peut pas affronter lui-même (pas de boucles).

Exercice 8 - Solution (2/5)

1. Dessiner un exemple de graphe pour le cas de cinq (5) clubs.



Exercice 8 - Solution (3/5)

2. Est-il possible de trouver une situation où le club C_1 a gagné contre C_2 et C_2 a battu C_3 et ainsi de suite jusqu'à C_{n-1} a gagné contre C_n ? Justifier. (C_i est un club quelconque)

- Ceci revient à avoir des arcs :
  – (1, 2), (2, 3) … (n-1, n)

Exercice 8 - Solution (4/5)

- En d'autres termes avoir un chemin dans G
  - γ=1 2 3 … n-1 n : élémentaire de longueur n-1 (passant par tous les sommets) $\Rightarrow$ Chemin Hamiltonien
  - On sait que G est un graphe simple et complet d'ordre $n \Rightarrow G$ est un tournoi T_n.
  - On sait que tout tournoi admet un chemin Hamiltonien

  $\Rightarrow$ C'est possible

Exercice 8 - Solution (5/5)

3. Dans quels cas, peut-on avoir la même situation et de plus C_n a gagné contre C_1.

- Cette situation correspond à un circuit Hamiltonien dans G
- On sait qu'un tournoi T_n fortement connexe admet un circuit Hamiltonien.
- Il faut que le graphe associé soit fortement connexe

Exercice 9

- Démontrer que si deux sommets $\boldsymbol{x}$ et $\boldsymbol{y} \in$ à une même composante fortement connexe $\boldsymbol{C}$, alors tout chemin de $\boldsymbol{x}$ à $\boldsymbol{y}$ est entièrement inclus dans $\boldsymbol{C}$.

Exercice 9 - Solution (1/2)

- On démontre par l'absurde
- On suppose qu'on a un graphe G qui admet au moins 2 CFCs C et C'
- x et $y \in C$ et $z \in C'$
- Tel qu'on a un chemin de x vers y qui passe par z
- $\gamma = x \dots z \dots y$

$\Rightarrow x\alpha z$…(1) et $z\alpha y$…(2)

- x et y dans la même CFC

$\Rightarrow x\alpha y$…(3) et $y\alpha x$…(4)

Exercice 9 - Solution (2/2)

- $x\alpha z$…(1)
- $z\alpha y$…(2)
- $x\alpha y$…(3)
- $y\alpha x$…(4)
- De (2) et (4) et par transitivité $\Rightarrow z\alpha x$…(5)
- De (1) et (5), on a z dans la même CFC que x

$\Rightarrow z \in C$ et sachant que $z \in C'$

$\Rightarrow z \in C \cap C' \Rightarrow C \cap C' \neq \phi$

$\Rightarrow$Contradiction

Chapitre 3
Arbre de Couverture Optimal
Série d'exercices de TD

Présenté par :
H. BENKAOUHA
Bureau 222, Faculté d'Informatique, USTHB
haroun.benkaouha@usthb.edu.dz
haroun.benkaouha@gmail.com

Exercice 1

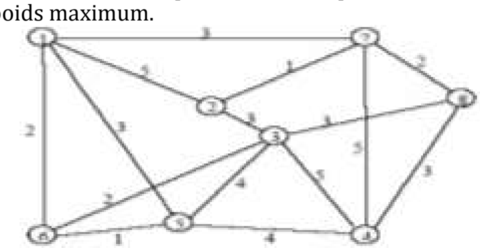
- Montrer que :
  - la moyenne des degrés des sommets d'un arbre
  - est strictement inférieure à 2.

Exercice 1 – Solution

- Par définition, dans un graphe d'ordre n et de taille m qui est un arbre, on a $m=n-1$
- D'un autre coté on a la somme des degrés est $2m$
- Donc la moyenne est $2m/n$
- On remplace m par $n-1$
- Moyenne_degrés = $2(n-1)/n = 2n/n - 2/n$
  $= 2 - 2/n < 2$

Exercice 2

- Trouver l'arbre de poids minimum puis l'arbre de poids maximum.



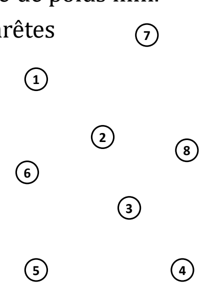
- Donner le codage de Prufer correspondant à l'arbre trouvé

Exercice 2 – Solution (1/4)

- On applique l'algorithme de **Kruksal** pour trouver l'arbre de couverture de poids min.
- Tri par ordre croissant des arêtes

Arête	Poids	Décision
{2,7}	1	
{5,6}	1	
{1,6}	2	
{3,6}	2	
{7,8}	2	
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

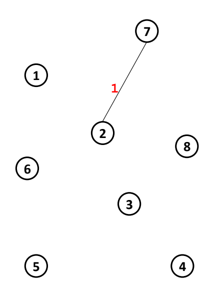


Exercice 2 – Solution (1/4)

- Algo. de Kruksal (1/7)

Arête	Poids	Décision
{2,7}	1	Oui
{5,6}	1	
{1,6}	2	
{3,6}	2	
{7,8}	2	
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

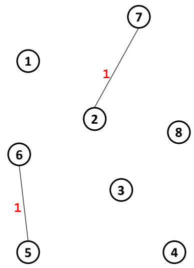


Exercice 2 – Solution (1/4)

- Algo. de Kruksal (2/7)

Arête	Poids	Décision
{2,7}	1	Oui
{5,6}	1	Oui
{1,6}	2	
{3,6}	2	
{7,8}	2	
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

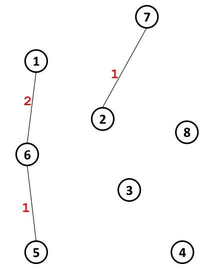


Exercice 2 – Solution (1/4)

- Algo. de Kruksal (3/7)

Arête	Poids	Décision
{2,7}	1	Oui
{5,6}	1	Oui
{1,6}	2	Oui
{3,6}	2	
{7,8}	2	
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

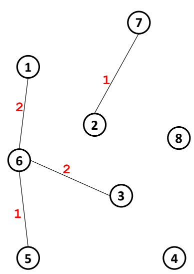


Exercice 2 – Solution (1/4)

- Algo. de Kruksal (4/7)

Arête	Poids	Décision
{2,7}	1	Oui
{5,6}	1	Oui
{1,6}	2	Oui
{3,6}	2	Oui
{7,8}	2	
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

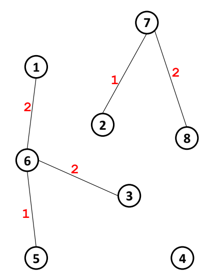


Exercice 2 – Solution (1/4)

- Algo. de Kruksal (5/7)

Arête	Poids	Décision
{2,7}	1	Oui
{5,6}	1	Oui
{1,6}	2	Oui
{3,6}	2	Oui
{7,8}	2	Oui
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

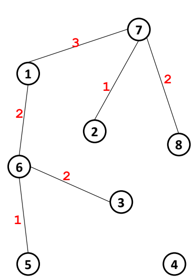


Exercice 2 – Solution (1/4)

- Algo. de Kruksal (6/7)

Arête	Poids	Décision
{2,7}	1	Oui
{5,6}	1	Oui
{1,6}	2	Oui
{3,6}	2	Oui
{7,8}	2	Oui
{1,5}	3	Non
{1,7}	3	Oui
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

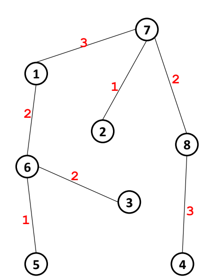


Exercice 2 – Solution (1/4)

- Algo. de Kruksal (7/7)

Arête	Poids	Décision
{2,7}	1	Oui
{5,6}	1	Oui
{1,6}	2	Oui
{3,6}	2	Oui
{7,8}	2	Oui
{1,5}	3	Non
{1,7}	3	Oui
{2,3}	3	Non

Arête	Poids	Décision
{3,8}	3	Non
{4,8}	3	Oui
{3,5}	4	Non
{4,5}	4	Non
{1,2}	5	Non
{3,4}	5	Non
{4,7}	5	Non

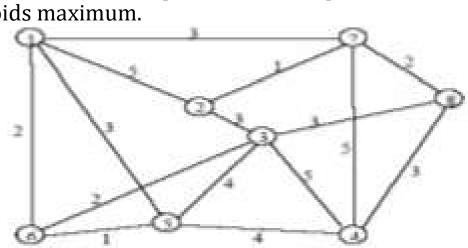


– Poids de l'arbre de couverture de poids minimal : 14

Enseignant : Dr. H. BENKAOUHA (haroun.benkaouha@usthb.edu.dz)

Exercice 2

- Trouver l'arbre de poids minimum puis l'arbre de poids maximum.



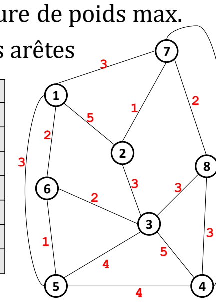
- Donner le codage de Prufer correspondant à l'arbre trouvé

Exercice 2 – Solution (2/4)

- On applique l'algorithme de Kruksal pour trouver l'arbre de couverture de poids max.
- Tri par ordre croissant des arêtes

Arête	Poids	Décision
{2,7}	1	
{5,6}	1	
{1,6}	2	
{3,6}	2	
{7,8}	2	
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

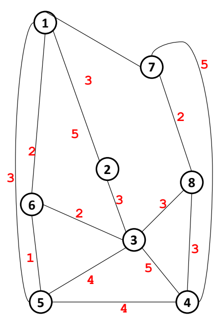


Exercice 2 – Solution (2/4)

- Algo. de Kruksal 1/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	
{1,6}	2	
{3,6}	2	
{7,8}	2	
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

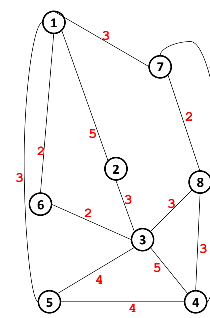


Exercice 2 – Solution (2/4)

- Algo. de Kruksal 2/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	
{3,6}	2	
{7,8}	2	
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

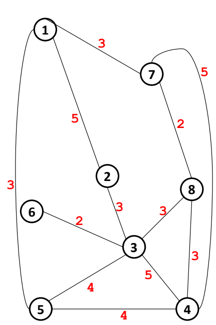


Exercice 2 – Solution (2/4)

- Algo. de Kruksal 3/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	Non
{3,6}	2	
{7,8}	2	
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

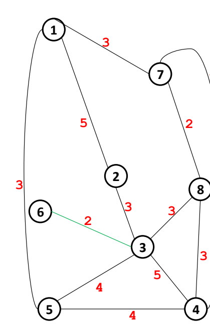


Exercice 2 – Solution (2/4)

- Algo. de Kruksal 4/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	Non
{3,6}	2	Oui
{7,8}	2	
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

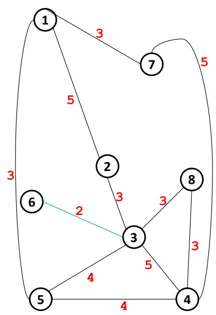


Exercice 2 – Solution (2/4)

- Algo. de Kruksal 5/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	Non
{3,6}	2	Oui
{7,8}	2	Non
{1,5}	3	
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

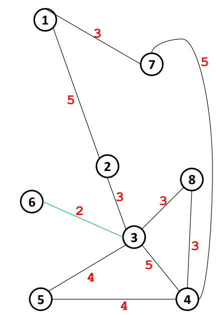


Exercice 2 – Solution (2/4)

- Algo. de Kruksal 6/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	Non
{3,6}	2	Oui
{7,8}	2	Non
{1,5}	3	Non
{1,7}	3	
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

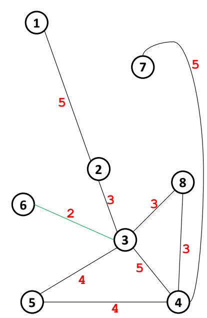


Exercice 2 – Solution (2/4)

- Algo. de Kruksal 7/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	Non
{3,6}	2	Oui
{7,8}	2	Non
{1,5}	3	Non
{1,7}	3	Non
{2,3}	3	

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

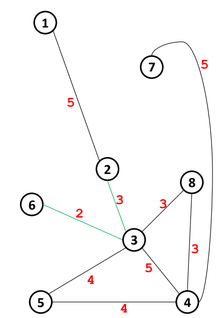


Exercice 2 – Solution (2/4)

- Algo. de Kruksal 8/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	Non
{3,6}	2	Oui
{7,8}	2	Non
{1,5}	3	Non
{1,7}	3	Non
{2,3}	3	Oui

Arête	Poids	Décision
{3,8}	3	
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

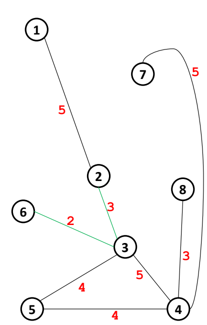


Exercice 2 – Solution (2/4)

- Algo. de Kruksal 9/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	Non
{3,6}	2	Oui
{7,8}	2	Non
{1,5}	3	Non
{1,7}	3	Non
{2,3}	3	Oui

Arête	Poids	Décision
{3,8}	3	Non
{4,8}	3	
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

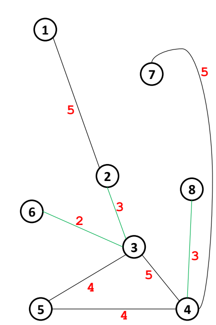


Exercice 2 – Solution (2/4)

- Algo. de Kruksal 10/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	Non
{3,6}	2	Oui
{7,8}	2	Non
{1,5}	3	Non
{1,7}	3	Non
{2,3}	3	Oui

Arête	Poids	Décision
{3,8}	3	Non
{4,8}	3	Oui
{3,5}	4	
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	



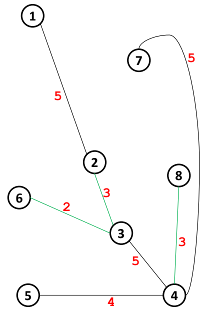
Exercice 2 – Solution (2/4)

- Algo. de Kruksal 11/11

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	Non
{3,6}	2	Oui
{7,8}	2	Non
{1,5}	3	Non
{1,7}	3	Non
{2,3}	3	Oui

Arête	Poids	Décision
{3,8}	3	Non
{4,8}	3	Oui
{3,5}	4	Non
{4,5}	4	
{1,2}	5	
{3,4}	5	
{4,7}	5	

On prend toutes les arêtes restantes



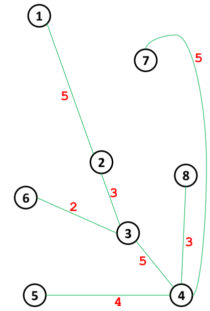
Exercice 2 – Solution (2/4)

- Algo. de Kruksal (fin)

Arête	Poids	Décision
{2,7}	1	Non
{5,6}	1	Non
{1,6}	2	Non
{3,6}	2	Oui
{7,8}	2	Non
{1,5}	3	Non
{1,7}	3	Non
{2,3}	3	Oui

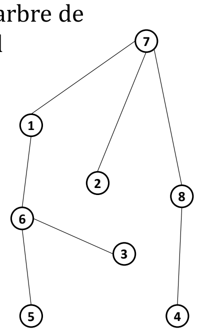
Arête	Poids	Décision
{3,8}	3	Non
{4,8}	3	Oui
{3,5}	4	Non
{4,5}	4	Oui
{1,2}	5	Oui
{3,4}	5	Oui
{4,7}	5	Oui

Poids de l'arbre de couverture de poids max : 27



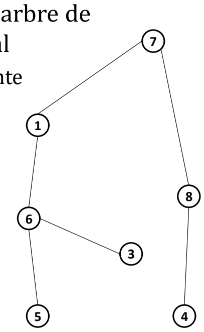
Exercice 2 – Solution (3/4)

- Codage de Prufer (1/7) de l'arbre de couverture de poids minimal
  - Sommets de degré 1 :
  - 2, 3, 4, 5
  - Le min : 2
  - 2 est relié à 7
  - P=7



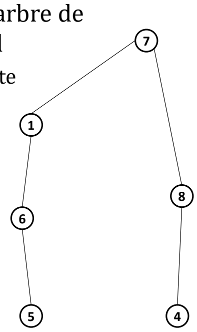
Exercice 2 – Solution (3/4)

- Codage de Prufer (2/7) de l'arbre de couverture de poids minimal
  - Supprimer 2 et l'arête incidente
  - Sommets de degré 1 :
  - 3, 4, 5
  - Le min : 3
  - 3 est relié à 6
  - P=7 6



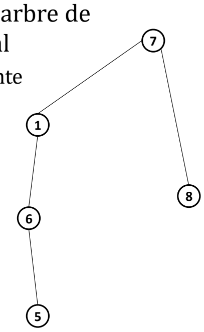
Exercice 2 – Solution (3/4)

- Codage de Prufer (3/7) de l'arbre de couverture de poids minimal
  - Supprimer 3 et l'arête incidente
  - Sommets de degré 1 :
  - 4, 5
  - Le min : 4
  - 4 est relié à 8
  - P=7 6 8



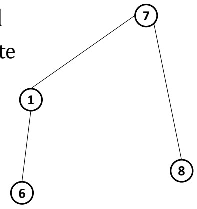
Exercice 2 – Solution (3/4)

- Codage de Prufer (4/7) de l'arbre de couverture de poids minimal
  - Supprimer 4 et l'arête incidente
  - Sommets de degré 1 :
  - 5, 8
  - Le min : 5
  - 5 est relié à 6
  - P=7 6 8 6



