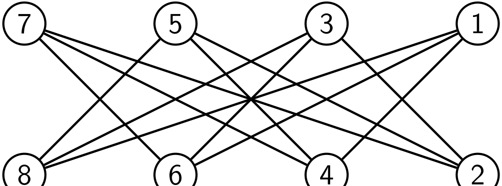
Des algorithmes de coloration

Les algorithmes suivants permettent (souvent) de donner de meilleures bornes aux nombres chromatiques mais n'en donnent pas systématiquement.

L'algorithme glouton

fonction G = Coloriage (G)

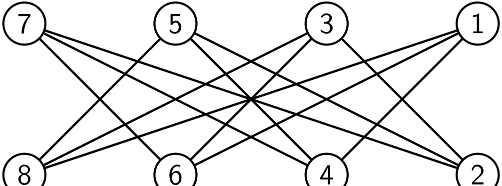
Ex: $color = \{Red \leq Blue \leq Green \leq Orange\}$



L'algorithme glouton

fonction G = Coloriage (G)
    couleur courante = 1

Ex: $color = \{Red \leq Blue \leq Green \leq Orange\}$

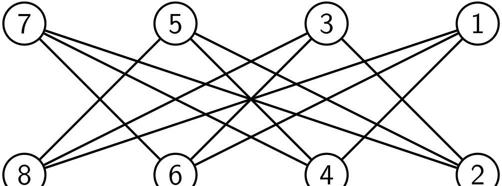


L'algorithme glouton

fonction G = Coloriage (G)
couleur courante $= 1$
pour tout x sommet de G **faire**

fin faire

Ex: $color = \{Red \leq Blue \leq Green \leq Orange\}$



L'algorithme glouton

<u>**fonction**</u> G = Coloriage (G)

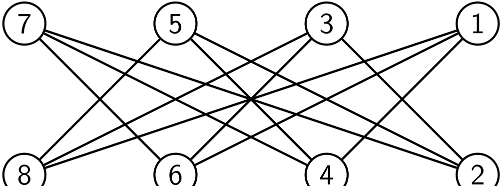
couleur courante $= 1$

<u>**pour tout**</u> x sommet de G <u>**faire**</u>

V = liste des voisins de x

<u>**fin faire**</u>

Ex: $color = \{Red \leq Blue \leq Green \leq Orange\}$

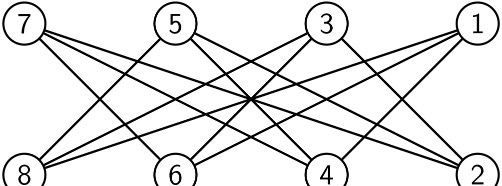


L'algorithme glouton

<u>**fonction**</u> G = Coloriage (G)
    couleur courante $= 1$
        <u>**pour tout**</u> x sommet de G <u>**faire**</u>
            V = liste des voisins de x
            couleur = plus petite couleur non encore utilisée dans V

        <u>**fin faire**</u>

Ex: $color = \{Red \leq Blue \leq Green \leq Orange\}$

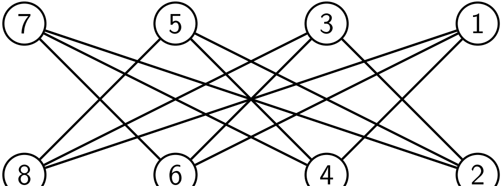


L'algorithme glouton

fonction G = Coloriage (G)
    couleur courante $= 1$
        pour tout x sommet de G **faire**
            V = liste des voisins de x
            couleur = plus petite couleur non encore utilisée dans V
            si *couleur* $\leq$ *couleur courante*

            fin
        fin faire

Ex: $color = \{Red \leq Blue \leq Green \leq Orange\}$

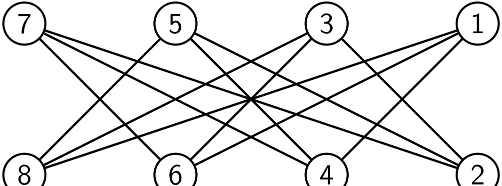


L'algorithme glouton

fonction G = Coloriage (G)
couleur courante $= 1$
pour tout x sommet de G **faire**
V = liste des voisins de x
couleur = plus petite couleur non encore utilisée dans V
si *couleur* $\leq$ *couleur courante*
alors colorier x avec cette couleur

