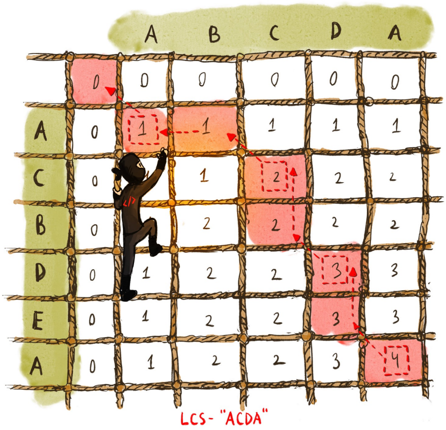
Chapitre 6 : Programmation dynamique



Introduction

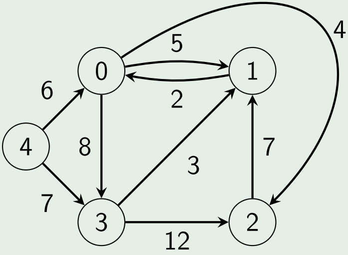
Problème

- $f : \mathcal{U} \to \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \min_{y \in \mathcal{U}}\{f(y)\}$ (ou $f(x) = \max_{y \in \mathcal{U}}\{f(y)\}$)

Problème

- $f : \mathcal{U} \rightarrow \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \min_{y \in \mathcal{U}}\{f(y)\}$ (ou $f(x) = \max_{y \in \mathcal{U}}\{f(y)\}$)

Exemple : plus court(s) chemin(s) dans un graphe

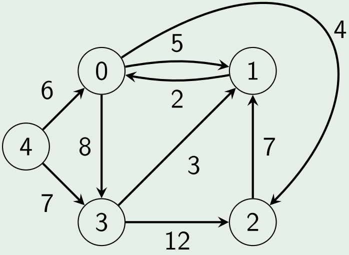


Introduction

Problème

- $f : \mathcal{U} \to \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \min_{y\in\mathcal{U}}\{f(y)\}$ (ou $f(x) = \max_{y\in\mathcal{U}}\{f(y)\}$)

Exemple : plus court(s) chemin(s) dans un graphe



On verra ça l'an prochain !

Introduction

Problème

- $f : \mathcal{U} \to \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \min_{y \in \mathcal{U}}\{f(y)\}$ (ou $f(x) = \max_{y \in \mathcal{U}}\{f(y)\}$)

Chemin dans une matrice

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

Introduction

Problème

- $f : \mathcal{U} \to \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \min_{y \in \mathcal{U}} \{f(y)\}$ (ou $f(x) = \max_{y \in \mathcal{U}} \{f(y)\}$)

Chemin dans une matrice

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

Problème : chemin de la case en haut à gauche à la case en bas à droite via $\downarrow$ et $\rightarrow$, qui maximise les entiers rencontrés ?

Introduction

Problème

- $f : \mathcal{U} \to \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \min_{y \in \mathcal{U}}\{f(y)\}$ (ou $f(x) = \max_{y \in \mathcal{U}}\{f(y)\}$)

Chemin dans une matrice

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

Problème : chemin de la case en haut à gauche à la case en bas à droite via $\downarrow$ et $\rightarrow$, qui maximise les entiers rencontrés ?

Sous-séquence commune maximale ?

Une plus longue sous-séquence commune à « arythmie » et « rhomboédrique » est « rhmie ».