Exercice 2 – Solution (3/4)

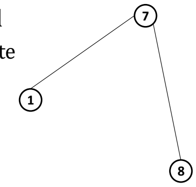
- Codage de Prufer (5/7) de l'arbre de couverture de poids minimal
  - Supprimer 5 et l'arête incidente
  - Sommets de degré 1 :
  - 6, 8
  - Le min : 6
  - 6 est relié à 1
  - P=7 6 8 6 1



Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 31

Exercice 2 – Solution (3/4)

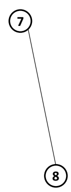
- Codage de Prufer (6/7) de l'arbre de couverture de poids minimal
  - Supprimer 6 et l'arête incidente
  - Sommets de degré 1 :
  - 1, 8
  - Le min : 1
  - 1 est relié à 7
  - P=7 6 8 6 1 7



Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 32

Exercice 2 – Solution (3/4)

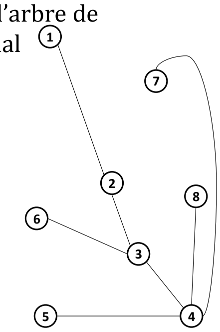
- Codage de Prufer (7/7) de l'arbre de couverture de poids minimal
  - Supprimer 1 et l'arête incidente
  - Il ne reste que 2 sommet
  - ⇒ **Fin.**
  - **P=7 6 8 6 1 7**



Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 33

Exercice 2 – Solution (4/4)

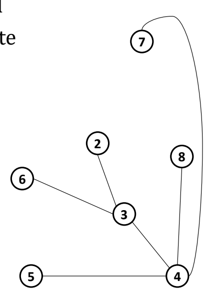
- Codage de Prufer (1/7) de l'arbre de couverture de poids maximal
  - Sommets de degré 1 :
  - 1, 5, 6, 7, 8
  - Le min : 1
  - 1 est relié à 2
  - P=2



Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 34

Exercice 2 – Solution (4/4)

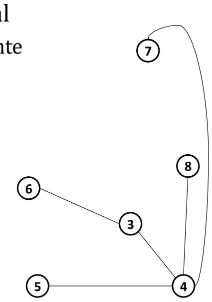
- Codage de Prufer (2/7) de l'arbre de couverture de poids minimal
  - Supprimer 1 et l'arête incidente
  - Sommets de degré 1 :
  - 2, 5, 6, 7, 8
  - Le min : 2
  - 2 est relié à 3
  - P=2 3



Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 35

Exercice 2 – Solution (4/4)

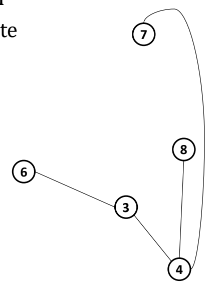
- Codage de Prufer (3/7) de l'arbre de couverture de poids minimal
  - Supprimer 2 et l'arête incidente
  - Sommets de degré 1 :
  - 5, 6, 7, 8
  - Le min : 5
  - 5 est relié à 4
  - P=2 3 4



Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 36

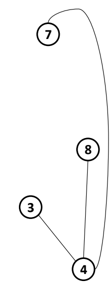
Exercice 2 – Solution (4/4)

- Codage de Prufer (4/7) de l'arbre de couverture de poids minimal
  - Supprimer 5 et l'arête incidente
  - Sommets de degré 1 :
  - 6, 7, 8
  - Le min : 6
  - 6 est relié à 3
  - P=2 3 4 3



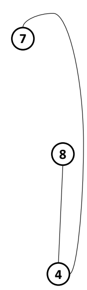
Exercice 2 – Solution (4/4)

- Codage de Prufer (5/7) de l'arbre de couverture de poids minimal
  - Supprimer 6 et l'arête incidente
  - Sommets de degré 1 :
  - 3, 7, 8
  - Le min : 3
  - 3 est relié à 4
  - P=2 3 4 3 4



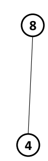
Exercice 2 – Solution (4/4)

- Codage de Prufer (6/7) de l'arbre de couverture de poids minimal
  - Supprimer 3 et l'arête incidente
  - Sommets de degré 1 :
  - 7, 8
  - Le min : 7
  - 7 est relié à 4
  - P=2 3 4 3 4 4



Exercice 2 – Solution (4/4)

- Codage de Prufer (7/7) de l'arbre de couverture de poids minimal
  - Supprimer 7 et l'arête incidente
  - Il ne reste que 2 sommet
  - ⇒ **Fin.**
  - **P=2 3 4 3 4 4**



Exercice 3

- On désire installer au moindre coût un réseau de communication entre divers sites.
- Les coûts des connexions intersites sont les suivants (symétriques) :

	B	***C***	***D***	***E***	***F***	***G***	***H***
A	5	18	9	13	7	38	22
	B	17	11	7	10	38	15
		C	27	23	15	20	25
			D	20	15	40	25
				E	15	40	30
					F	35	10
						G	45

Exercice 3 - Suite

1. Identifier le problème associé.
2. Déterminer la solution optimale.

Exercice 3 – Solution (1/4)

1. Identifier le problème associé.

- Modélisation
  - Chaque sommet x représente un site x, x de A à H.
  - Chaque arête $\{i, j\}$ représente une connexion intersites.
  - Le coût de la connexion intersites est représenté par le poids de l'arête correspondante.

Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 43

Exercice 3 – Solution (2/4)

1. Identifier le problème associé.

- Identification
  - Tous les sites connectés : graphe connexe
  - Coût minimal : graphe connexe minimal avec somme de poids des arêtes minimal
  - Revient à trouver l'arbre de couverture maximal de poids minimal.

Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 44

Exercice 3 – Solution (3/4)

2. Déterminer la solution optimale.
   - On applique l'algorithme de Kruksal
   - Tri par ordre croissant des arêtes

Arête	Poids	Décision
{A,B}	5	
{A,F}	7	
{B,E}	7	
{A,D}	9	
{D,F}	10	
{F,H}	10	
{B,D}	11	
{A,E}	13	
{B,H}	15	
{C,F}	15	

Arête	Poids	Décision
{D,F}	15	
{E,F}	15	
{B,C}	17	
{A,C}	18	
{C,G}	20	
{D,E}	20	
{A,H}	22	
{C,E}	23	
{C,H}	25	

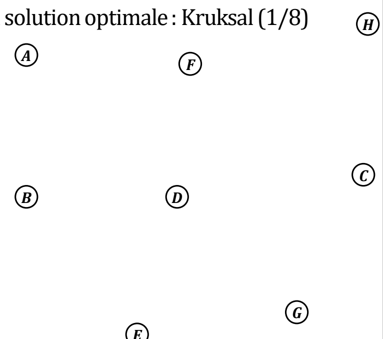
Arête	Poids	Décision
{D,H}	25	
{C,D}	27	
{E,H}	30	
{F,G}	35	
{A,G}	38	
{B,G}	38	
{D,G}	40	
{E,G}	40	
{G,H}	45	

Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 45

Exercice 3 – Solution (4/4)

2. Déterminer la solution optimale : Kruksal (1/8)

Arête	Poids	Décision
{A,B}	5	
{A,F}	7	
{B,E}	7	
{A,D}	9	
{D,F}	10	
{F,H}	10	
{B,D}	11	
{A,E}	13	
{B,H}	15	
{C,F}	15	

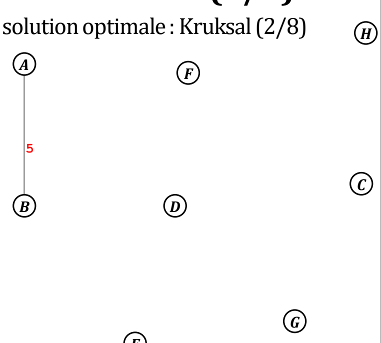


Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 46

Exercice 3 – Solution (4/4)

2. Déterminer la solution optimale : Kruksal (2/8)

Arête	Poids	Décision
{A,B}	5	Oui
{A,F}	7	
{B,E}	7	
{A,D}	9	
{D,F}	10	
{F,H}	10	
{B,D}	11	
{A,E}	13	
{B,H}	15	
{C,F}	15	

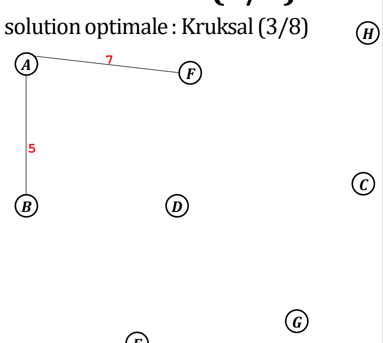


Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 47

Exercice 3 – Solution (4/4)

2. Déterminer la solution optimale : Kruksal (3/8)

Arête	Poids	Décision
{A,B}	5	Oui
{A,F}	7	Oui
{B,E}	7	
{A,D}	9	
{D,F}	10	
{F,H}	10	
{B,D}	11	
{A,E}	13	
{B,H}	15	
{C,F}	15	

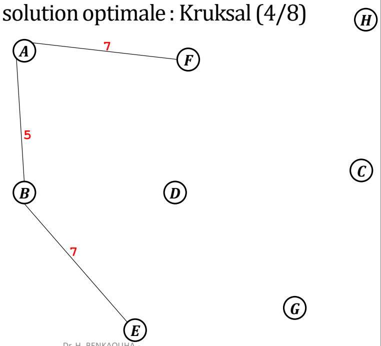


Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 48

Exercice 3 – Solution (4/4)

2. Déterminer la solution optimale : Kruksal (4/8)

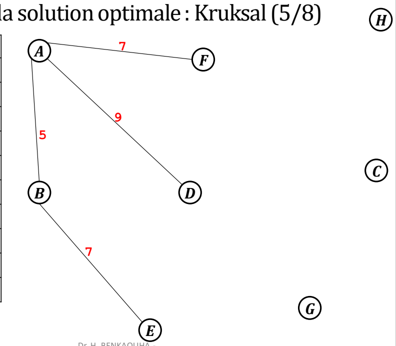
Arête	Poids	Décision
{A,B}	5	Oui
{A,F}	7	Oui
{B,E}	7	Oui
{A,D}	9	
{D,F}	10	
{F,H}	10	
{B,D}	11	
{A,E}	13	
{B,H}	15	
{C,F}	15	



Exercice 3 – Solution (4/4)

2. Déterminer la solution optimale : Kruksal (5/8)

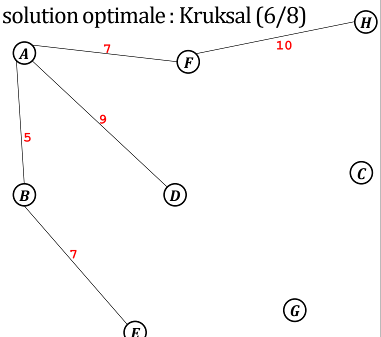
Arête	Poids	Décision
{A,B}	5	Oui
{A,F}	7	Oui
{B,E}	7	Oui
{A,D}	9	Oui
{D,F}	10	
{F,H}	10	
{B,D}	11	
{A,E}	13	
{B,H}	15	
{C,F}	15	



Exercice 3 – Solution (4/4)

2. Déterminer la solution optimale : Kruksal (6/8)

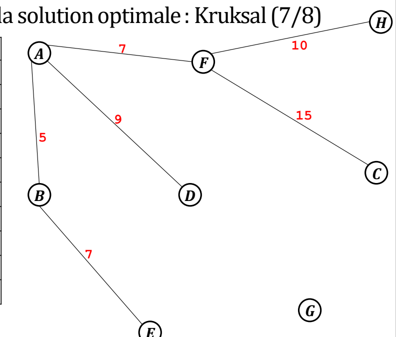
Arête	Poids	Décision
{A,B}	5	Oui
{A,F}	7	Oui
{B,E}	7	Oui
{A,D}	9	Non
{D,F}	10	Non
{F,H}	10	Oui
{B,D}	11	
{A,E}	13	
{B,H}	15	
{C,F}	15	



Exercice 3 – Solution (4/4)

2. Déterminer la solution optimale : Kruksal (7/8)

Arête	Poids	Décision
{A,B}	5	Oui
{A,F}	7	Oui
{B,E}	7	Oui
{A,D}	9	Non
{D,F}	10	Non
{F,H}	10	Oui
{B,D}	11	Non
{A,E}	13	Non
{B,H}	15	Non
{C,F}	15	Oui

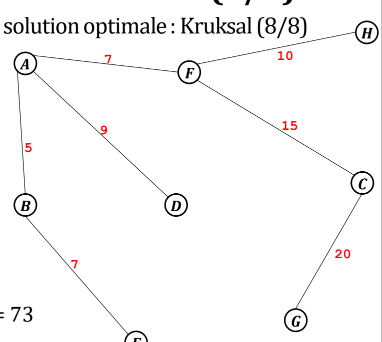


Exercice 3 – Solution (4/4)

2. Déterminer la solution optimale : Kruksal (8/8)

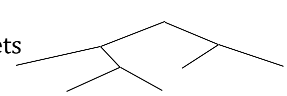
Arête	Poids	Décision
{D,F}	15	Non
{E,F}	15	Non
{B,C}	17	Non
{A,C}	18	Non
{C,G}	20	Oui
{D,E}	20	
{A,H}	22	
{C,E}	23	
{C,H}	25	

Arbre de poids = 73



Exercice 4

- Un arbre est dit binaire, s'il est constitué :
  - d'un unique sommet de degré 2 (appelé racine de l'arbre)
  - et tout autre sommet est soit de degré 3, soit de degré 1.
- Les sommets de degré 1 sont appelés les feuilles de l'arbre.
- Exemple de 9 sommets et 5 feuilles

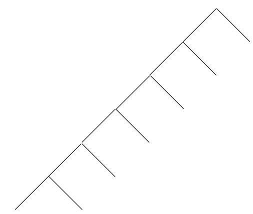


Exercice 4 - Suite

1. Énumérez (à isomorphisme près) tous les arbres binaires ayant exactement 7 feuilles.
2. Combien de sommets peut avoir un arbre binaire ayant exactement 7 feuilles.
3. Combien de sommets peut avoir un arbre binaire ayant exactement k feuilles (avec $k \geq 2$).

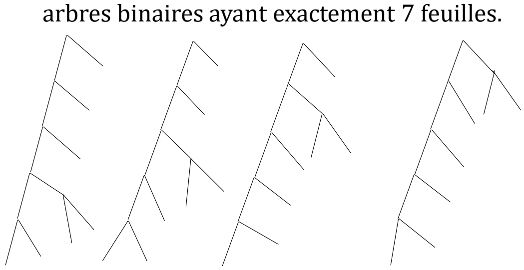
Exercice 4 – Solution (1/7)

1. Énumérez (à isomorphisme près) tous les arbres binaires ayant exactement 7 feuilles.



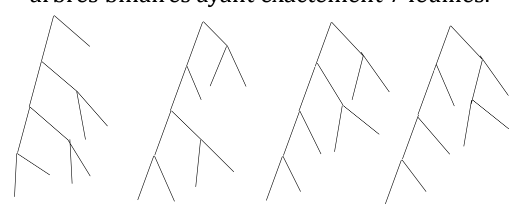
Exercice 4 – Solution (2/7)

1. Énumérez (à isomorphisme près) tous les arbres binaires ayant exactement 7 feuilles.



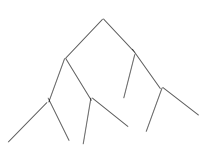
Exercice 4 – Solution (3/7)

1. Énumérez (à isomorphisme près) tous les arbres binaires ayant exactement 7 feuilles.



Exercice 4 – Solution (4/7)

1. Énumérez (à isomorphisme près) tous les arbres binaires ayant exactement 7 feuilles.



Exercice 4 – Solution (5/7)

2. Combien de sommets peut avoir un arbre binaire ayant exactement 7 feuilles.

- Il y a 13 sommets selon la question précédente.

Exercice 4 – Solution (6/7)

3. Combien de sommets peut avoir un arbre binaire ayant exactement k feuilles (avec $k \geq 2$).

- On pose n le nombre de sommets et m le nombre d'arêtes
- On a k feuilles (degré 1), 1 racine (degré 2) et p autres sommets (degré 3)
- $n=k+p+1$
- $2*1 + 1*k + 3*p = 2m$ (formule des degrés)
- $m=n-1$ (propriété d'un arbre)

Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 61

Exercice 4 – Solution (7/7)

3. Combien de sommets peut avoir un arbre binaire ayant exactement k feuilles (avec $k \geq 2$).

- $n=k+p+1$ … (1)
- $k + 3p + 2 = 2m$ … (2)
- $m=n-1$ … (3)
- De (2) et (3) $\Rightarrow k + 3p + 2 = 2\,(n-1)$ … (4)
- De (1) $\Rightarrow p=n - k - 1$
- On remplace dans (4) $\Rightarrow k+3(n-k-1)+2=2(n-1)$
- $\Rightarrow k+3n-3k-3+2=2n-2 \Rightarrow n-2k=-1 \Rightarrow n=2k-1$

Dr. H. BENKAOUHA - Faculté d'Informatique (USTHB) 62

Chapitre 4 : Chemin optimal
Série d'exercices de TD
avec Corrigé

Présenté par :
H. BENKAOUHA
Bureau 222, Faculté d'Informatique, USTHB
haroun.benkaouha@usthb.edu.dz
haroun.benkaouha@gmail.com

Exercice 1

- Soit le graphe orienté valué $G = (X, U, p)$ donné par la matrice ci-dessous :

	1	2	3	4	5	6	7	8
1		1		5		-4		2
2								
3		8			2		1	
4						-1		
5								
6		3						
7				2	1			
8		1						

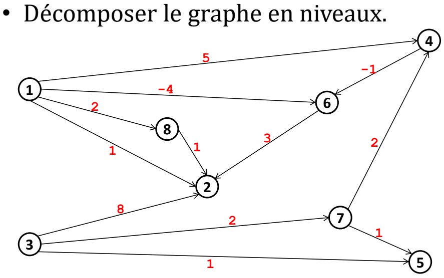
- $m_{ij}=k$ veut dire que le poids de l'arc (i, j) est k.
- Si m_{ij} est vide alors l'arc (i, j) n'existe pas.