fin
fin faire

Ex: $color = \{Red \leq Blue \leq Green \leq Orange\}$

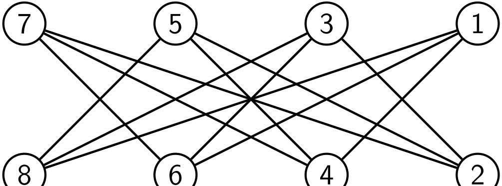


L'algorithme glouton

fonction G = Coloriage (G)
    couleur courante $= 1$
        pour tout x sommet de G **faire**
            V = liste des voisins de x
            couleur = plus petite couleur non encore utilisée dans V
            si *couleur* $\leq$ *couleur courante*
                    alors colorier x avec cette couleur
                    sinon incrémenter la *couleur courante*

            fin
        fin faire

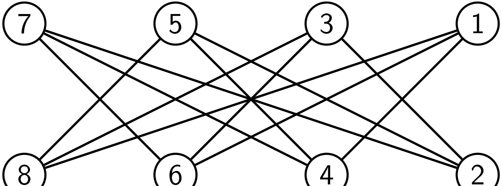
Ex: $color = \{Red \leq Blue \leq Green \leq Orange\}$



L'algorithme glouton

fonction G = Coloriage (G)
couleur courante $= 1$
pour tout x sommet de G **faire**
V = liste des voisins de x
couleur = plus petite couleur non encore utilisée dans V
si *couleur* $\leq$ *couleur courante*
alors colorier x avec cette couleur
sinon incrémenter la *couleur courante*
et colorer x avec
fin
fin faire

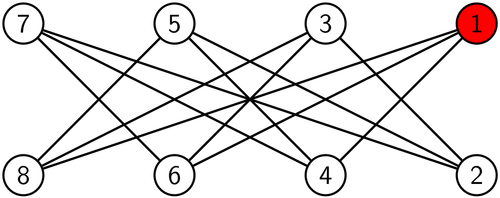
Ex: $color = \{Red \leq Blue \leq Green \leq Orange\}$



L'algorithme glouton

fonction G = Coloriage (G)
couleur courante $= 1$
pour tout x sommet de G **faire**
V = liste des voisins de x
couleur = plus petite couleur non encore utilisée dans V
si *couleur* $\leq$ *couleur courante*
alors colorier x avec cette couleur
sinon incrémenter la *couleur courante*
et colorer x avec
fin
fin faire

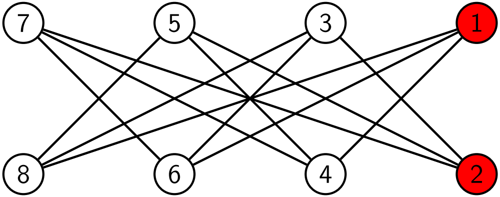
Ex: $color = \{\underline{Red} \leq Blue \leq Green \leq Orange\}$



L'algorithme glouton

fonction G = Coloriage (G)
couleur courante $= 1$
pour tout x sommet de G **faire**
V = liste des voisins de x
couleur = plus petite couleur non encore utilisée dans V
si *couleur* $\leq$ *couleur courante*
alors colorier x avec cette couleur
sinon incrémenter la *couleur courante*
et colorer x avec
fin
fin faire

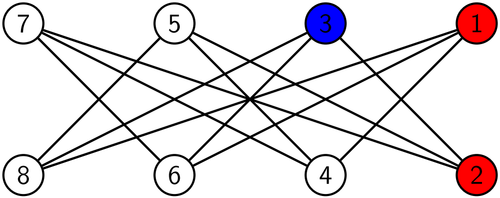
Ex: $color = \{\underline{Red} \leq Blue \leq Green \leq Orange\}$



