Chapitre 3
Arbre de couverture optimal

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

Définition d'un arbre

- Soit $G=(X,U)$ un graphe orienté d'ordre $n \geq 2$. G est un **arbre** si l'une des **six (6) propriétés** suivantes est vérifiée :

1. G est <u>connexe</u> et <u>sans cycles</u>.
2. G est <u>sans cycles</u> et admet <u>$n - 1$ arcs ou arêtes</u>.
3. G est <u>connexe</u> et admet <u>$n - 1$ arcs ou arêtes</u>.
4. G est <u>sans cycle maximal</u> (tout arc ou arête supplémentaire créé un cycle dans G).

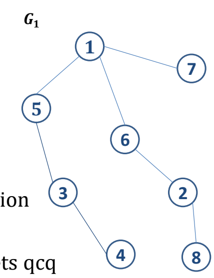
Définition d'un arbre

5. G est <u>connexe minimal</u> (la suppression d'un arc ou arête quelconque le rend non connexe).
6. Pour toute <u>paire de sommets</u> ($x, y \in X, x \neq y$) Il existe dans G <u>une chaîne et une seule</u> joignant x à y.

- Les 6 caractéristiques (propriétés) ci-dessus (propres aux arbres) sont équivalentes.

Arbre - Exemple

- Un arbre de :
  - 8 sommets
  - 7 arêtes
  - Connexe
  - Pas de cycles
  - Supprimer 1 arête : déconnexion
  - Rajouter 1 arête : cycle
  - 1 seule chaîne entre 2 sommets qcq

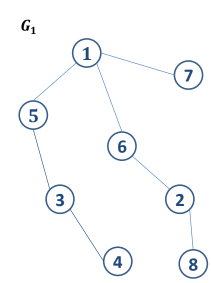


Codage de Prufer – algorithme de codage

```
P ← vide;
Tant que |X|>2
 Faire
  Choisir x dans X tel que dG(x)=1 et x
  minimal; // feuille de numéro minimal
  P ← P . Adjacent(x);
  // Rajouter le sommet adjacent à x dans la liste ordonnée P
  X ← X - {x}; // Supprimer le sommet x
  E ← E - {x, Adjacent(x)};
  // Supprimer l'arête incidente à x
 Fait
```

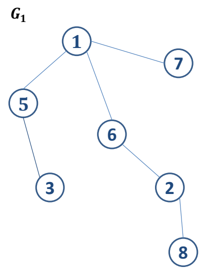

Prufer : Exemple Algorithme de codage

- Sommets de degré 1 :
  - 4 7 8
  - Min : 4
  - Voisin de 4 : 3
  - Rajouter 3 dans P
  - P : 3



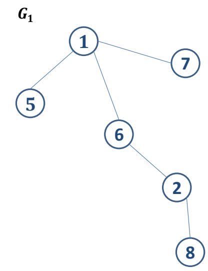
Prufer : Exemple Algorithme de codage

- Supprimer
  - le sommet 4
  - l'arête incidente à 4
- Sommets de degré 1 :
  - 3 7 8
  - Min : 3
  - Voisin de 3 : 5
  - Rajouter 5 dans P
  - P : 3 5



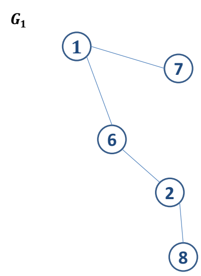
Prufer : Exemple Algorithme de codage

- Supprimer
  - le sommet 3
  - l'arête incidente à 3
- Sommets de degré 1 :
  - 5 7 8
  - Min : 5
  - Voisin de 5 : 1
  - Rajouter 1 dans P
  - P : 3 5 1