Exercice 1 - Suite

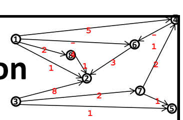
- Décomposer le graphe en niveaux.
- Appliquer l'algorithme de Bellmann-Ford sur ce graphe à partir du (des) sommet(s) source(s) afin d'obtenir les chemins de poids minimaux.

Exercice 1 – Solution

- Décomposer le graphe en niveaux.



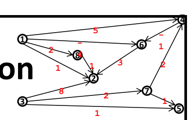
Exercice 1 – Solution



- Décomposer le graphe en niveaux.
- Niveau 0
- 1 et 3 sont des sources

x	1	2	3	4	5	6	7	8
v(x)	0		0					

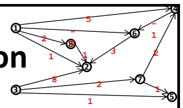
Exercice 1 – Solution



- Décomposer le graphe en niveaux.
- Niveau 1
- Le 7 et le 8 ont uniquement des arcs entrant venant de sommets de niveau 0

x	1	2	3	4	5	6	7	8
v(x)	0		0				1	1

Exercice 1 – Solution

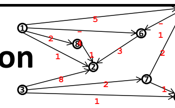


- Décomposer le graphe en niveaux.
- Niveau 2
- Le 4 et le 5 ont uniquement des arcs entrant venant de sommets de niveau ≤ 1

x	1	2	3	4	5	6	7	8
v(x)	0		0	2	2		1	1

Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 7

Exercice 1 – Solution

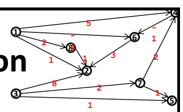


- Décomposer le graphe en niveaux.
- Niveau 3
- Seul le 5 qui uniquement des arcs entrant venant de sommets de niveau ≤ 2

x	1	2	3	4	5	6	7	8
v(x)	0		0	2	2	3	1	1

Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 8

Exercice 1 – Solution



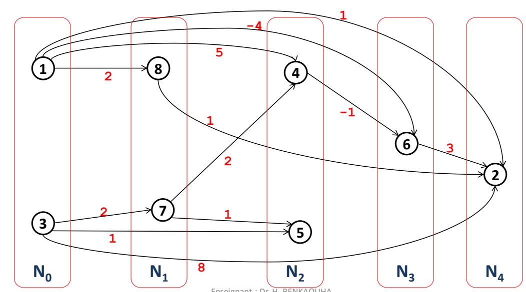
- Décomposer le graphe en niveaux.
- Niveau 4
- Seul le 2 qui uniquement des arcs entrant venant de sommets de niveau ≤ 3

x	1	2	3	4	5	6	7	8
v(x)	0	4	0	2	2	3	1	1

Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 9

Exercice 1 – Solution

- Décomposer le graphe en niveaux.

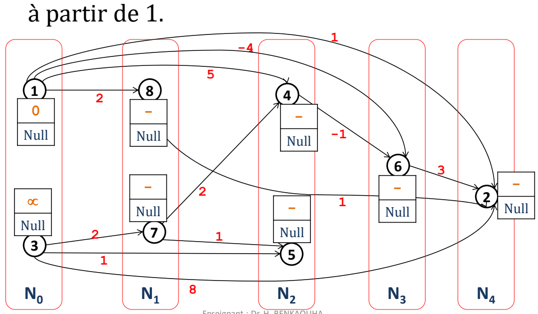


Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 10

Exercice 1 – Solution

π
Préd

- Application de l'algorithme de Bellmann-Ford à partir de 1.

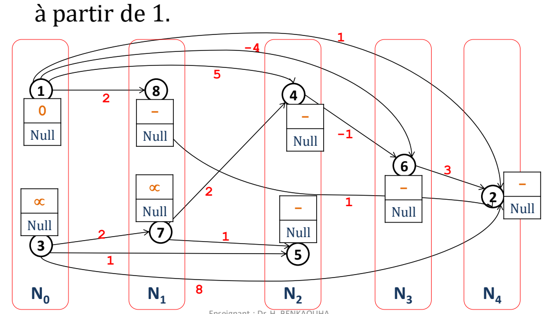


Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 11

Exercice 1 – Solution

π
Préd

- Application de l'algorithme de Bellmann-Ford à partir de 1.

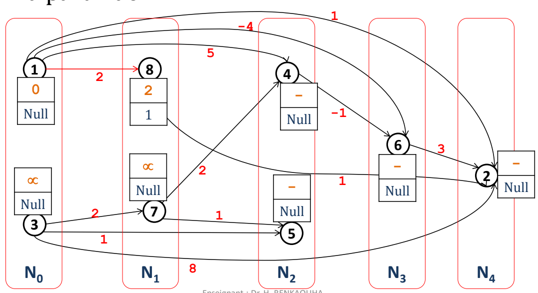


Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 12

π
Préd

Exercice 1 – Solution

- Application de l'algorithme de Bellmann-Ford à partir de 1.

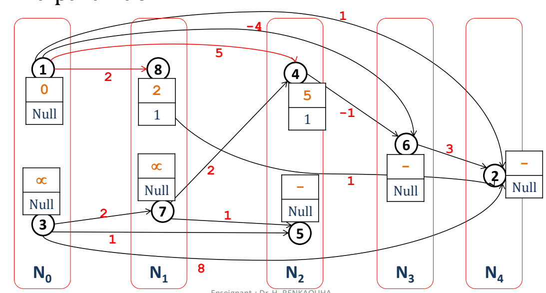


Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 13

π
Préd

Exercice 1 – Solution

- Application de l'algorithme de Bellmann-Ford à partir de 1.

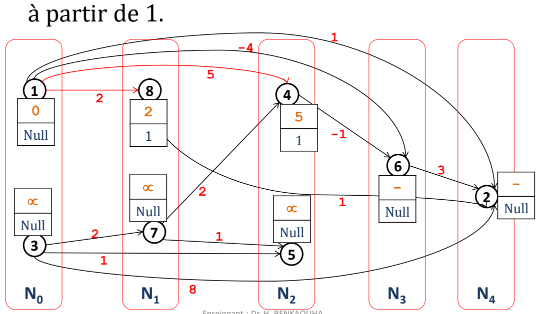


Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 14

π
Préd

Exercice 1 – Solution

- Application de l'algorithme de Bellmann-Ford à partir de 1.

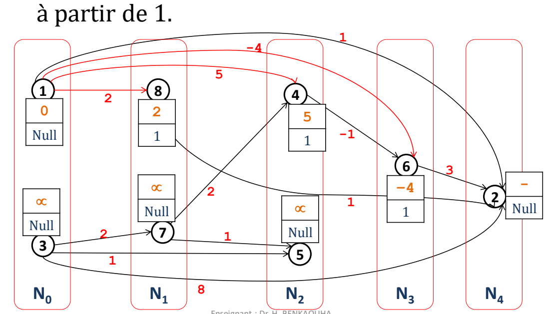


Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 15

π
Préd

Exercice 1 – Solution

- Application de l'algorithme de Bellmann-Ford à partir de 1.

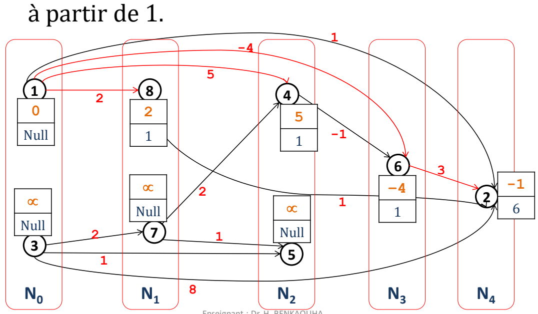


Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 16

π
Préd

Exercice 1 – Solution

- Application de l'algorithme de Bellmann-Ford à partir de 1.

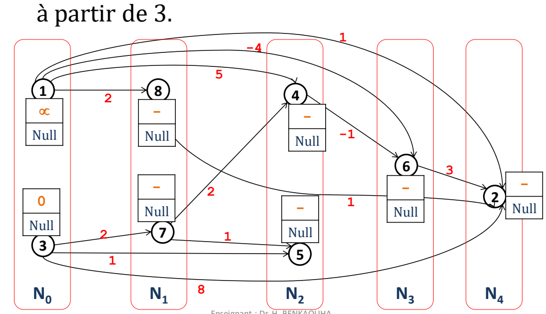


Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 17

π
Préd

Exercice 1 – Solution

- Application de l'algorithme de Bellmann-Ford à partir de 3.

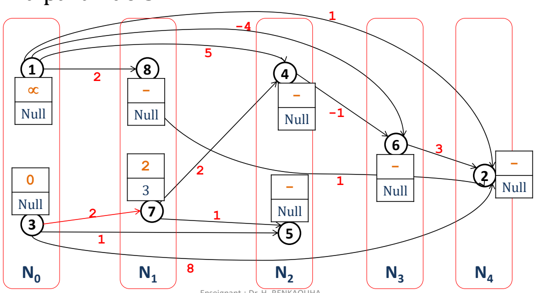


Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 18

π | Préd

Exercice 1 – Solution

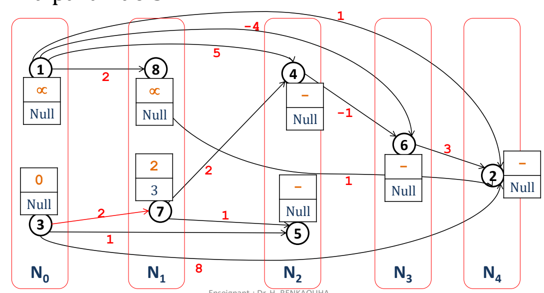
- Application de l'algorithme de Bellmann-Ford à partir de 3.



π | Préd

Exercice 1 – Solution

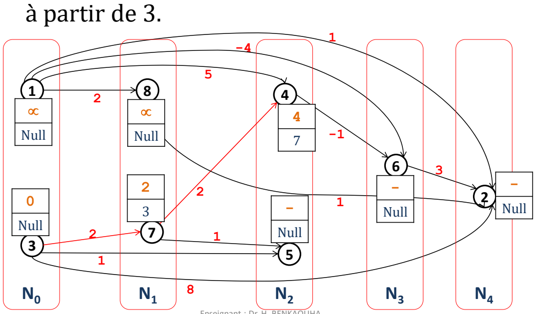
- Application de l'algorithme de Bellmann-Ford à partir de 3.



π | Préd

Exercice 1 – Solution

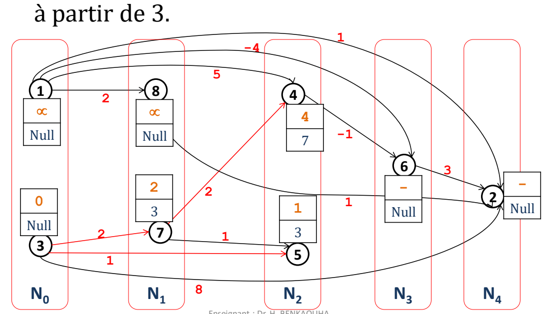
- Application de l'algorithme de Bellmann-Ford à partir de 3.



π | Préd

Exercice 1 – Solution

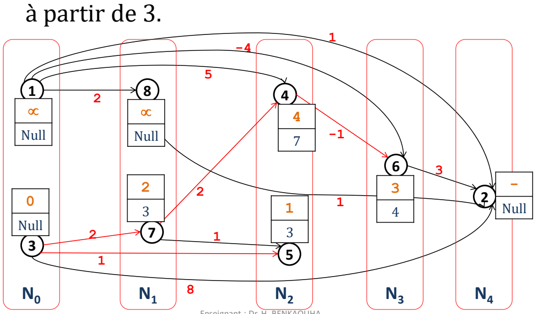
- Application de l'algorithme de Bellmann-Ford à partir de 3.



π | Préd

Exercice 1 – Solution

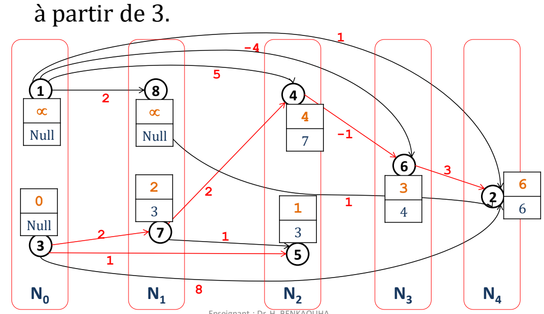
- Application de l'algorithme de Bellmann-Ford à partir de 3.



π | Préd

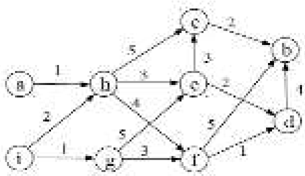
Exercice 1 – Solution

- Application de l'algorithme de Bellmann-Ford à partir de 3.



Exercice 2

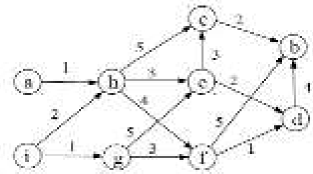
- Considérons le graphe G suivant :



1. Déterminer les niveaux de ce graphe
2. Donner la longueur des chemins minimaux reliant le sommet a aux autres sommets du graphe.

Exercice 2 – Solution

- Les niveaux du graphe :

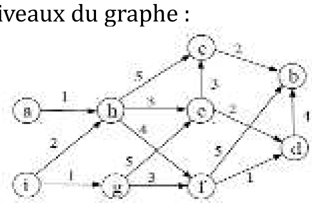


- Niveau 0

x	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*
v(x)	0								0

Exercice 2 – Solution

- Les niveaux du graphe :

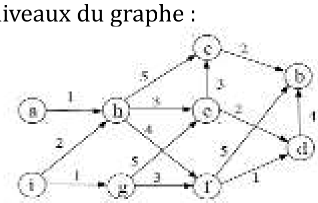


- Niveau 1

x	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*
v(x)	0						1	1	0

Exercice 2 – Solution

- Les niveaux du graphe :

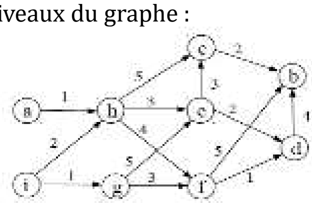


- Niveau 2

x	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*
v(x)	0				2	2	1	1	0

Exercice 2 – Solution

- Les niveaux du graphe :

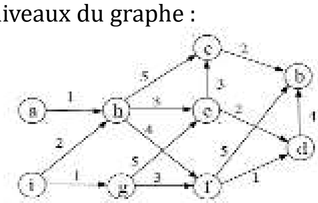


- Niveau 3

x	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*
v(x)	0		3	3	2	2	1	1	0

Exercice 2 – Solution

- Les niveaux du graphe :



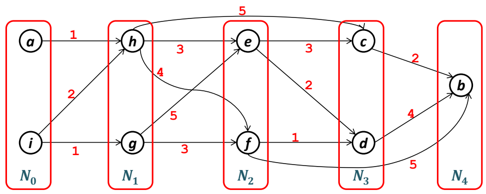
- Niveau 4

x	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*
v(x)	0	4	3	3	2	2	1	1	0

Exercice 2 – Solution

- Les niveaux du graphe :

x	a	b	c	d	e	f	g	h	i
$v(x)$	0	4	3	3	2	2	1	1	0

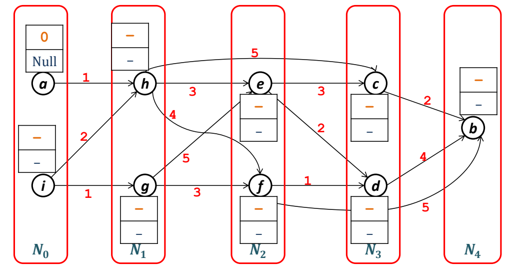


Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 31

Exercice 2 – Solution

π / Préd

- Les chemin optimaux (poids min.) à partir de a :

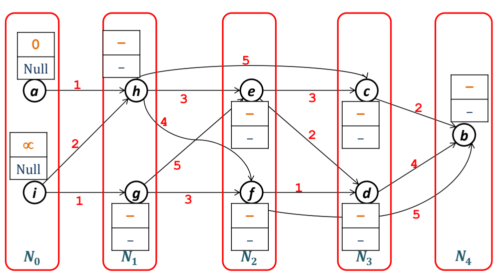


Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 32

Exercice 2 – Solution

π / Préd

- Les chemin optimaux (poids min.) à partir de a :

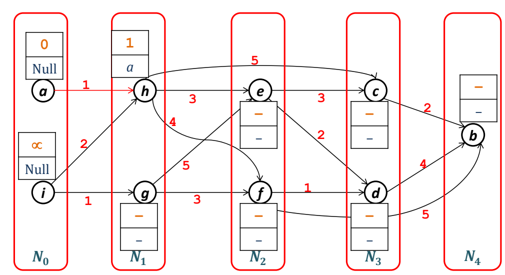


Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 33

Exercice 2 – Solution

π / Préd

- Les chemin optimaux (poids min.) à partir de a :