L'algorithme glouton

fonction G = Coloriage (G)
couleur courante $= 1$
pour tout x sommet de G **faire**
V = liste des voisins de x
couleur = plus petite couleur non encore utilisée dans V
si *couleur* $\leq$ *couleur courante*
alors colorier x avec cette couleur
sinon incrémenter la *couleur courante*
et colorer x avec
fin
fin faire

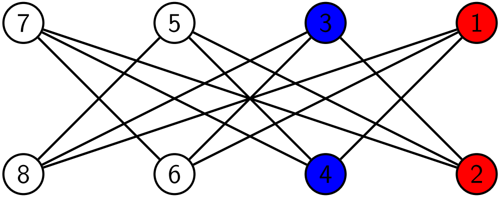
Ex: $color = \{Red \leq \underline{Blue} \leq Green \leq Orange\}$



L'algorithme glouton

fonction G = Coloriage (G)
  couleur courante $= 1$
    pour tout x sommet de G **faire**
      V = liste des voisins de x
      couleur = plus petite couleur non encore utilisée dans V
      si *couleur* $\leq$ *couleur courante*
        alors colorier x avec cette couleur
        sinon incrémenter la *couleur courante*
          et colorer x avec
      fin
    fin faire

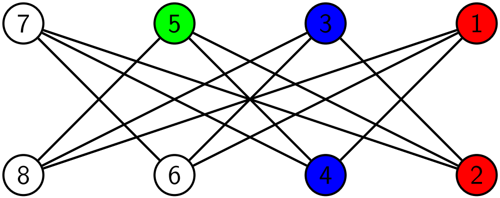
Ex: $color = \{Red \leq \underline{Blue} \leq Green \leq Orange\}$



L'algorithme glouton

<u>fonction</u> G = Coloriage (G)
    couleur courante $= 1$
        <u>pour tout</u> x sommet de G **<u>faire</u>**
            V = liste des voisins de x
            couleur = plus petite couleur non encore utilisée dans V
            <u>si</u> *couleur* $\leq$ *couleur courante*
                    <u>alors</u> colorier x avec cette couleur
                    <u>sinon</u> incrémenter la *couleur courante*
                        et colorer x avec
            <u>fin</u>
        <u>fin faire</u>

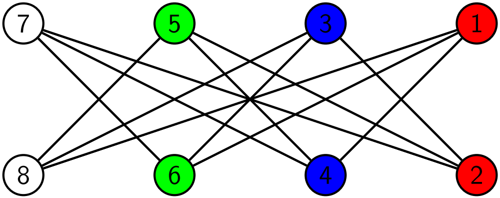
Ex: $color = \{Red \leq Blue \leq \underline{Green} \leq Orange\}$



L'algorithme glouton

fonction G = Coloriage (G)
couleur courante $= 1$
pour tout x sommet de G **faire**
V = liste des voisins de x
couleur = plus petite couleur non encore utilisée dans V
si *couleur* $\leq$ *couleur courante*
alors colorier x avec cette couleur
sinon incrémenter la *couleur courante*
et colorer x avec
fin
fin faire

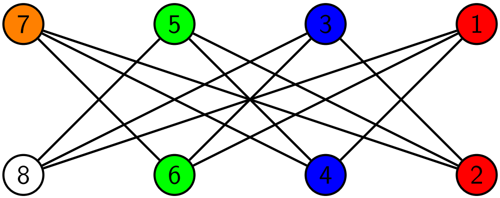
Ex: $color = \{Red \leq Blue \leq \underline{Green} \leq Orange\}$



L'algorithme glouton

fonction G = Coloriage (G)
  couleur courante $= 1$
    pour tout x sommet de G **faire**
      V = liste des voisins de x
      couleur = plus petite couleur non encore utilisée dans V
      si *couleur* $\leq$ *couleur courante*
        alors colorier x avec cette couleur
        sinon incrémenter la *couleur courante*
          et colorer x avec
      fin
    fin faire