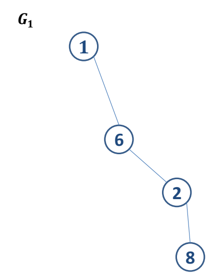
Prufer : Exemple Algorithme de codage

- Supprimer
  - le sommet 5
  - l'arête incidente à 5
- Sommets de degré 1 :
  - 7 8
  - Min : 7
  - Voisin de 7 : 1
  - Rajouter 1 dans P
  - P : 3 5 1 1



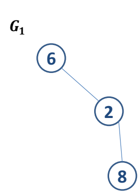
Prufer : Exemple Algorithme de codage

- Supprimer
  - le sommet 7
  - l'arête incidente à 7
- Sommets de degré 1 :
  - 1 8
  - Min : 1
  - Voisin de 1 : 6
  - Rajouter 6 dans P
  - P : 3 5 1 1 6



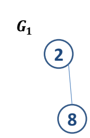
Prufer : Exemple Algorithme de codage

- Supprimer
  - le sommet 1
  - l'arête incidente à 1
- Sommets de degré 1 :
  - 6 8
  - Min : 6
  - Voisin de 6 : 2
  - Rajouter 2 dans P
  - P : 3 5 1 1 6 2



Prufer : Exemple Algorithme de codage

- Supprimer
  - le sommet 6
  - l'arête incidente à 6
- Il ne reste que 2 sommets
  - Fin du codage
  - P= 3 5 1 1 6 2



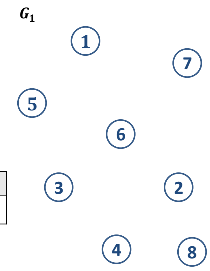
Codage de Prufer – algorithme de décodage

```
n ← longueur(P)+ 2;
X ← ∅;   E ← ∅;
Pour i de 1 à n
 X ← X ∪ {i};   D[i] ← 1;
Pour chaque valeur j de P
 D[i] ← D[i] + 1;
Pour chaque valeur j de P
  Chercher k tel que D[k]=1 et k minimal;
  E ← E ∪ {j,k};
  D[j] ← D[j]-1;   D[k] ← 0;
Relier les deux sommets ayant D[i]=1;
```


Prufer : Exemple Algorithme de décodage

- P= 3 5 1 1 6 2
- Longueur de P =6
- On prépare un graphe de 6+2=8 sommets isolés

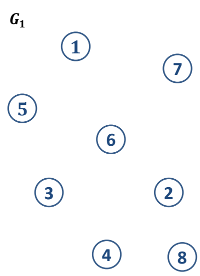
1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1



Prufer : Exemple Algorithme de décodage

- P= 3 5 1 1 6 2
- Mettre à jour le vecteur D selon P

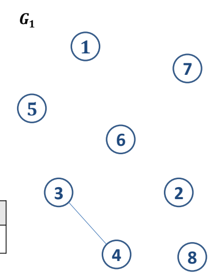
1	2	3	4	5	6	7	8
3	2	2	1	2	2	1	1



Prufer : Exemple Algorithme de décodage

- P= 3 5 1 1 6 2
  - Élément de P : 3
- D[i]=1 et i minimal
  - i=4
- Créer {3, 4}

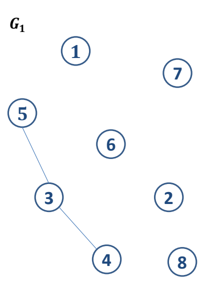
1	2	3	4	5	6	7	8
3	2	1	0	2	2	1	1



Prufer : Exemple Algorithme de décodage

- P= 3 5 1 1 6 2
  - Élément de P : 5
- D[i]=1 et i minimal
  - i=3
- Créer {3, 5}

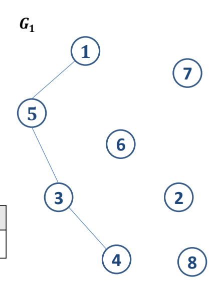
1	2	3	4	5	6	7	8
3	2	0	0	1	2	1	1



Prufer : Exemple Algorithme de décodage

- P= 3 5 1 1 6 2
  - Élément de P : 1
- D[i]=1 et i minimal
  - i=5
- Créer {1, 5}