Chemin de poids max dans une matrice

Problème

- $A = (a_{i,j})_{0 \leq i < n, 0 \leq j < m} \in \mathcal{M}_{n,m}(\mathbb{N})$

Chemin de poids max dans une matrice

Problème

- $A = (a_{i,j})_{0 \leq i < n, 0 \leq j < m} \in \mathcal{M}_{n,m}(\mathbb{N})$
- univers $\mathcal{U}$: chemins de la case en haut à gauche $(0,0)$ à la case en bas à droite $(n-1, m-1)$ n'utilisant que les déplacements $\rightarrow$ et $\downarrow$

Chemin de poids max dans une matrice

Problème

- $A = (a_{i,j})_{0 \leq i < n, 0 \leq j < m} \in \mathcal{M}_{n,m}(\mathbb{N})$
- univers $\mathcal{U}$: chemins de la case en haut à gauche $(0,0)$ à la case en bas à droite $(n-1, m-1)$ n'utilisant que les déplacements $\rightarrow$ et $\downarrow$
- fonction p : *Poids* d'un chemin, somme des entiers rencontrés ;

Chemin de poids max dans une matrice

Problème

- $A = (a_{i,j})_{0 \leq i < n, 0 \leq j < m} \in \mathcal{M}_{n,m}(\mathbb{N})$
- univers $\mathcal{U}$: chemins de la case en haut à gauche $(0,0)$ à la case en bas à droite $(n-1, m-1)$ n'utilisant que les déplacements $\rightarrow$ et $\downarrow$
- fonction p : *Poids* d'un chemin, somme des entiers rencontrés ;
- But : trouver c tel que $p(c) = \max_{u \in \mathcal{U}} \{p(u)\}$

Chemin de poids max dans une matrice

Problème

- $A = (a_{i,j})_{0 \leq i < n, 0 \leq j < m} \in \mathcal{M}_{n,m}(\mathbb{N})$
- univers $\mathcal{U}$: chemins de la case en haut à gauche $(0,0)$ à la case en bas à droite $(n-1, m-1)$ n'utilisant que les déplacements $\rightarrow$ et $\downarrow$
- fonction p : *Poids* d'un chemin, somme des entiers rencontrés ;
- But : trouver c tel que $p(c) = \max_{u \in \mathcal{U}}\{p(u)\}$

Recherche exhaustive ?

- Il y a $N_{n,m} =$ tels chemins ;

Chemin de poids max dans une matrice

Problème

- $A = (a_{i,j})_{0 \leq i < n, 0 \leq j < m} \in \mathcal{M}_{n,m}(\mathbb{N})$
- univers $\mathcal{U}$: chemins de la case en haut à gauche $(0,0)$ à la case en bas à droite $(n-1, m-1)$ n'utilisant que les déplacements $\rightarrow$ et $\downarrow$
- fonction p : *Poids* d'un chemin, somme des entiers rencontrés ;
- But : trouver c tel que $p(c) = \max_{u \in \mathcal{U}} \{p(u)\}$

Recherche exhaustive ?

- Il y a $N_{n,m} = \binom{n+m-2}{n-1}$ tels chemins ;

Chemin de poids max dans une matrice

Problème

- $A = (a_{i,j})_{0 \leq i < n, 0 \leq j < m} \in \mathcal{M}_{n,m}(\mathbb{N})$
- univers $\mathcal{U}$: chemins de la case en haut à gauche $(0,0)$ à la case en bas à droite $(n-1, m-1)$ n'utilisant que les déplacements $\rightarrow$ et $\downarrow$
- fonction p : *Poids* d'un chemin, somme des entiers rencontrés ;
- But : trouver c tel que $p(c) = \max_{u \in \mathcal{U}} \{p(u)\}$

Recherche exhaustive ?

- Il y a $N_{n,m} = \binom{n+m-2}{n-1}$ tels chemins ;
- pour $n = m$, $N_{n,n} \underset{n \to +\infty}{\sim} \frac{2^{2n-2}}{\sqrt{\pi n}}$

Chemin de poids max dans une matrice

Problème

- $A = (a_{i,j})_{0 \leq i < n, 0 \leq j < m} \in \mathcal{M}_{n,m}(\mathbb{N})$
- univers $\mathcal{U}$: chemins de la case en haut à gauche $(0,0)$ à la case en bas à droite $(n-1, m-1)$ n'utilisant que les déplacements $\rightarrow$ et $\downarrow$
- fonction p : *Poids* d'un chemin, somme des entiers rencontrés ;
- But : trouver c tel que $p(c) = \max_{u \in \mathcal{U}} \{p(u)\}$

Recherche exhaustive ?