Ex: $color = \{Red \leq Blue \leq Green \leq \underline{Orange}\}$



L'algorithme glouton

fonction G = Coloriage (G)

couleur courante $= 1$

pour tout x sommet de G **faire**

V = liste des voisins de x

couleur = plus petite couleur non encore utilisée dans V

si *couleur* $\leq$ *couleur courante*

alors colorier x avec cette couleur

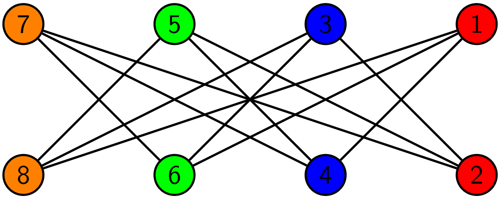
sinon incrémenter la *couleur courante*

et colorer x avec

fin

fin faire

Ex: $color = \{Red \leq Blue \leq Green \leq \underline{Orange}\}$



Une amélioration

Cet algorithme peut-être amélioré en traitant les sommets dans l'ordre croissant de leur degré.
C'est l'algorithme de Welsh-Powell.
Il ne donne toujours pas nécessairement le nombre chromatique.

Algorithme de Welsh-Powell

<u>fonction</u> $G = \text{Welsh (G)}$

Algorithme de Welsh-Powell

fonction G = Welsh (G)
L = liste des sommets classés dans l'ordre décroissant de leur degré

Algorithme de Welsh-Powell

<u>**fonction**</u> $G =$ Welsh (G)

$L =$ liste des sommets classés dans l'ordre décroissant de leur degré

$\mathit{couleur\ courante} = 0$

Algorithme de Welsh-Powell

fonction G = Welsh (G)
L = liste des sommets classés dans l'ordre décroissant de leur degré
couleur courante $= 0$
tant que $L \neq \emptyset$ **faire**

Algorithme de Welsh-Powell

fonction G = Welsh (G)

L = liste des sommets classés dans l'ordre décroissant de leur degré

couleur courante $= 0$

tant que $L \neq \emptyset$ **faire**

incrémenter la *couleur courante*

Algorithme de Welsh-Powell

fonction G = Welsh (G)
    L = liste des sommets classés dans l'ordre décroissant de leur degré
    couleur courante $= 0$
    tant que $L \neq \emptyset$ **faire**
        incrémenter la *couleur courante*
        Colorier s le premier sommet de L avec la *couleur courante*

Algorithme de Welsh-Powell

fonction G = Welsh (G)
L = liste des sommets classés dans l'ordre décroissant de leur degré
couleur courante $= 0$
tant que $L \neq \emptyset$ **faire**
incrémenter la *couleur courante*
Colorier s le premier sommet de L avec la *couleur courante*
Eliminer s de L

Algorithme de Welsh-Powell

fonction G = Welsh (G)

L = liste des sommets classés dans l'ordre décroissant de leur degré

couleur courante $= 0$

tant que $L \neq \emptyset$ **faire**

incrémenter la *couleur courante*

Colorier s le premier sommet de L avec la *couleur courante*

Eliminer s de L

V = liste des voisins de s

Algorithme de Welsh-Powell

fonction G = Welsh (G)

L = liste des sommets classés dans l'ordre décroissant de leur degré

couleur courante $= 0$

tant que $L \neq \emptyset$ **faire**

incrémenter la *couleur courante*

Colorier s le premier sommet de L avec la *couleur courante*

Eliminer s de L

V = liste des voisins de s

pour tout x dans L **faire**

Algorithme de Welsh-Powell

fonction G = Welsh (G)

L = liste des sommets classés dans l'ordre décroissant de leur degré

couleur courante $= 0$

tant que $L \neq \emptyset$ **faire**

incrémenter la *couleur courante*

Colorier s le premier sommet de L avec la *couleur courante*

Eliminer s de L

V = liste des voisins de s

pour tout x dans L **faire**

si $x \notin V$ **alors** colorier x avec la *couleur courante*

Algorithme de Welsh-Powell

<u>fonction</u> G = Welsh (G)
  L = liste des sommets classés dans l'ordre décroissant de leur degré
  couleur courante = 0
  <u>tant que</u> $L \neq \emptyset$ **<u>faire</u>**
    incrémenter la *couleur courante*
    Colorier s le premier sommet de L avec la *couleur courante*
    Eliminer s de L
    V = liste des voisins de s
    <u>pour tout</u> x dans L **<u>faire</u>**
      <u>si</u> $x \notin V$ **<u>alors</u>** colorier x avec la *couleur courante*
        ajouter les voisins de x à V