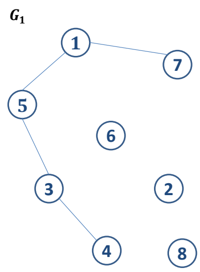
1	2	3	4	5	6	7	8
2	2	0	0	0	2	1	1



Prufer : Exemple Algorithme de décodage

- P= 3 5 1 1 6 2
  - Élément de P : 1
- D[i]=1 et i minimal
  - i=7
- Créer {1, 7}

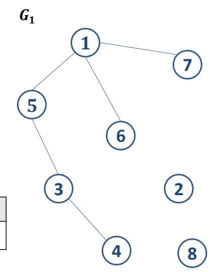
1	2	3	4	5	6	7	8
1	2	0	0	0	2	0	1



Prufer : Exemple Algorithme de décodage

- P= 3 5 1 1 6 2
  - Élément de P : 6
- D[i]=1 et i minimal
  - i=1
- Créer {1, 6}

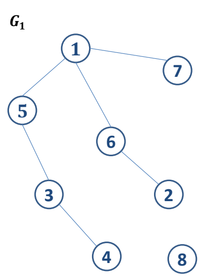
1	2	3	4	5	6	7	8
0	2	0	0	0	1	0	1



Prufer : Exemple Algorithme de décodage

- P= 3 5 1 1 6 2
  - Élément de P : 2
- D[i]=1 et i minimal
  - i=6
- Créer {2, 6}

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

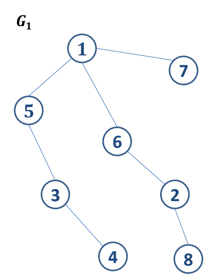


Prufer : Exemple Algorithme de décodage

- P= 3 5 1 1 6 2

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

- Il n'y a que 2 sommets ayant D[i]=1 : 2 et 8
- Créer {2, 8}
- Fin

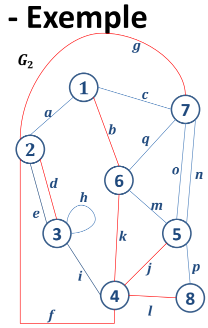


Arbre de couverture d'un graphe

- Soit *G=(X,U)* un graphe orienté d'ordre $n \geq 2$.
- On appelle <u>arbre dans *G*</u> un sous-graphe partiel de *G*, <u>*H=(Y, V)* connexe et sans cycles</u>.
- Un arbre est maximal dans *G* s'il contient le maximum possible de sommets de *G* (c'est-à-dire *Y=X* si *G* est connexe).

Arbre de couverture - Exemple

- En rouge, un arbre de couverture de G_2
- Cet arbre est maximal
- Il y a tous les sommets de *G*



Graphe pondéré (valué)

- **Poids d'un arc**
  - Soit *G=(X,U)* un graphe orienté,
  - On définit $p : U \rightarrow \Re$ une application
  - associe pour chaque arc $u \in U$ de *G* une valeur réelle *p(u)*
  - appelée poids de l'arc *u*.
- Un tel graphe est appelé:
  - graphe pondéré, graphe valué ou réseau.
  - Noté *G=(X,U, p)*

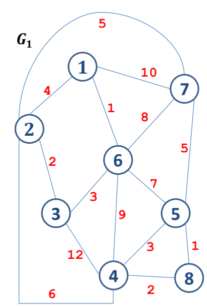
Identification du problème

- Soit *G=(X,U)* un graphe orienté connexe muni d'une application poids *p*.
- La recherche dans *G* d'un arbre de poids optimal revient à :
  - trouver un graphe partiel *G'=(X,U')* de *G* qui soit un arbre maximal
  - et pour lequel la somme des poids des arcs de *G'* soit optimale (maximale ou minimale selon la situation).