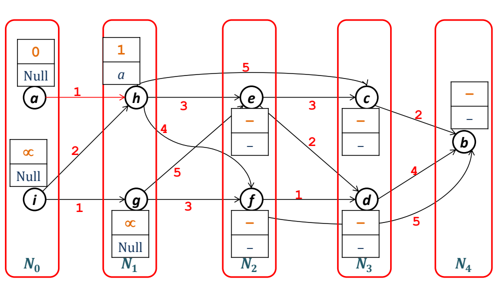


Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 34

Exercice 2 – Solution

π / Préd

- Les chemin optimaux (poids min.) à partir de a :

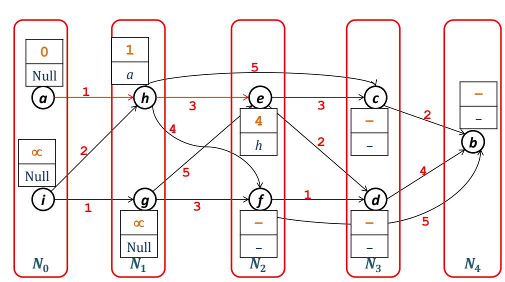


Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 35

Exercice 2 – Solution

π / Préd

- Les chemin optimaux (poids min.) à partir de a :

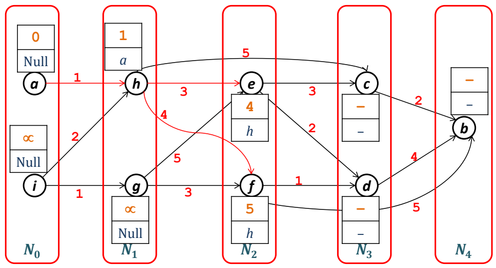


Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 36

π | Préd

Exercice 2 – Solution

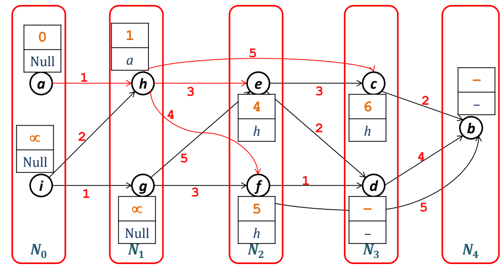
- Les chemin optimaux (poids min.) à partir de a :



π | Préd

Exercice 2 – Solution

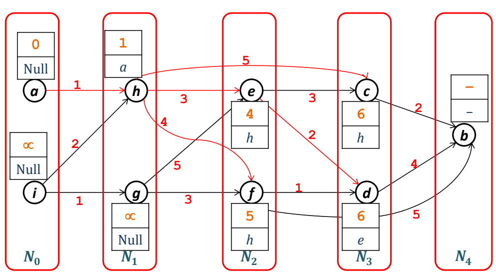
- Les chemin optimaux (poids min.) à partir de a :



π | Préd

Exercice 2 – Solution

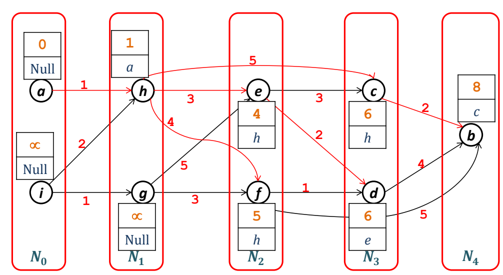
- Les chemin optimaux (poids min.) à partir de a :



π | Préd

Exercice 2 – Solution

- Les chemin optimaux (poids min.) à partir de a :



Exercice 3

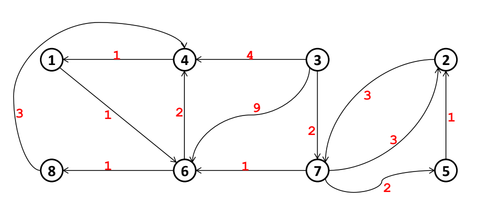
- Soit un graphe orienté pondéré $G=(X, U, p)$. Nous donnons les poids associés aux arcs comme le montre le tableau ci-dessous.

	1	2	3	4	5	6	7	8
1						1		
2							3	
3				4		9	2	
4	1							
5		1						
6				2				1
7		3			2	1		
8				3				

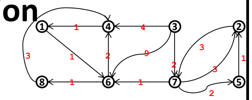
- Appliquer l'algorithme le plus adéquat pour calculer les chemins de poids minimaux à partir du sommet 3.

Exercice 3 – Solution

- Tous les sommets ont des arcs sortants ⇒ Pas de puits ⇒ Le graphe admet au moins un circuit.
- On applique l'algorithme de Dijkstra.

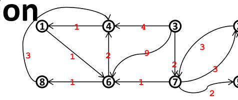


Exercice 3 – Solution



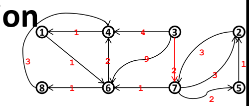
k	x	(x,y)	π								PRE							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
			∝	∝	0	∝	∝	∝	∝	∝	-	-	-	-	-	-	-	-
1																		
2																		
3																		
4																		
5																		
6																		
7																		

Exercice 3 – Solution



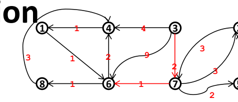
k	x	(x,y)	π								PRE							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
			∝	∝	0	~~∝~~	∝	~~∝~~	~~∝~~	∝	-	-	-	-	-	-	-	-
1	3	(3,4) (3,6) (3,7)				4		9	2					3		3	3	

Exercice 3 – Solution



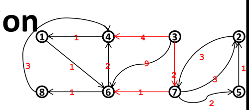
k	x	(x,y)	π								PRE							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
			∝	~~∝~~	0	~~∝~~	~~∝~~	~~∝~~	~~∝~~	∝	-	-	-	-	-	-	-	-
1	3	(3,4) (3,6) (3,7)				4		~~9~~	2					3		~~3~~	3	
2	7	(7,2) (7,5) (7,6)		5			4	3				7			7	7		

Exercice 3 – Solution



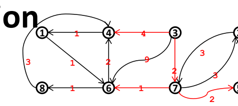
k	x	(x,y)	π								PRE							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
			∝	~~∝~~	0	~~∝~~	~~∝~~	~~∝~~	~~∝~~	~~∝~~	-	-	-	-	-	-	-	-
1	3	(3,4) (3,6) (3,7)				4		~~9~~	2					3		~~3~~	3	
2	7	(7,2) (7,5) (7,6)		5			4	3				7			7	7		
3	6	(6,4) (6,8)				~~5~~				4								6

Exercice 3 – Solution



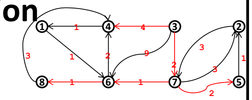
k	x	(x,y)	π								PRE							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
			~~∝~~	~~∝~~	0	~~∝~~	~~∝~~	~~∝~~	~~∝~~	~~∝~~	-	-	-	-	-	-	-	-
1	3	(3,4) (3,6) (3,7)				4		~~9~~	2					3		~~3~~	3	
2	7	(7,2) (7,5) (7,6)		5			4	3				7			7	7		
3	6	(6,4) (6,8)				~~5~~				4								6
4	4	(4,1)	5								4							

Exercice 3 – Solution



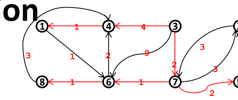
k	x	(x,y)	π								PRE							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
			~~∝~~	~~∝~~	0	~~∝~~	~~∝~~	~~∝~~	~~∝~~	~~∝~~	-	-	-	-	-	-	-	-
1	3					4		~~9~~	2					3		~~3~~	3	
2	7	(7,2) (7,5) (7,6)		5			4	3				7			7	7		
3	6	(6,4) (6,8)				~~5~~				4								6
4	4	(4,1)	5								4							
5	5	(5,2)		~~5~~														

Exercice 3 – Solution



k	x	(x,y)	π								PRE							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
			~~∞~~	~~∞~~	0	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~	-	-	-	-	-	-	-	-
1	3					4		~~9~~	2					3		~~3~~	3	
2	7			5			4	3				7			7	7		
3	6	(6,4) (6,8)				~~5~~				4								6
4	4	(4,1)	5								4							
5	5	(5,2)		~~5~~														
6	8	~~(8,4)~~																

Exercice 3 – Solution

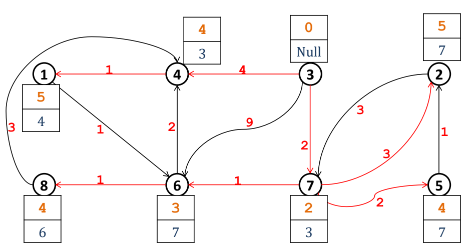


k	x	(x,y)	π								PRE							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
			~~∞~~	~~∞~~	0	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~	-	-	-	-	-	-	-	-
1	3					4		~~9~~	2					3		~~3~~	3	
2	7			5			4	3				7			7	7		
3	6	(6,4) (6,8)				~~5~~				4								6
4	4	(4,1)	5								4							
5	5	(5,2)		~~5~~														
6	8	~~(8,4)~~																
7	1	~~(1,6)~~																

π
Préd

Exercice 3 – Solution

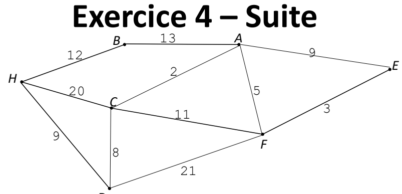
- Fin Algo après n–1 itérations.



Exercice 4

- Des touristes sont logés dans un hôtel H.
- Un guide fait visiter six (6) sites touristiques (A, B, C, D, E et F).
- Le graphe ci-dessous représente cette situation,
  - sommet : l'hôtel ou un site touristique
  - arête : l'existence de route entre les deux (2) endroits étiquetée par la longueur de cette route.

Exercice 4 – Suite



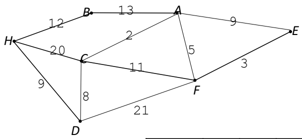
1. Est-il possible pour ce guide, à partir de l'hôtel, d'emprunter toutes les routes (une seule et unique fois) ? Justifier. Si oui, où se terminera son parcours ?
2. Est-il possible au guide de visiter tous les sites une seule et unique fois et revenir à l'hôtel ? Justifier.
3. Déterminer le plus court chemin qui le mène de l'hôtel vers n'importe quel site.

Exercice 4 – Solution

1. Est-il possible pour ce guide, à partir de l'hôtel, d'emprunter toutes les routes (une seule et unique fois) ? Justifier. Si oui, où se terminera son parcours ?

- Emprunter toutes les routes revient à utiliser toutes les arêtes ⇒ former une chaîne
- Utiliser chaque route une seule et unique fois ⇒ chaque arête apparaît exactement une fois ⇒ Chaine élémentaire passant par toutes les arêtes ⇒ Chaîne Eulérienne.
- Selon le théorème d'Euler, G admet une chaine Eulérienne ssi il est connexe (à des sommets isolés près) et le nombre de sommets de degrés impairs est 0 ou 2.

Exercice 4 – Solution

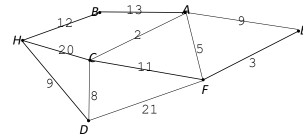


- *G* est connexe.
- Calculons les degrés :

x	*A*	*B*	*C*	*D*	*E*	*F*	*H*
dG(x)	4	2	4	3	2	4	3

- 2 sommets de degrés impairs (*D* et *H*)
- *G* admet une chaîne Eulérienne
- Il est possible de réaliser un tel parcours en démarrant de l'hôtel (*H*) car son degré est impair.
- L'arrivée est au niveau de l'autre sommet de degré impair, c'est-à-dire *D*.

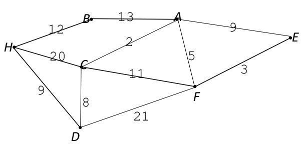
Exercice 4 – Solution



2. Est-il possible au guide de visiter tous les sites une seule et unique fois et revenir à l'hôtel ? Justifier.

- Visiter tous les sites une seule et unique fois ⇒ passer par tous les sommets une seule et unique fois ⇒ Former une chaîne Hamiltonienne
- Revenir au point de départ ⇒ Former un cycle Hamiltonien
- Oui, c'est possible
- Le cycle Hamiltonien est : *H B A E F C D H*

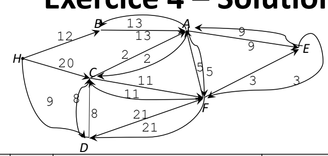
Exercice 4 – Solution



3. Déterminer le plus court chemin qui le mène de l'hôtel vers n'importe quel site.

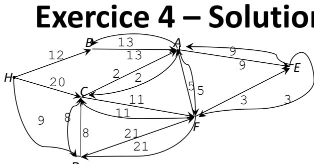
- Le graphe n'est pas orienté, on doit l'orienter
- On remplace chaque arête *{i,j}* de poids *p* par 2 arc (*i,j*) de poids *p* et (*j,i*) de poids *p*.
- Pour optimiser et vu que le départ c'est le *H*, on peut ne pas représenter les arcs entrants vers *H*.
- Le graphe contient des circuits ⇒ Algorithme de Dijkstra

Exercice 4 – Solution



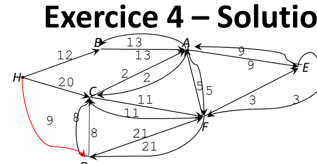
k	*x*	*(x,y)*	*π*							*PRE*						
			A	*B*	*C*	*D*	*E*	*F*	*H*	*A*	*B*	*C*	*D*	*E*	*F*	*H*
			∝	∝	∝	∝	∝	∝	0	-	-	-	-	-	-	-
1	*H*															
2																
3																
4																
5																
6																

Exercice 4 – Solution



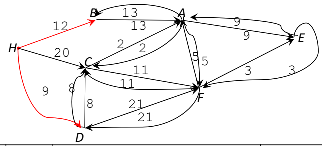
k	*x*	*(x,y)*	*π*							*PRE*						
			A	*B*	*C*	*D*	*E*	*F*	*H*	*A*	*B*	*C*	*D*	*E*	*F*	*H*
			∝	~~∝~~	~~∝~~	~~∝~~	∝	∝	0	-	-	-	-	-	-	-
1	*H*	*(H,B)* *(H,C)* *(H,D)*		12	20	9					*H*	*H*	*H*			

Exercice 4 – Solution



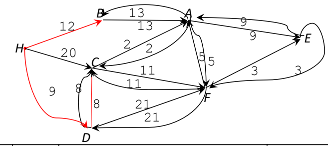
k	*x*	*(x,y)*	*π*							*PRE*						
			A	*B*	*C*	*D*	*E*	*F*	*H*	*A*	*B*	*C*	*D*	*E*	*F*	*H*
			∝	~~∝~~	~~∝~~	~~∝~~	∝	~~∝~~	0	-	-	-	-	-	-	-
1	*H*	*(H,B)* *(H,C)* *(H,D)*		12	~~20~~	9					*H*	~~*H*~~	*H*			
2	*D*	*(D,C)* *(D,F)*			17			30				*D*			*D*	

Exercice 4 – Solution



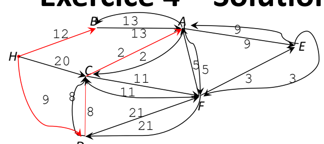
k	x	(x,y)	π							PRE						
			A	B	C	D	E	F	H	A	B	C	D	E	F	H
			∝	~~∝~~	~~∝~~	~~∝~~	∝	~~∝~~	0	-	-	-	-	-	-	-
1	H	(H,B) (H,C) (H,D)		12	~~20~~	9					H	~~H~~	H			
2	D	(D,C) (D,F)			17			30				D			D	
3	B	(B,A)	25							B						

Exercice 4 – Solution



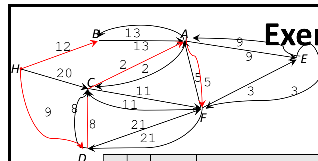
k	x	(x,y)	π							PRE						
			A	B	C	D	E	F	H	A	B	C	D	E	F	H
			∝	~~∝~~	~~∝~~	~~∝~~	∝	~~∝~~	0	-	-	-	-	-	-	-
1	H			12	~~20~~	9					H	~~H~~	H			
2	D	(D,C) (D,F)			17			~~30~~				D			~~D~~	
3	B	(B,A)	~~25~~							~~B~~						
4	C	(C,A) ~~(C,D)~~ (C,F)	19					28		C					C	

Exercice 4 – Solution



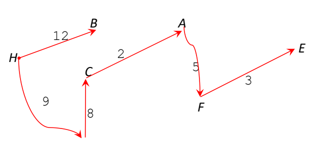
k	x	(x,y)	π							PRE						
			A	B	C	D	E	F	H	A	B	C	D	E	F	H
			∝	~~∝~~	~~∝~~	~~∝~~	~~∝~~	~~∝~~	0	-	-	-	-	-	-	-
1	H			12	~~20~~	9					H	~~H~~	H			
2	D				17			~~30~~				D			~~D~~	
3	B	(B,A)	~~25~~							~~B~~						
4	C		19					~~28~~		C					~~C~~	
5	A	~~(A,B)~~ ~~(A,C)~~ (A,E) (A,F)					28	24						A	A	

Exercice 4 – Solution



k	x	(x,y)	π							PRE						
			A	B	C	D	E	F	H	A	B	C	D	E	F	H
			∝	~~∝~~	~~∝~~	~~∝~~	~~∝~~	~~∝~~	0	-	-	-	-	-	-	-
1	H			12	~~20~~	9					H	~~H~~	H			
2	D				17			~~30~~				D			~~D~~	
3	B	(B,A)	~~25~~							~~B~~						
4	C		19					~~28~~		C					~~C~~	
5	A						~~28~~	24						~~A~~	A	
6	F	~~(F,A)~~ ~~(F,C)~~ ~~(F,D)~~ (F,E)					27							F		

Exercice 4 – Solution



π							PRE						
A	B	C	D	E	F	H	A	B	C	D	E	F	H
19	12	17	9	27	24	0	C	H	D	H	F	A	-

Exercice 5

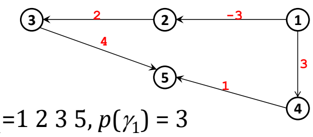
- L'algorithme de *Dijkstra* n'admet pas qu'il y ait des poids négatifs dans le graphe.
- Si le poids minimal est $-k$, où $k > 0$
- On rajoute $+k$ au poids de chaque arc afin de les rendre tous positifs.
- Est-il possible d'utiliser ce raisonnement pour résoudre le problème du chemin de poids optimal à l'aide de l'algorithme de *Dijkstra* dans des graphes ayant des poids négatifs.