Algorithme de Welsh-Powell

fonction G = Welsh (G)
    L = liste des sommets classés dans l'ordre décroissant de leur degré
    couleur courante $= 0$
    tant que $L \neq \emptyset$ **faire**
        incrémenter la *couleur courante*
        Colorier s le premier sommet de L avec la *couleur courante*
        Eliminer s de L
        V = liste des voisins de s
        pour tout x dans L **faire**
            si $x \notin V$ **alors** colorier x avec la *couleur courante*
                ajouter les voisins de x à V
            fin

Algorithme de Welsh-Powell

fonction G = Welsh (G)

    L = liste des sommets classés dans l'ordre décroissant de leur degré
    couleur courante = 0
    tant que L ≠ ∅ faire
        incrémenter la couleur courante
        Colorier s le premier sommet de L avec la couleur courante
        Eliminer s de L
        V = liste des voisins de s
        pour tout x dans L faire
                si x ∉ V alors colorier x avec la couleur courante
                               ajouter les voisins de x à V
                fin
        fin faire

Algorithme de Welsh-Powell

fonction G = Welsh (G)
  L = liste des sommets classés dans l'ordre décroissant de leur degré
  couleur courante $= 0$
  tant que $L \neq \emptyset$ **faire**
    incrémenter la *couleur courante*
    Colorier s le premier sommet de L avec la *couleur courante*
    Eliminer s de L
    V = liste des voisins de s
    pour tout x dans L **faire**
      si $x \notin V$ **alors** colorier x avec la *couleur courante*
        ajouter les voisins de x à V
      fin
    fin faire
    Eliminer les sommets coloriés de L

Algorithme de Welsh-Powell

fonction G = Welsh (G)

L = liste des sommets classés dans l'ordre décroissant de leur degré
couleur courante = 0
tant que $L \neq \emptyset$ **faire**
  incrémenter la *couleur courante*
  Colorier s le premier sommet de L avec la *couleur courante*
  Eliminer s de L
  V = liste des voisins de s
  pour tout x dans L **faire**
    si $x \notin V$ **alors** colorier x avec la *couleur courante*
      ajouter les voisins de x à V
    fin
  fin faire
  Eliminer les sommets coloriés de L

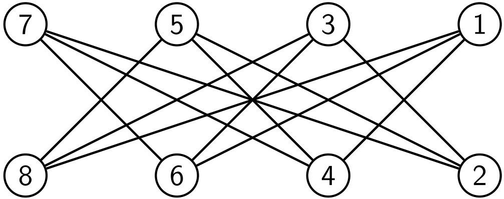
fin faire

Ces algorithmes ne sont pas optimaux

Considérons le graphe ci-dessous.

Ces algorithmes ne sont pas optimaux

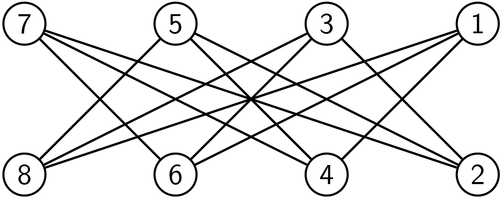
Considérons le graphe ci-dessous.



Ces algorithmes ne sont pas optimaux

Considérons le graphe ci-dessous.

L'algorithme glouton donne 4 couleurs

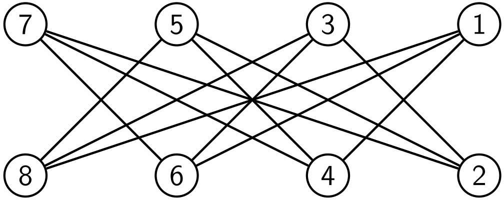


Ces algorithmes ne sont pas optimaux

Considérons le graphe ci-dessous.