Algorithme de Kruksal

```
V = tri_poids_croissant(U,p);
H ← {V[1]};        i ← 1 ;        j ← 1;
Tant que (j < n-1)
 Faire
  i ← i+1; u ← V[i];
  Si (H ∪ {u} ne contient pas de cycle)
     Alors H ← H ∪ {u}; j ← j+1;
     // L'arc sélectionné ne doit pas créer de cycle
  fSi
 Fait
```


Algorithme de Kruksal - Exemple

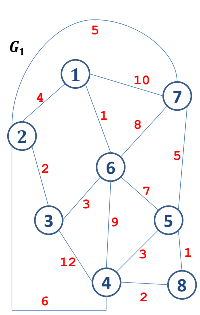


Algorithme de Kruksal - Exemple

- Classer les arêtes dans un ordre croissant des poids.

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

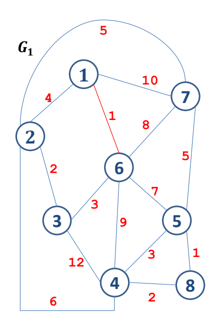
Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12



Algorithme de Kruksal - Exemple

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

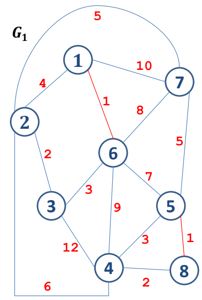
Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12



Algorithme de Kruksal - Exemple

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

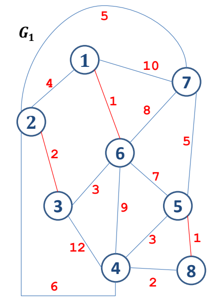
Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12



Algorithme de Kruksal - Exemple

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

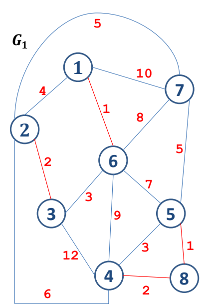
Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12



Algorithme de Kruksal - Exemple

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

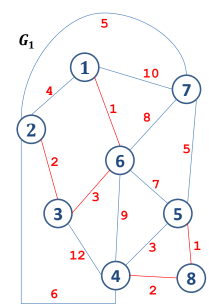
Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12



Algorithme de Kruksal - Exemple

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

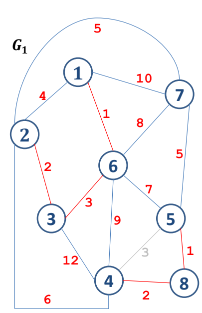
Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12



Algorithme de Kruksal - Exemple

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

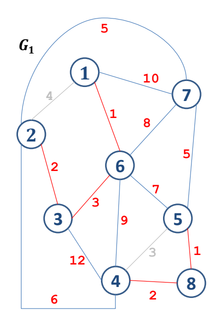
Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12



Algorithme de Kruksal - Exemple

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

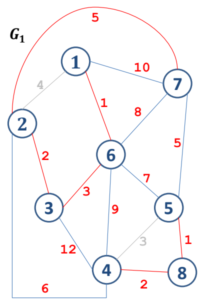
Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12



Algorithme de Kruksal - Exemple

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12

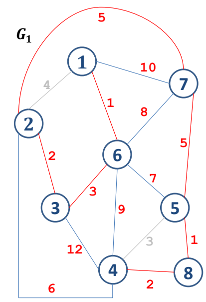


Algorithme de Kruksal - Exemple

- 7 arêtes sélectionnées pour 8 sommets : Fin Algo.

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12

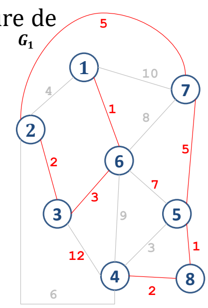


Algorithme de Kruksal - Exemple

- En rouge, l'arbre de couverture de poids minimal : Poids = 19.