Exercice 5 – Solution

- Soit un chemin γ_1 allant de x vers y de longueur l_1 et de poids p_1
- Soit un chemin γ_2 allant de x vers y de longueur l_2 et de poids p_2
- Après rajout de $+k$ à chaque arc, les poids changent :
  - $p(\gamma_1)= p_1 + (l_1 * k)$
  - $p(\gamma_2)= p_2 + (l_2 * k)$
- Les poids vont changer selon la longueur des chemins et il est possible qu'un chemin optimal ne le soit plus à cause de la longueur.

Exercice 5 – Solution

- Contre-exemple



γ_1=1 2 3 5, $p(\gamma_1) = 3$

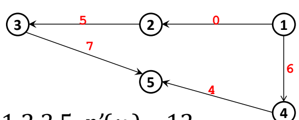
γ_2=1 4 5, $p(\gamma_2) = 5$

$p(\gamma_1) < p(\gamma_2) \Rightarrow \gamma_1$ est meilleur

- On rajoute +3 aux poids de tous les arcs

Exercice 5 – Solution

- Contre-exemple



γ_1=1 2 3 5, $p'(\gamma_1) = 12$

γ_2=1 4 5, $p'(\gamma_2) = 10$

$p(\gamma_2) < p(\gamma_1) \Rightarrow \gamma_2$ est devenu meilleur

Chapitre 5 : Ordonnancement
Série d'exercices de TD avec Corrigé

Présenté par :
H. BENKAOUHA
Bureau 222, Faculté d'Informatique, USTHB
haroun.benkaouha@usthb.edu.dz
haroun.benkaouha@gmail.com

Exercice 1

- Soit un projet constitué des huit (8) tâches décrites dans le tableau ci-dessous :

N° Tâche	Durée en jours	Tâches précédentes
1	2	-
2	3	-
3	7	-
4	4	2
5	10	1
6	6	1, 4
7	5	1, 3
8	2	5, 6, 7

Exercice 1 - Suite

1. Modéliser le problème sous forme d'un graphe potentiel tâches (MPM).
2. Calculer les dates au plus tôt et les dates au plus tard de chaque tâche.
3. Calculer la marge totale pour chaque tâche et déduire les tâches critiques ainsi que le chemin critique.

Exercice 1 - Solution

$t_4 - t_2 \geq 3$
$t_5 - t_1 \geq 2$
$t_6 - t_1 \geq 2$
$t_6 - t_4 \geq 4$
$t_7 - t_1 \geq 2$
$t_7 - t_3 \geq 7$
$t_8 - t_5 \geq 10$
$t_8 - t_6 \geq 6$
$t_8 - t_7 \geq 5$

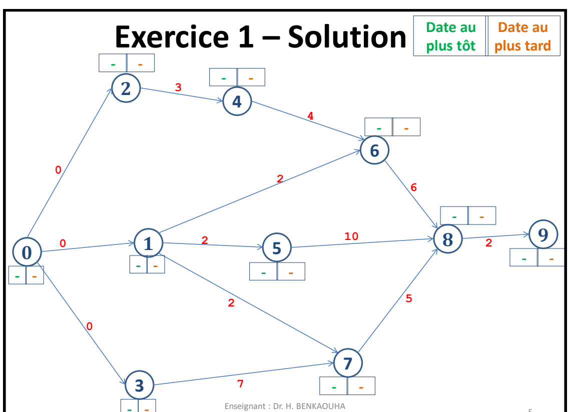
Tâche fictive de début : 0

$t_1 - t_0 \geq 0$
$t_2 - t_0 \geq 0$
$t_3 - t_0 \geq 0$

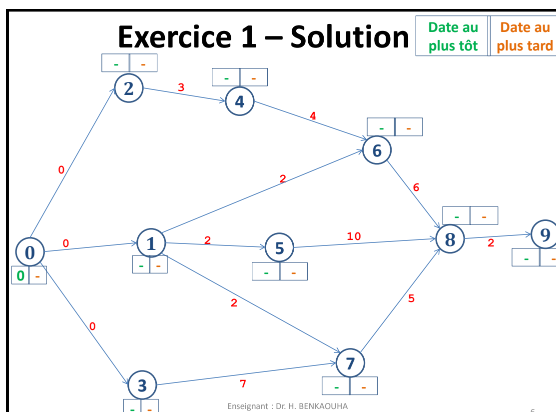
Tâche fictive de fin : 9

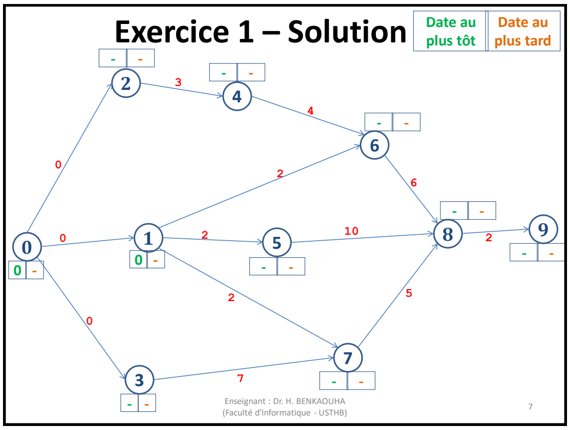
$t_9 - t_i \geq d_i \;\; \forall i$ de 1 à 8

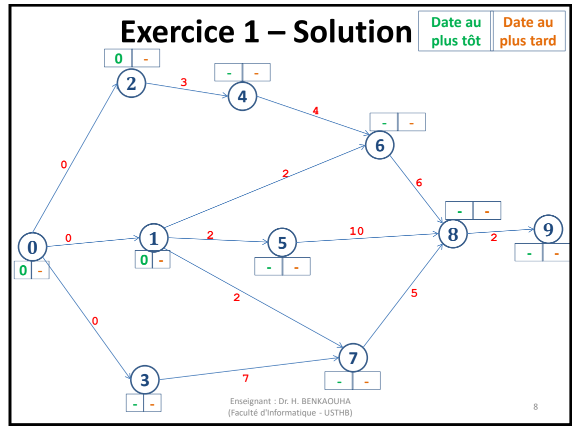
Exercice 1 – Solution

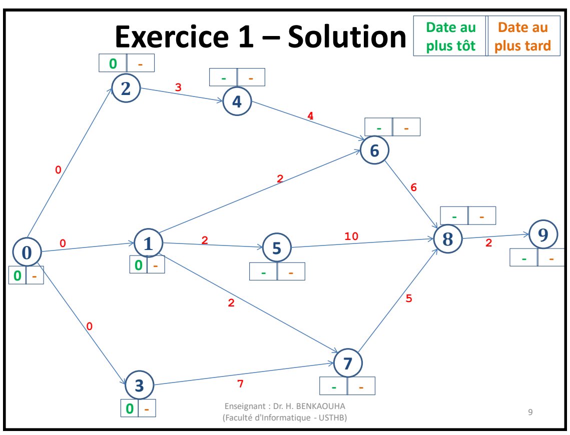


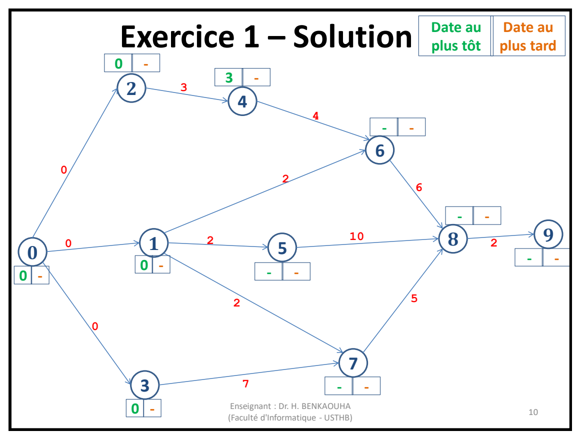
Exercice 1 – Solution

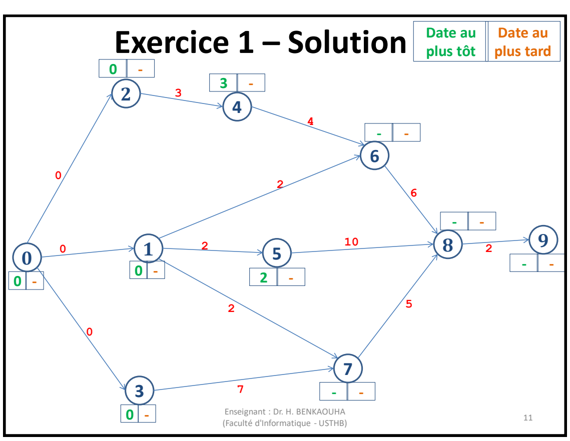


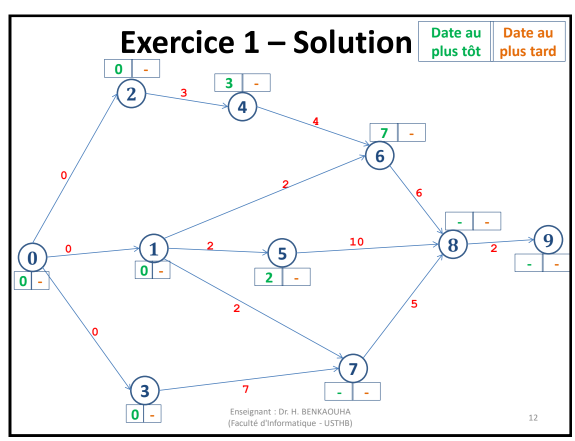

Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
7


Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
8


Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
9

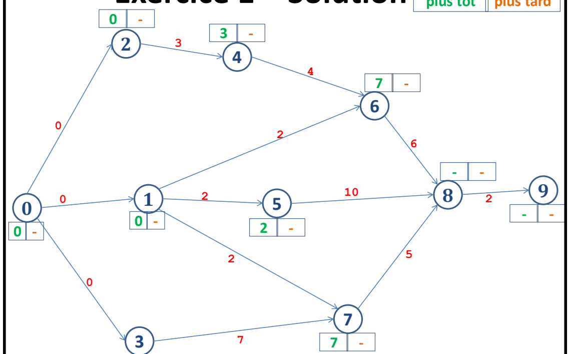

Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
10


Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
11


Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
12

Exercice 1 – Solution

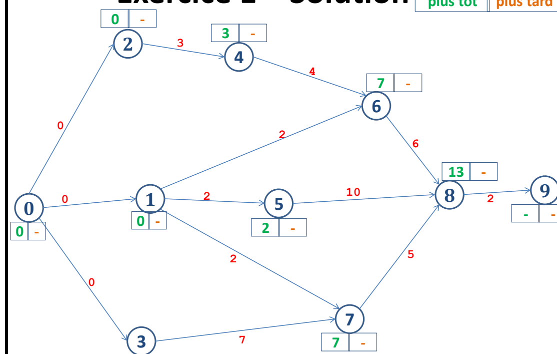
Date au plus tôt | Date au plus tard



Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 13

Exercice 1 – Solution

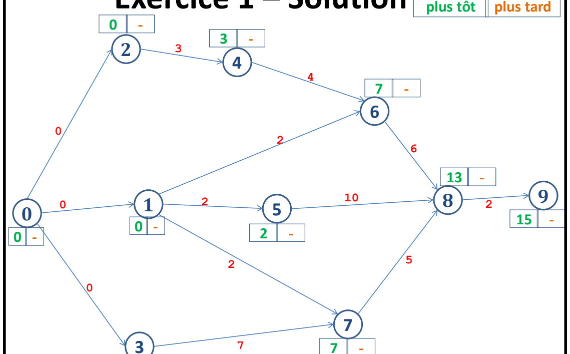
Date au plus tôt | Date au plus tard



Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 14

Exercice 1 – Solution

Date au plus tôt | Date au plus tard



Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 15

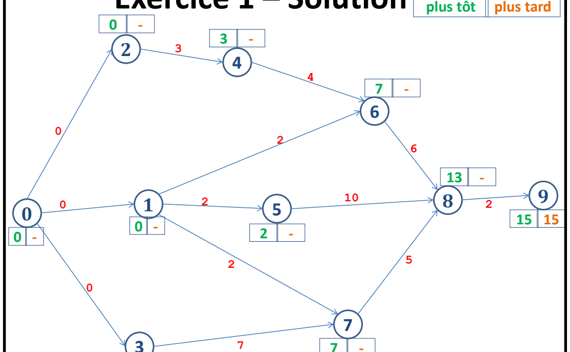
Execice 1 - Solution

Tâches	1	2	3	4	5	6	7	8
Dates au plus tôt	0	0	0	3	2	7	7	13

Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 16

Exercice 1 – Solution

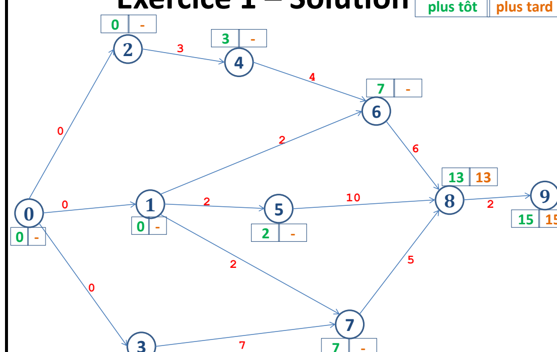
Date au plus tôt | Date au plus tard



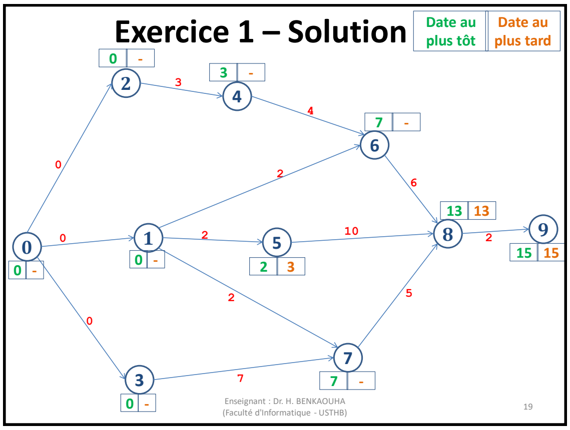
Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 17

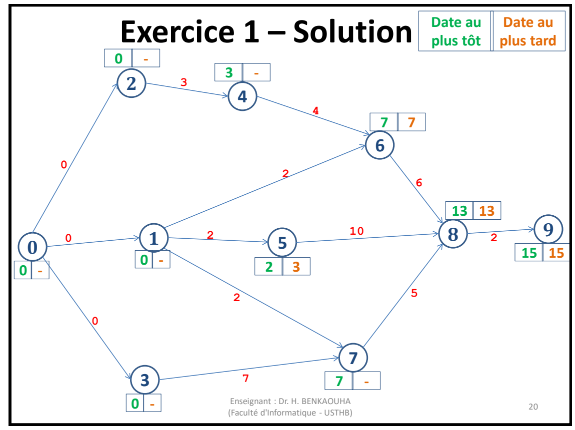
Exercice 1 – Solution

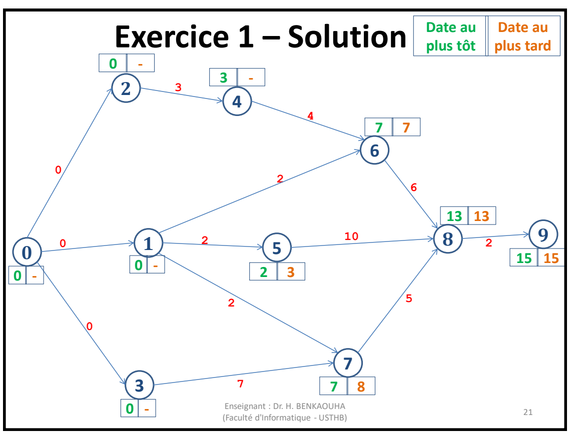
Date au plus tôt | Date au plus tard

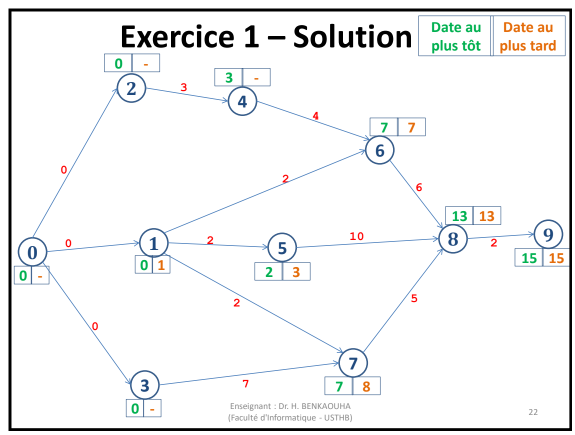


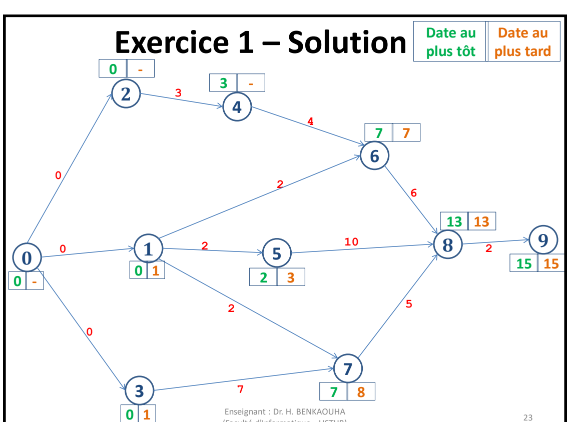
Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 18