L'algorithme glouton donne 4 couleurs

L'algorithme de Welsh-Powell donne 4 couleurs

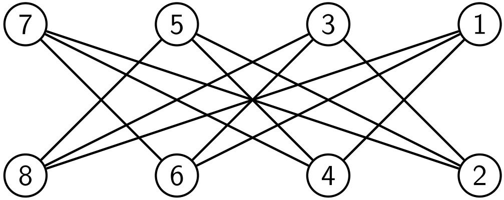


Ces algorithmes ne sont pas optimaux

Considérons le graphe ci-dessous.

L'algorithme glouton donne 4 couleurs

L'algorithme de Welsh-Powell donne 4 couleurs
Pourtant le nombre chromatique vaut ...

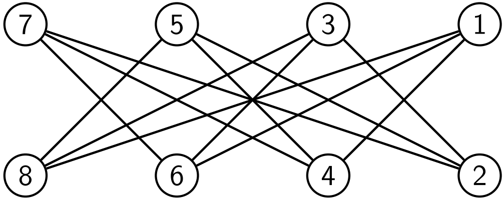


Ces algorithmes ne sont pas optimaux

Considérons le graphe ci-dessous.

L'algorithme glouton donne 4 couleurs

L'algorithme de Welsh-Powell donne 4 couleurs
Pourtant le nombre chromatique vaut ... 2



Un exemple

Dans un groupe de TP de 14 étudiants on doit former des groupes de PPP de quelques étudiants en faisant en sorte que les étudiants d'un même groupe ne s'entendent pas trop mal entre eux. On connaît pour chaque étudiant les membres du groupe avec lesquels il ne s'entend pas :

Le tableau des inimitiés

l'étudiant(e)	1	2	3	4	5	6	7
ne s'entend pas avec	3 ;5 9 12	5 ;6 7 ;8 12	4 ;5 10 1; 11	3 10 14	1 ;2 ;3 7 ; 8 9;11	2 ; 8 12 13	2 ;5 9 12

l'étudiant(e)	8	9	10	11	12	13	14
ne s'entend pas avec	2 ;11 5 ;13 6;14	1 ;5 7 12	3 ;4 11 14	3 ;5 10 14; 8	1 ;2 7 ;9 6	6 14 8	4 ;8 11 ;1 10

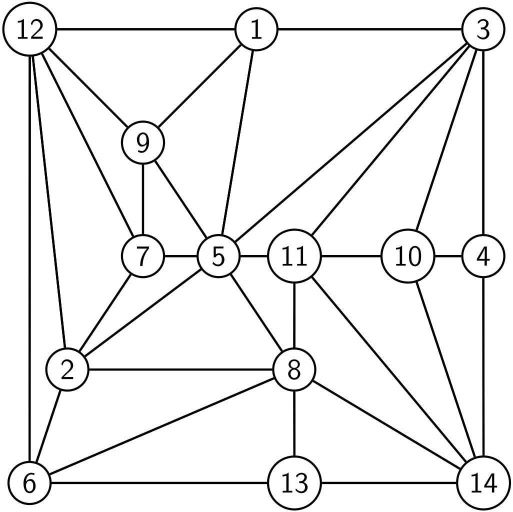
Représentation sagittale

On représente la situation par un graphe simple non-orienté.

Les sommets représentant les étudiants.

On trace une arête quand deux étudiants ne s'entendent pas.

Représentation sagittale



L'exemple continue

Pour former les groupes il suffit de colorier le graphe, chaque couleur constituera un groupe.

L'exemple continue

Pour former les groupes il suffit de colorier le graphe, chaque couleur constituera un groupe.

Montrer qu'avec l'algorithme glouton et de Welsh-Powell on trouve 4 groupes ... et que les 2 solutions sont différentes.

L'exemple continue

Pour former les groupes il suffit de colorier le graphe, chaque couleur constituera un groupe.

Montrer qu'avec l'algorithme glouton et de Welsh-Powell on trouve 4 groupes ... et que les 2 solutions sont différentes.

Ensuite, montrer que le nombre chromatique est 4.