Arête	Poids
{1,6}	1
{5,8}	1
{2,3}	2
{4,8}	2
{3,6}	3
{4,5}	3
{1,2}	4
{2,7}	5

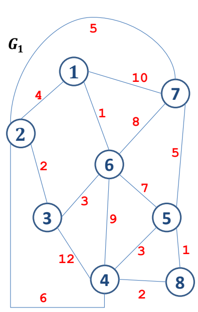
Arête	Poids
{5,7}	5
{2,4}	6
{5,6}	7
{6,7}	8
{4,6}	9
{1,7}	10
{3,4}	12



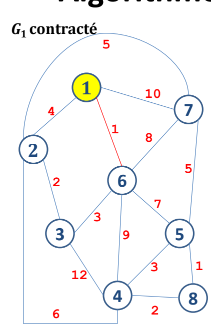
Algorithme de Prim

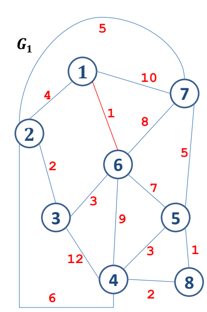
```
G ← R ;          H ← ∅;
Tant que |X|≥2
 Faire
  Choisir x dans X;
  u ← u(x);
  // choisir l'arc u de poids min ayant x comme extrémité
  H ← H ∪ {u};
  G ← Gu ; // Contracter G par rapport à l'arc u
Fait
```