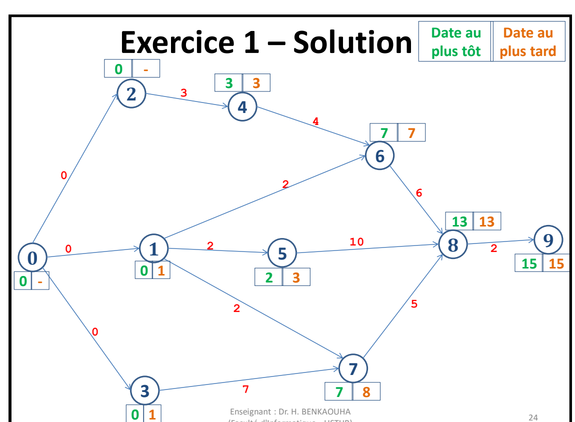

Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
19


Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
20

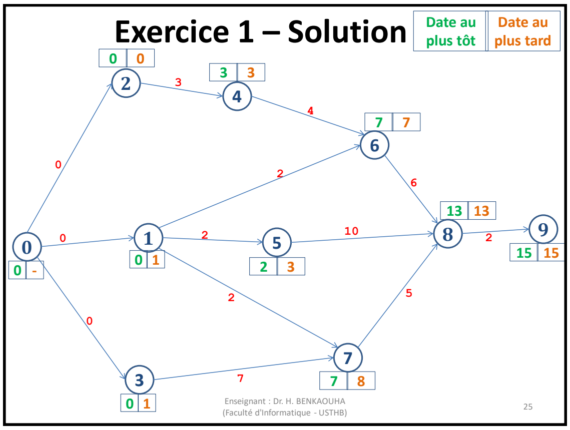

Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
21


Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
22


Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
23

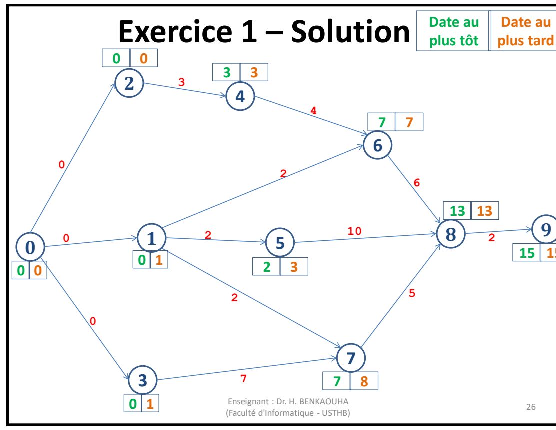

Exercice 1 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
24

Exercice 1 – Solution



Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
25

Exercice 1 – Solution



Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
26

Execice 1 - Solution

Tâches	1	2	3	4	5	6	7	8
Dates au plus tard	1	0	1	3	3	7	8	13

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
27

Execice 1 - Solution

Tâches	1	2	3	4	5	6	7	8
Dates au plus tôt	0	0	0	3	2	7	7	13
Dates au plus tard	1	0	1	3	3	7	8	13
Marge totale	1	0	1	0	1	0	1	0

- Les tâches critiques : **2**, **4**, **6** et **8**
- Le chemin critique : **0 2 4 6 8 9**

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
28

Exercice 2

- Un projet requiert la réalisation de huit (08) activités, le tableau suivant donne pour chaque activité, le temps (en jours) requis et les activités pré-requises.

Activité	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***
Durée	7	4	4	5	6	8	4	6
Activités requises			*B*	*A, C*	*B*	*D, E*	*B*	*D, E, G*

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
29

Exercice 2 - Suite

1. Donner la représentation du problème en graphe MPM (Potentiel-tâches).
2. Donner les dates de début au plus tôt de chaque tâche et la durée optimale du projet.
3. Donner les dates au plus tard, et déduire les taches critiques.
4. Si la tâche *E* commence avec 03 jours de retard et elle dure une journée de plus que prévu, quel est alors l'impact sur la durée optimale du projet ?

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
30

Exercice 2 - Solution

$t_C - t_B \geq 4$
$t_D - t_A \geq 7$
$t_D - t_C \geq 4$
$t_E - t_B \geq 4$
$t_F - t_D \geq 5$
$t_F - t_E \geq 6$
$t_G - t_B \geq 4$
$t_H - t_D \geq 5$
$t_H - t_E \geq 6$
$t_H - t_G \geq 4$

Tâche fictive de début : α

$t_A - t_\alpha \geq 0$
$t_B - t_\alpha \geq 0$

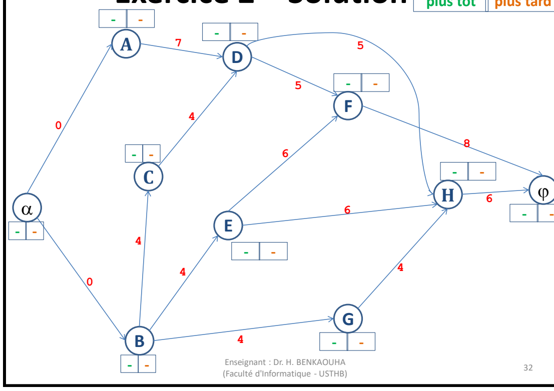
Tâche fictive de fin : φ

$t_\varphi - t_i \geq d_i \quad \forall i$ de A à H

Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 31

Exercice 2 – Solution

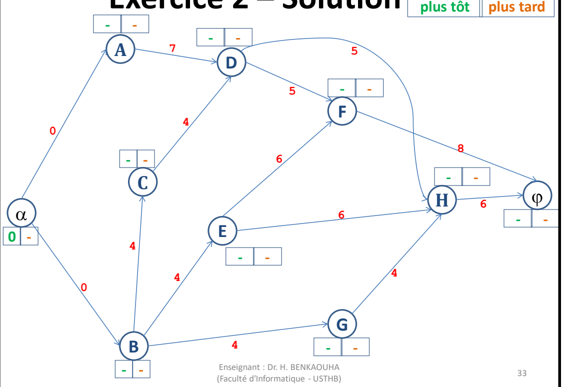
Date au plus tôt | Date au plus tard



Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 32

Exercice 2 – Solution

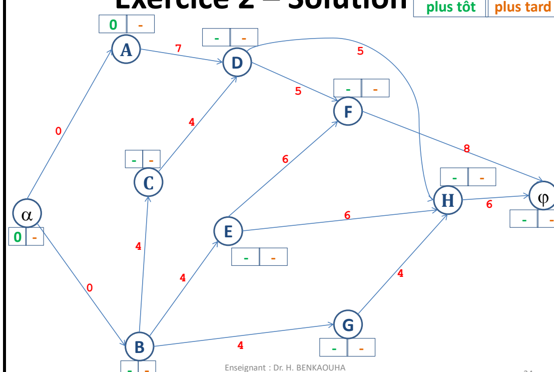
Date au plus tôt | Date au plus tard



Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 33

Exercice 2 – Solution

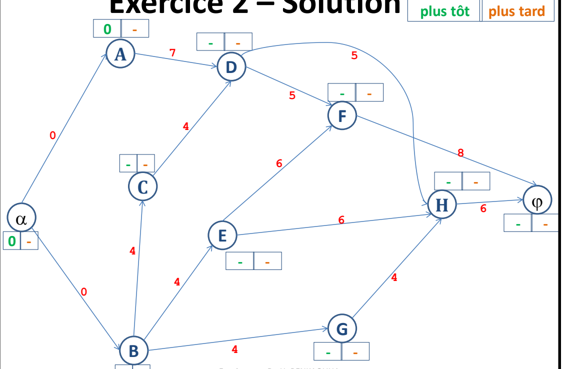
Date au plus tôt | Date au plus tard



Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 34

Exercice 2 – Solution

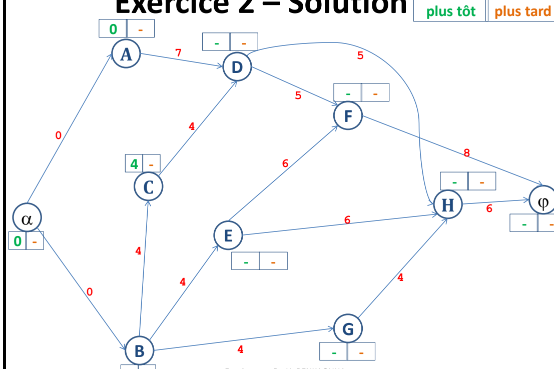
Date au plus tôt | Date au plus tard



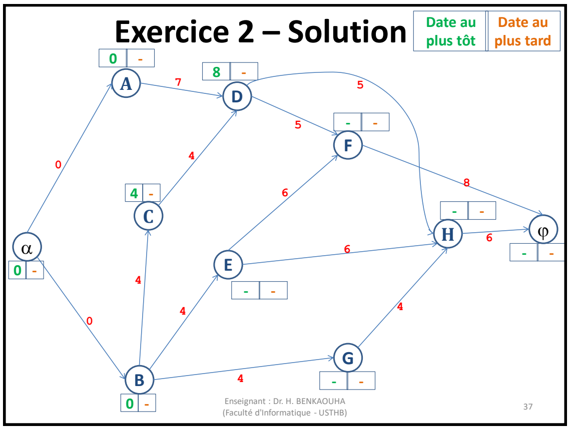
Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 35

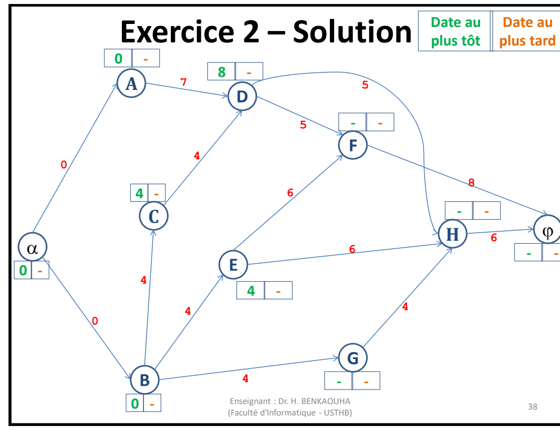
Exercice 2 – Solution

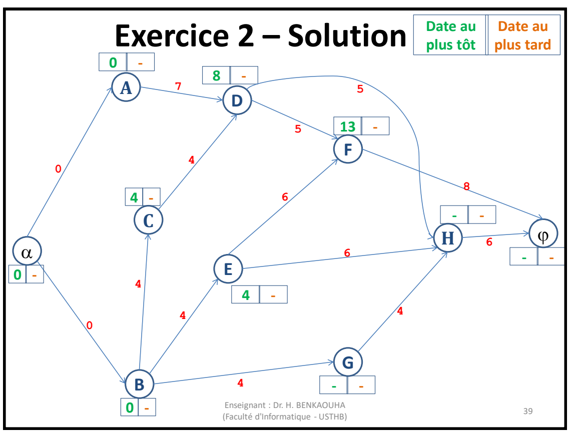
Date au plus tôt | Date au plus tard

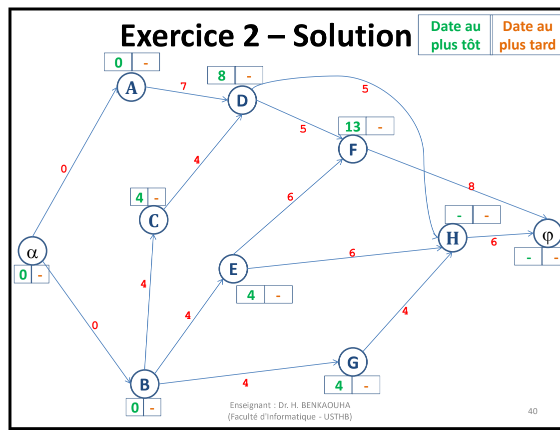


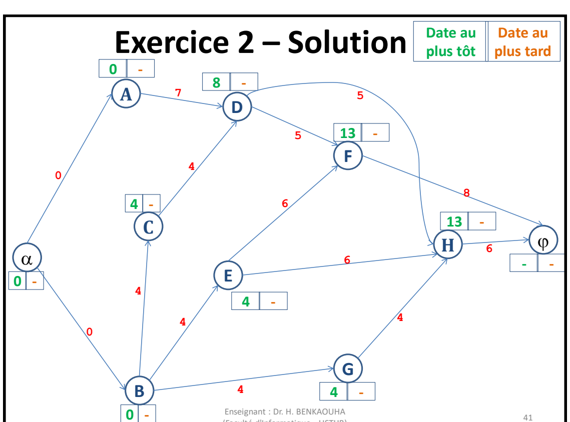
Enseignant : Dr. H. BENKAOUHA (Faculté d'Informatique - USTHB) 36

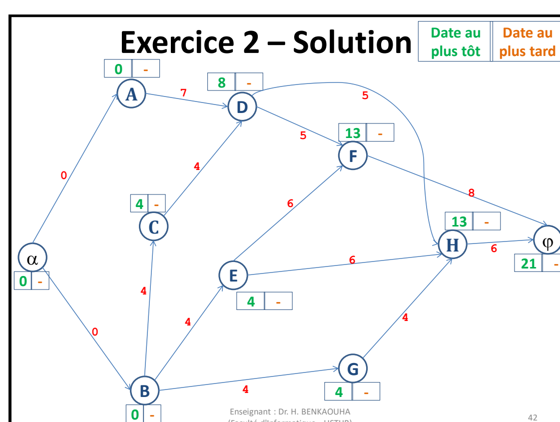

Exercice 2 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
37


Exercice 2 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
38


Exercice 2 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
39


Exercice 2 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
40


Exercice 2 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
41

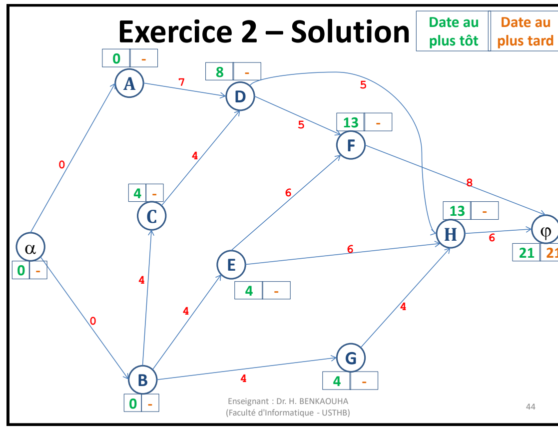

Exercice 2 – Solution
Date au plus tôt
Date au plus tard
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
42

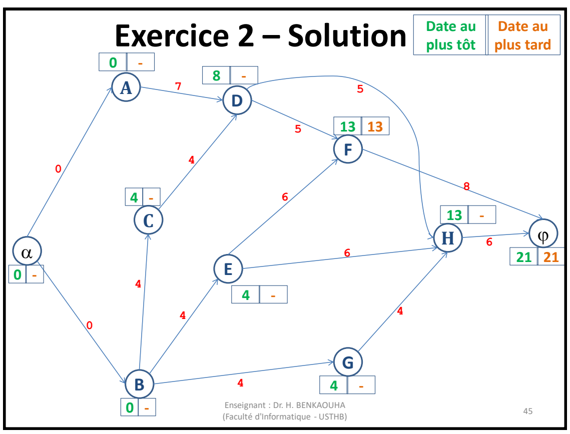
Execice 2 - Solution

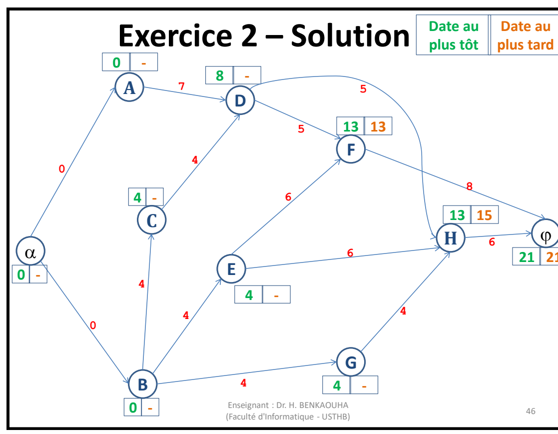
Tâches	A	B	C	D	E	F	G	H
Dates au plus tôt	0	0	4	8	4	13	4	13

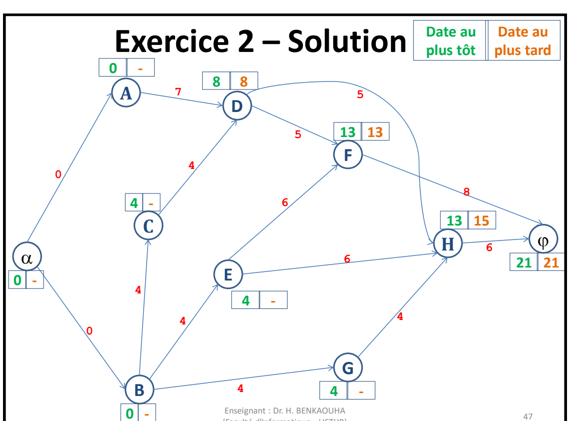
- Durée optimale du projet : 21 jours

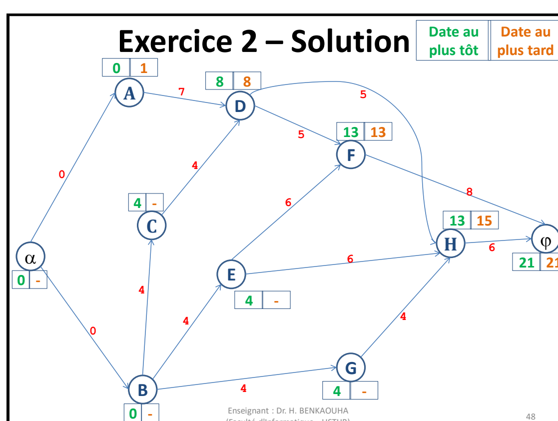
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 43



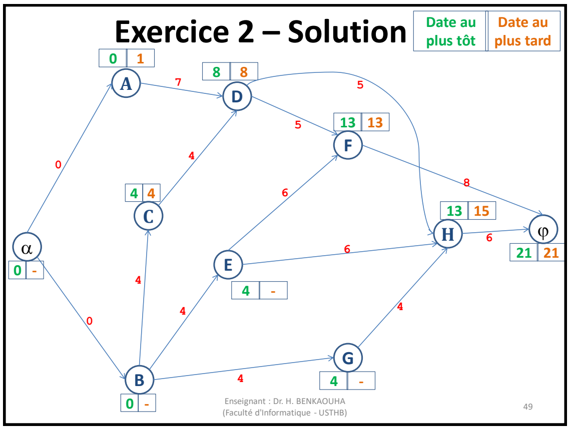




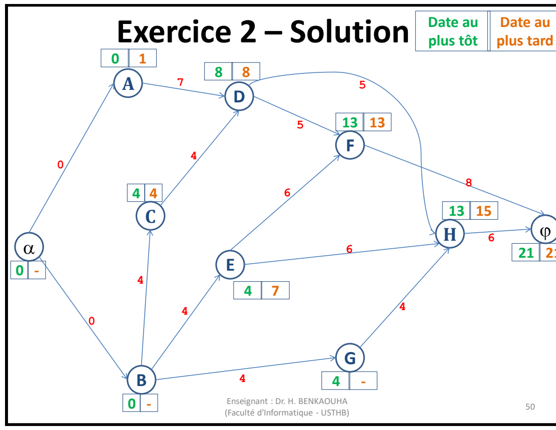




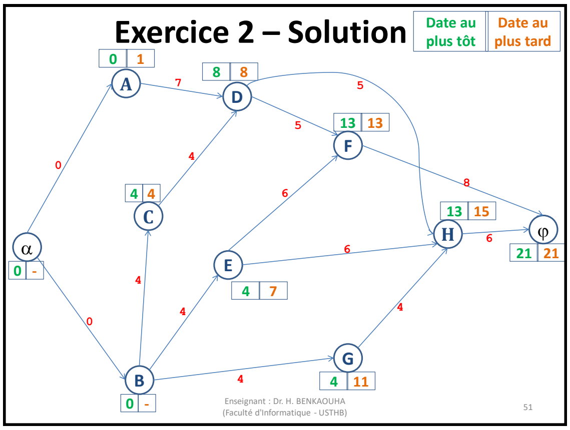
Exercice 2 – Solution



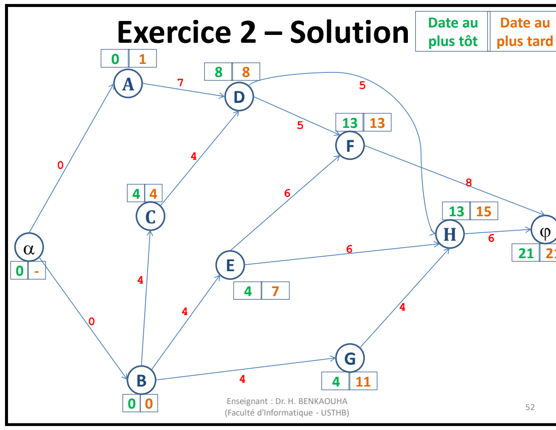
Exercice 2 – Solution



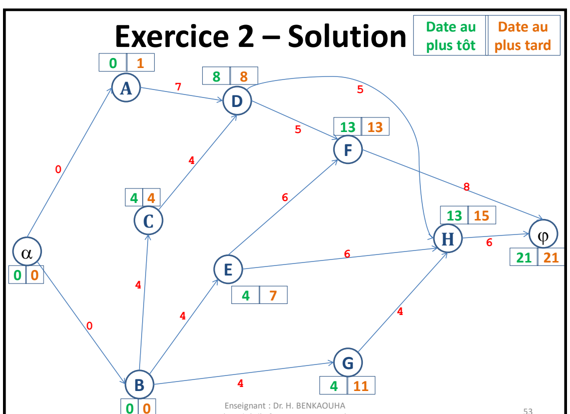
Exercice 2 – Solution



Exercice 2 – Solution



Exercice 2 – Solution



Execice 2 - Solution

Tâches	A	B	C	D	E	F	G	H
Dates au plus tard	1	0	4	8	7	13	11	15

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 54

Execice 2 - Solution

Tâches	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*
Dates au plus tôt	0	0	4	8	4	13	4	13
Dates au plus tard	1	0	4	8	7	13	11	15
Marge totale	1	0	0	0	3	0	7	2

- Les tâches critiques : **B**, **C**, **D** et **F**
- Le chemin critique : $\boldsymbol{\alpha}$**BCDF**$\boldsymbol{\varphi}$

Execice 2 - Solution

Tâches	*E*
Dates au plus tôt	4
Dates au plus tard	7
Marge totale	3

- *E* démarre 3 jours en retard.
- *E* dure 1 jour de plus.
- Retard au total de 4 jours pour *E*
- Marge totale = 3
- Retard = Marge totale + 1 $\Rightarrow$ Durée optimale du projet retardée d'une journée.

Exercice 3

- Un laboratoire doit effectuer une étude comprenant deux (02) groupes de travaux distincts *A* et *B*. *A* désigne les travaux de recherche et d'études préliminaires et *B* les travaux d'exécution. On se propose de minimiser la durée totale de cette étude.
- Les effectifs *nA* et *nB* affectés respectivement à *A* et *B* sont compris entre les limites :
- $3 \leq nA \leq 6$
- $6 \leq nB \leq 15$

Exercice 3 – Suite

- Par ailleurs, les durées de *A* et *B* sont respectivement estimées, en jours, à 600 /*nA* et 300 /*nB*.
- Il faut aussi signaler que *B* doit débuter au plus tôt à la date 10, et après que la moitié des travaux *A* soit accomplie. Il faut aussi s'assurer que la tâche *B* soit terminée avant que la tâche *A* ne soit accomplie.

1. Quelle est la condition pour que le problème soit réalisable ?
2. Quelle est l'influence de *n* sur la durée minimale de réalisation de l'étude ?

Exercice 3 – Solution

- Modélisation :

Tâche fictive de début : α

$t_A - t_\alpha \geq 0$

$t_B - t_\alpha \geq 10$

$t_B \geq t_A + (600 /nA)/2$

$\Rightarrow t_B - t_A \geq 300 /nA$

$t_B + (300 /nB) \leq t_A + (600 /nA)$

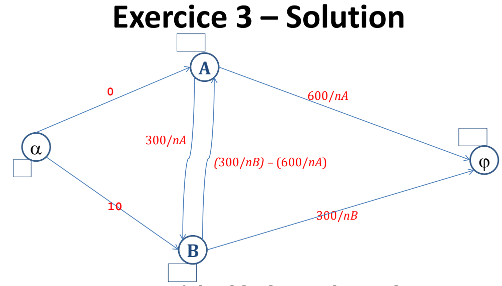
$\Rightarrow t_A - t_B \geq (300 /nB) - (600 /nA)$

Tâche fictive de fin : φ

$t_\varphi - t_A \geq 600 /nA$

$t_\varphi - t_B \geq 300 /nB$

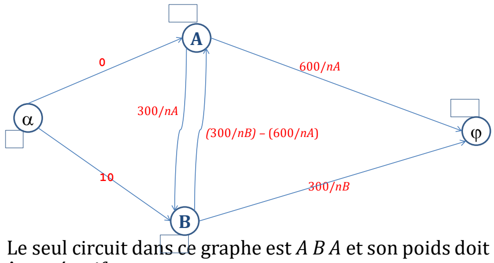
Exercice 3 – Solution



- Pour que ça soit réalisable, le graphe ne doit pas contenir de circuit absorbant
- Ici, le circuit absorbant est un circuit de poids >0

Enseignant : Dr. H. BENKAOUHA
(haroun.benkaouha@usthb.edu.dz)

Exercice 3 – Solution



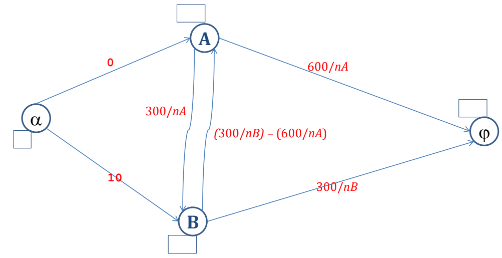
- Le seul circuit dans ce graphe est $A\ B\ A$ et son poids doit être négatif.

$p(A\ B\ A) = 300/nA + (300/nB) - (600/nA) \leq 0$

$\Rightarrow (300/nB) - (300/nA) \leq 0 \Rightarrow nA \leq nB$

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 61

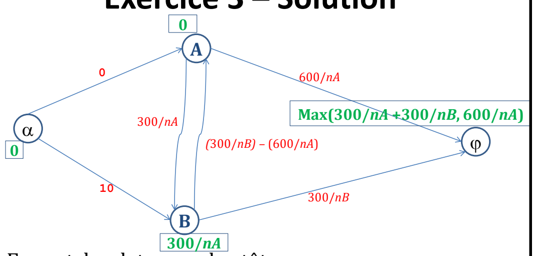
Exercice 3 – Solution



- Ceci est vérifié selon l'énoncé.
- Donc le problème est réalisable.

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 62

Exercice 3 – Solution



- En vert, les dates au plus tôt
- Pour la tâche A c'est 0 car $(300/nB)-(600/nA) < 0$
- Pour la tâche B c'est $300/nA$ car $300/nA>10$
- $nA \leq 6 \Rightarrow 1/nA \geq 1/6 \Rightarrow 300/nA \geq 300/6=50>10$

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 63

Exercice 3 – Solution

- La durée optimale du projet correspond à la date au plus tôt de la tâche fictive de fin de projet = $\text{Max}(300/nA +300/nB, 600/nA)$
- Comparons ces 2 valeurs
- $600/nA > 300/nA +300/nB$

$\Rightarrow 300/nA > 300/nB \Rightarrow 1/nA > 1/nB$

$\Rightarrow nA < nB$

- Or on a (selon l'énoncé) $3 \leq nA \leq 6$ et $6 \leq nB \leq 15 \Rightarrow nA \leq nB$
- Donc la durée du projet est $600/nA$ sauf si $nA = nB$ on aura $600/nA = 300/nA +300/nB$
- ***Conclusion :*** nB n'a pas d'effet sur la durée optimale du projet et nA doit être maximal c'est-à-dire nA*=6.*

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB) 64

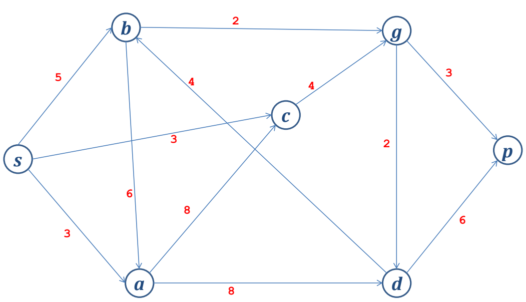
Chapitre 6 : Les flots
Série d'exercices de TD avec Corrigé

Présenté par :
H. BENKAOUHA
Bureau 222, Faculté d'Informatique, USTHB
haroun.benkaouha@usthb.edu.dz
haroun.benkaouha@gmail.com

Exercice 1

- Soit le réseau de transport ci-dessous (entre parenthèses les capacités des arcs). Trouver le flot maximal. :

Exercice 1 – Suite

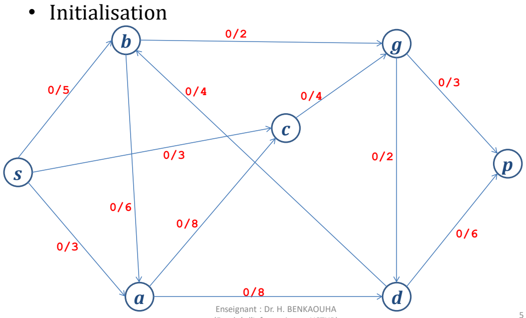


Exercice 1 – Solution (1/11)

- Le réseau donné est un réseau de transport car :
  - Il s'agit d'un graphe orienté valué
  - Tous les poids (capacités) sont ≥ 0
  - Il y a une seule source *s* (entrée du réseau)
  - Il y a un seul puits *p* (sortie du réseau)
- Pour appliquer l'algorithme de Ford-Fulkerson afin de trouver le flot maximal, il nous faut un flot initial compatible.
- Prenons le flot 0 partout.

Exercice 1 – Solution (2/11)

- Initialisation

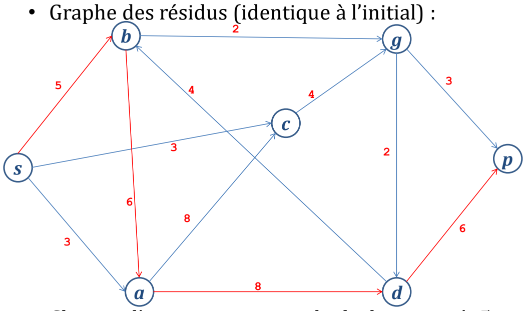


Exercice 1 – Solution (3/11)

- On aurait pu initialiser à un flot compatible différent de 0.
- Ça se fait par affectation avec tâtonnement.
- Il faut vérifier que :
  - La somme des flux sortant de l'entrée = la somme des flux entrant à la sortie.
  - Pour chaque autre sommet : la somme des flux entrants = la somme des flux sortants.

Exercice 1 – Solution (4/11)

- Graphe des résidus (identique à l'initial) :

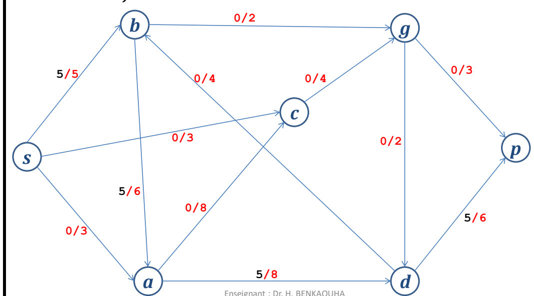


- Chemin d'augmentation: γ=*sbadp* de capacité=5

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
7

Exercice 1 – Solution (5/11)

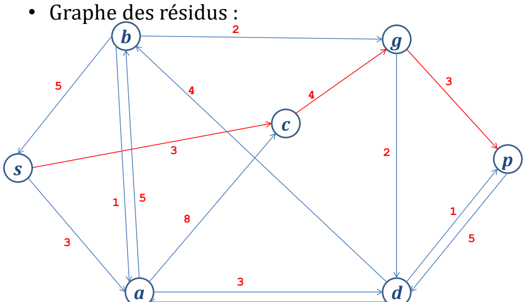
- Mise à jour du flot



Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
8

Exercice 1 – Solution (6/11)

- Graphe des résidus :

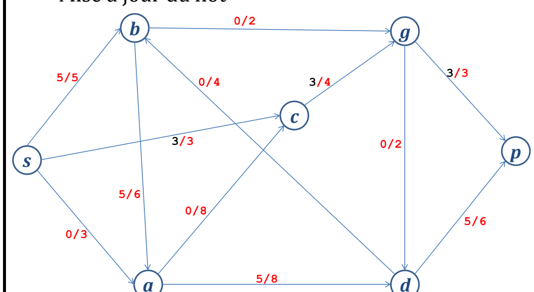


- Chemin d'augmentation: γ=*scgp* de capacité=3

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
9

Exercice 1 – Solution (7/11)

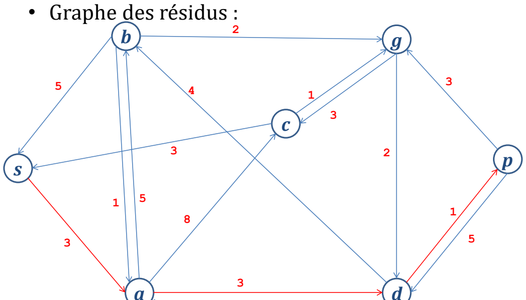
- Mise à jour du flot



Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
10

Exercice 1 – Solution (8/11)

- Graphe des résidus :

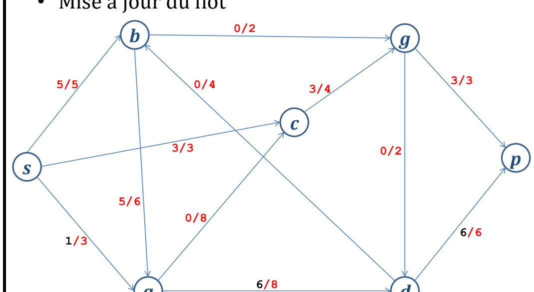


- Chemin d'augmentation: γ=*sadp* de capacité=1

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
11

Exercice 1 – Solution (9/11)

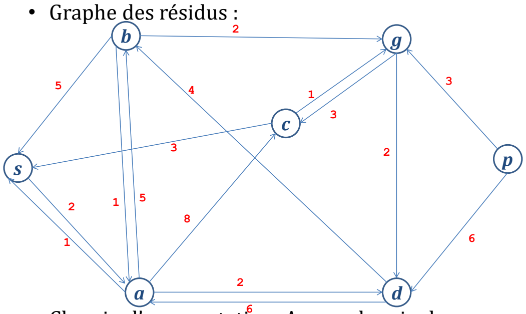
- Mise à jour du flot



Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatique - USTHB)
12

Exercice 1 – Solution (10/11)

- Graphe des résidus :



- Chemin d'augmentation: Aucun chemin de s vers p

Exercice 1 – Solution (11/11)

- Le flot obtenu est maximal (sortant de l'entrée s = entrant à la sortie p = 9)
- Remarques :
  - La solution obtenue n'est pas nécessairement la seule et unique.
  - Il est possible de trouver d'autres solutions avec un flot de 9.
  - Ce qui fait varier la solution optimale est : le flot initial et le choix du chemin d'augmentation à chaque itération.