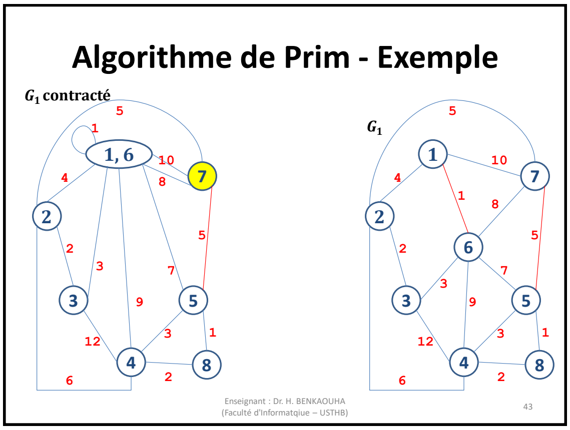

Algorithme de Prim - Exemple

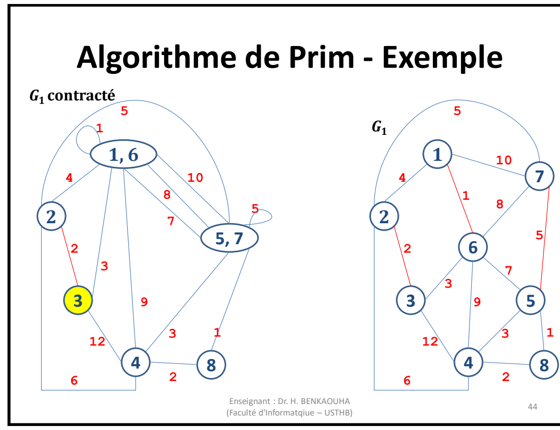


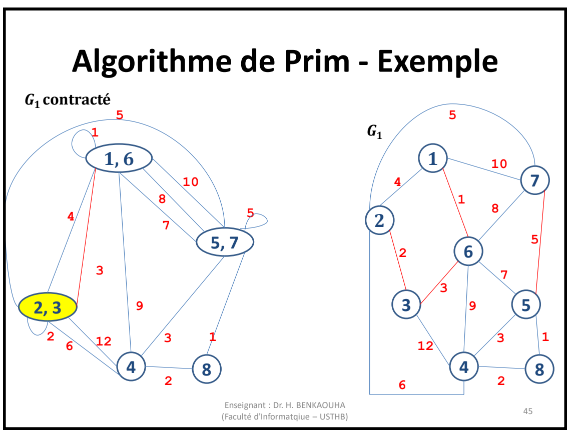
Algorithme de Prim - Exemple

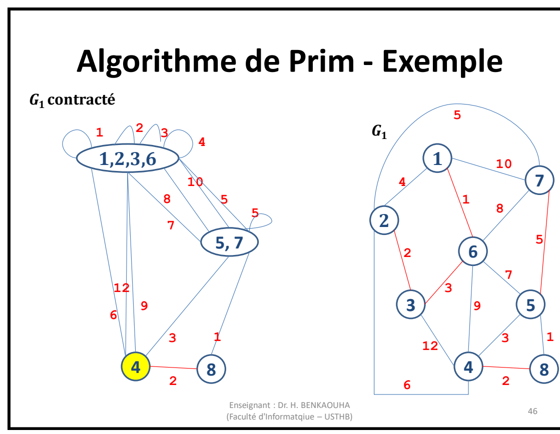


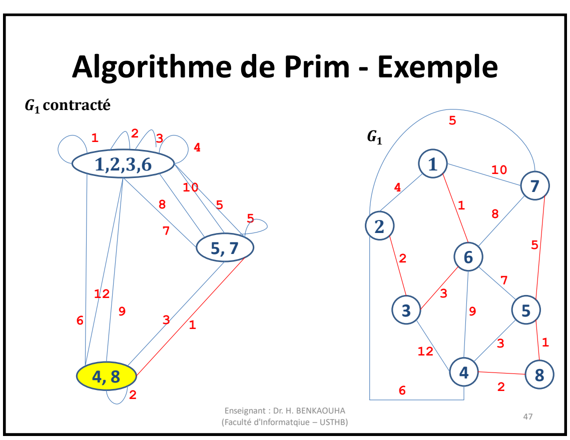


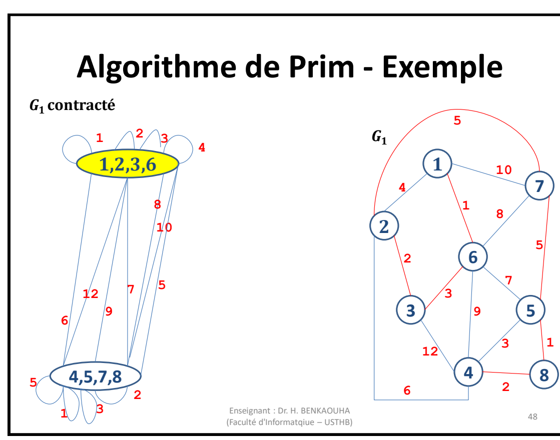

Algorithme de Prim - Exemple
G_1 contracté
G_1
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatqiue – USTHB)
43


Algorithme de Prim - Exemple
G_1 contracté
G_1
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatqiue – USTHB)
44


Algorithme de Prim - Exemple
G_1 contracté
G_1
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatqiue – USTHB)
45

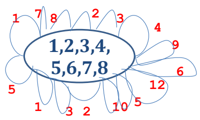

Algorithme de Prim - Exemple
G_1 contracté
G_1
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatqiue – USTHB)
46

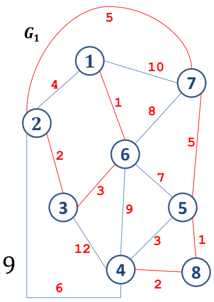

Algorithme de Prim - Exemple
G_1 contracté
G_1
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatqiue – USTHB)
47


Algorithme de Prim - Exemple
G_1 contracté
G_1
Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatqiue – USTHB)
48

Algorithme de Prim - Exemple

G_1 contracté





- Il n'y a qu'un seul sommet
  Dans le graphe contracté
- Fin de l'algorithme
- Arbre (en rouge) de poids 19

Enseignant : Dr. H. BENKAOUHA
(Faculté d'Informatqiue – USTHB)

49