Exercice 2

- Une ville F est alimentée en eau grâce à des réservoirs situés dans 3 villes (A, B et C).
- Chaque réservoir est alimenté à partir de différentes sources (nappes souterraines, châteaux d'eau, ...) comme suit :
  - 10000 $m^3/jour$ pour A et C
  - 1 000 $m^3/jour$ pour B.

Exercice 2 – Suite

- Le réseau de distribution reliant la ville F aux réservoirs passe par plusieurs points qui sont reliés entres eux à travers des canalisations de différentes capacités selon le tableau suivant :

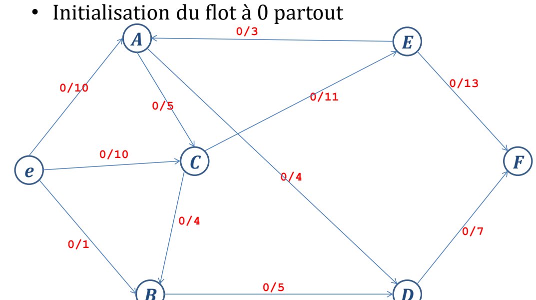
Point de départ	A	A	B	C	C	D	E	E
Point d'arrivée	C	D	D	B	E	F	A	F
Capacité du canal (en milliers de m^3)	2	4	5	4	11	7	3	13

Exercice 2

- Modélisation :
  - Par un réseau de transport
  - Chaque ville représenté par un sommet A à F.
  - On rajoute un sommet e représentant les différentes sources. C'est l'entrée du réseau.
  - F est la sortie du réseau.
  - Chaque canalisation est représentée par un arc.
  - La capacité de la canalisation représente la capacité (poids) de l'arc correspondant. L'unité est $1000m^3$.

Exercice 2 – Solution (1/13)

- Initialisation du flot à 0 partout

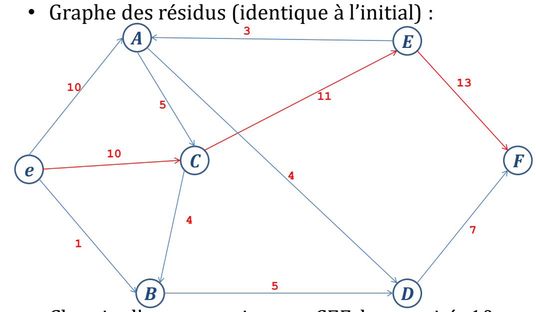


Exercice 2 – Solution (2/13)

- On aurait pu initialiser à un flot compatible différent de 0.
- Ça se fait par affectation avec tâtonnement.
- Il faut vérifier que :
  - La somme des flux sortant de l'entrée = la somme des flux entrant à la sortie.
  - Pour chaque autre sommet : la somme des flux entrants = la somme des flux sortants.

Exercice 2 – Solution (3/13)

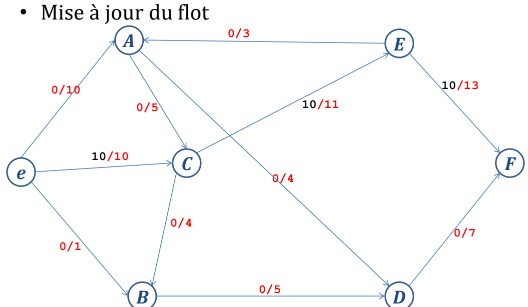
- Graphe des résidus (identique à l'initial) :



- Chemin d'augmentation: γ=*eCEF* de capacité=10

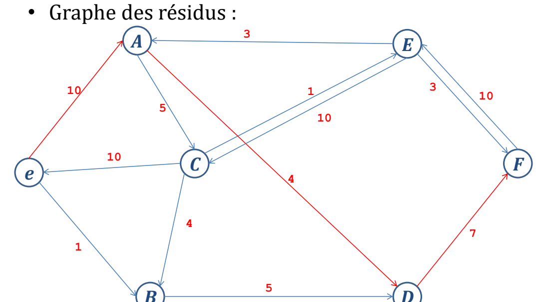
Exercice 2 – Solution (4/13)

- Mise à jour du flot



Exercice 2 – Solution (5/13)

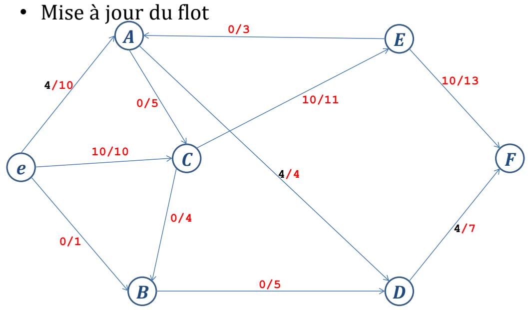
- Graphe des résidus :



- Chemin d'augmentation: γ=*eADF* de capacité=4

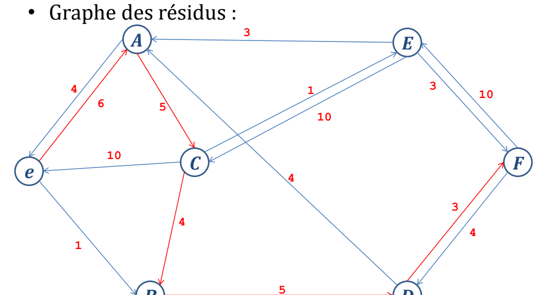
Exercice 2 – Solution (6/13)

- Mise à jour du flot



Exercice 2 – Solution (7/13)

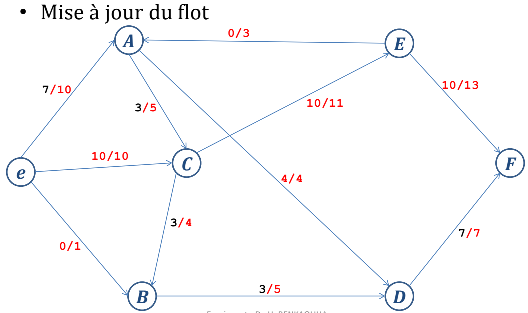
- Graphe des résidus :



- Chemin d'augmentation: γ=*eACBDF* de capacité=3

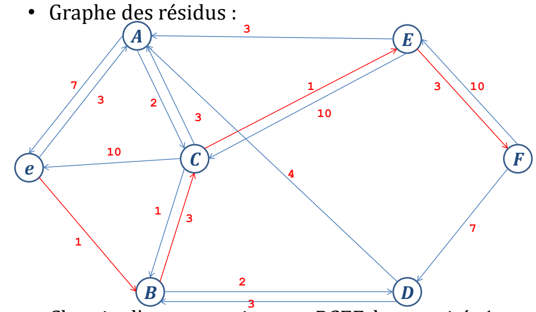
Exercice 2 – Solution (8/13)

- Mise à jour du flot



Exercice 2 – Solution (9/13)

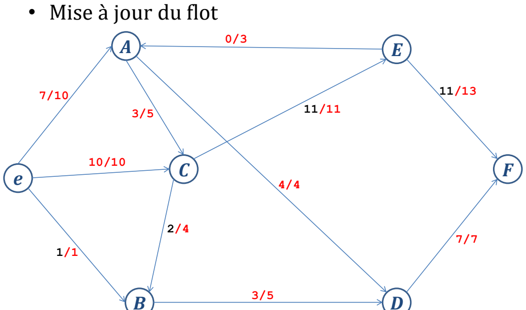
- Graphe des résidus :



- Chemin d'augmentation: γ=*eBCEF* de capacité=1

Exercice 2 – Solution (10/13)

- Mise à jour du flot

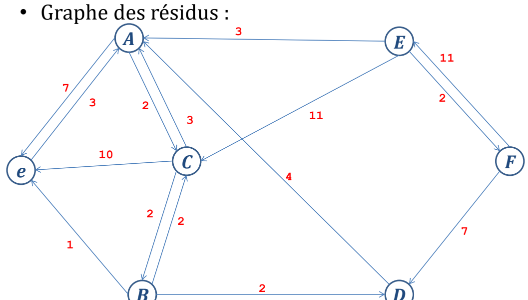


Exercice 2 – Solution (11/13)

- Pour l'arc (C, B)
  - On décrémente le flux de 1
  - Car il est pris dans le sens inverse sur le chemin d'augmentation.

Exercice 2 – Solution (12/13)

- Graphe des résidus :



- Chemin d'augmentation: Aucun chemin de e vers F

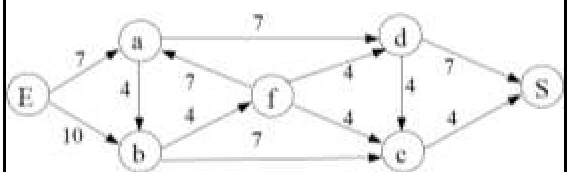
Exercice 2 – Solution (13/13)

- Le flot obtenu est maximal (sortant de l'entrée e = entrant à la sortie F = 18)
- La quantité journalière acheminée vers la ville F est de 18 000 m^3.

Enseignant : Dr. H. BENKAOUHA
(haroun.benkaouha@usthb.edu.dz)

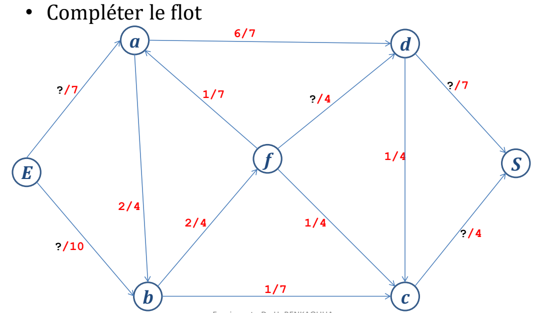
Exercice 3

- Soit le réseau de transport ci-dessous ayant comme entrée (source) le sommet E et comme sortie (puits) le sommet S.



Exercice 3 – Suite

- Compléter le flot



Exercice 3 – Solution (1/13)

- Nous devons résoudre le système d'équations suivant (loi de Kirchhoff au niveau des sommets) :

$c_{Ea} + 1 = c_{ab} + c_{ad}$

$c_{Eb} + 2 = 2 + 1$

$1 + 1 + 1 = c_{cS}$

$6 + c_{fd} = 1 + c_{dS}$

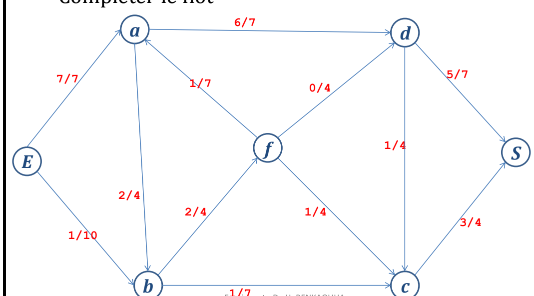
$2 = 1 + 1 + c_{fd}$

- De plus, il faut vérifier entre l'entrée et la sortie :

$c_{Ea} + c_{Eb} = c_{dS} + c_{cS}$

Exercice 3 – Solution (2/13)

- Compléter le flot

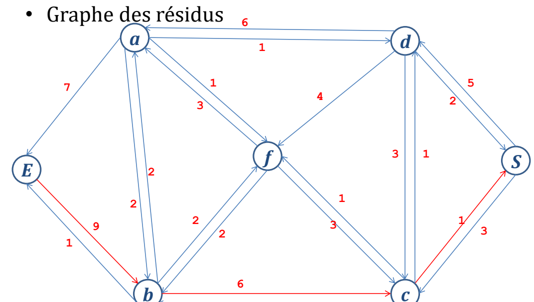


Exercice 3 – Solution (3/13)

- Pour vérifier si le flot est maximal ou non,
  - Nous devons tracer le graphe des résidus
  - Vérifier s'il y a ou non chemin d'augmentation de l'entrée du réseau (E) vers la sortie du réseau (S)

Exercice 3 – Solution (4/13)

- Graphe des résidus



Exercice 3 – Solution (5/13)

- Il y a au moins un chemin d'augmentation.
- Par exemple :
  - γ=*EbcS* de capacité=1
- Donc : Le flot n'est pas maximal.

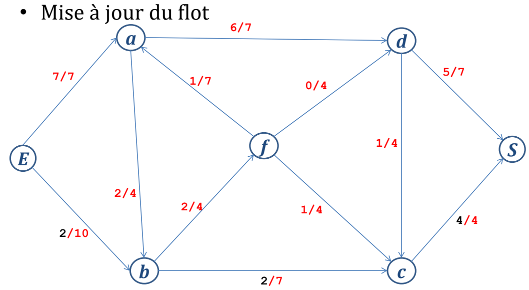
Exercice 3 – Solution (6/13)

- Application de l'algorithme de Ford-Fulkerson
- A partir :
  - Du flot obtenu en *Q*1

  et
  - Du graphe de résidus et le chemin d'augmentation obtenu en *Q*2
- Continuons l'algorithme.

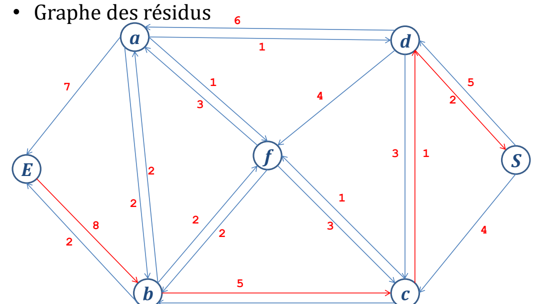
Exercice 3 – Solution (7/13)

- Mise à jour du flot



Exercice 3 – Solution (8/13)

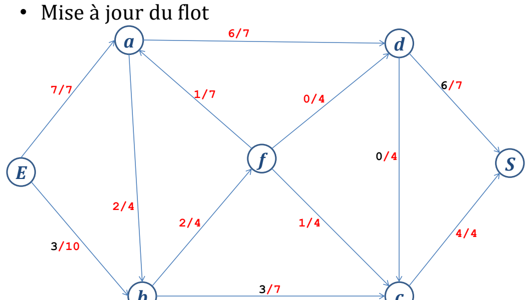
- Graphe des résidus



- Chemin d'augmentation : γ=*EbcdS* de capacité=1

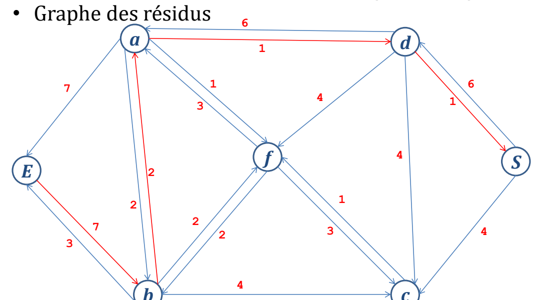
Exercice 3 – Solution (9/13)

- Mise à jour du flot



Exercice 3 – Solution (10/13)

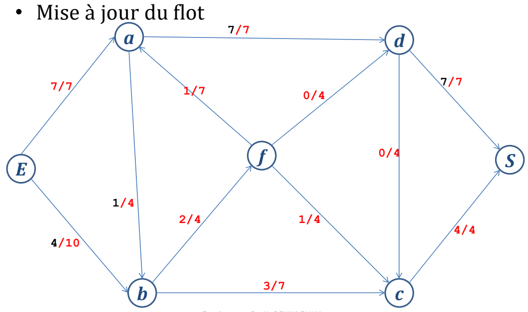
- Graphe des résidus



- Chemin d'augmentation : γ=*EbadS* de capacité=1

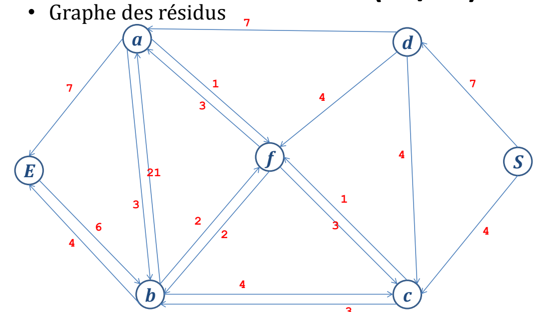
Exercice 3 – Solution (11/13)

- Mise à jour du flot



Exercice 3 – Solution (12/13)

- Graphe des résidus



- Chemin d'augmentation : Aucun

Exercice 3 – Solution (13/13)

- Le flot obtenu est maximal.

$\Rightarrow$ Sortant de l'entrée E

= entrant à la sortie F

= 11