- Il y a $N_{n,m} = \binom{n+m-2}{n-1}$ tels chemins ;
- pour $n = m$, $N_{n,n} \underset{n \to +\infty}{\sim} \frac{2^{2n-2}}{\sqrt{\pi n}}$
- $n = 50$: 32247603683100 chemins.

Chemin de poids max dans une matrice

Problème

- $A = (a_{i,j})_{0 \leq i < n, 0 \leq j < m} \in \mathcal{M}_{n,m}(\mathbb{N})$
- univers $\mathcal{U}$: chemins de la case en haut à gauche $(0,0)$ à la case en bas à droite $(n-1, m-1)$ n'utilisant que les déplacements $\rightarrow$ et $\downarrow$
- fonction p : *Poids* d'un chemin, somme des entiers rencontrés ;
- But : trouver c tel que $p(c) = \max_{u \in \mathcal{U}} \{p(u)\}$

Recherche exhaustive ?

- Il y a $N_{n,m} = \binom{n+m-2}{n-1}$ tels chemins ;
- pour $n = m$, $N_{n,n} \underset{n \to +\infty}{\sim} \frac{2^{2n-2}}{\sqrt{\pi n}}$
- $n = 50$: 32247603683100 chemins.
- Un algorithme de recherche exhaustive est impraticable sauf pour n très petit !

Chemin de poids max dans une matrice

Sous-structure optimale

Si un chemin optimal c passe par la case (i, j),
$c = (0, 0) \overset{c_1}{\rightsquigarrow} (i, j) \overset{c_2}{\rightsquigarrow} (n-1, m-1)$. Alors c_1 et c_2 sont des chemins de poids maximaux de la case $(0, 0)$ à (i, j) et de la case (i, j) à $(n-1, m-1)$.

Chemin de poids max dans une matrice

Sous-structure optimale

Si un chemin optimal c passe par la case (i,j), $c = (0,0) \overset{c_1}{\rightsquigarrow} (i,j) \overset{c_2}{\rightsquigarrow} (n-1, m-1)$. Alors c_1 et c_2 sont des chemins de poids maximaux de la case $(0,0)$ à (i,j) et de la case (i,j) à $(n-1, m-1)$.

$\rightsquigarrow$ Preuve !

Chemin de poids max dans une matrice

Sous-structure optimale

Si un chemin optimal c passe par la case (i, j),
$c = (0, 0) \overset{c_1}{\rightsquigarrow} (i, j) \overset{c_2}{\rightsquigarrow} (n-1, m-1)$. Alors c_1 et c_2 sont des chemins de poids maximaux de la case $(0, 0)$ à (i, j) et de la case (i, j) à $(n-1, m-1)$.
$\rightsquigarrow$ Preuve !

Formulation récursive

- $p_{i,j}$: poids d'un chemin optimal de $(0, 0)$ à (i, j).

Chemin de poids max dans une matrice

Sous-structure optimale

Si un chemin optimal c passe par la case (i, j),
$c = (0, 0) \overset{c_1}{\rightsquigarrow} (i, j) \overset{c_2}{\rightsquigarrow} (n-1, m-1)$. Alors c_1 et c_2 sont des chemins de poids maximaux de la case $(0, 0)$ à (i, j) et de la case (i, j) à $(n-1, m-1)$.
$\rightsquigarrow$ Preuve !

Formulation récursive

- $p_{i,j}$: poids d'un chemin optimal de $(0, 0)$ à (i, j).
- On veut $p_{n-1,m-1}$.

Chemin de poids max dans une matrice

Sous-structure optimale

Si un chemin optimal c passe par la case (i,j), $c = (0,0) \overset{c_1}{\rightsquigarrow} (i,j) \overset{c_2}{\rightsquigarrow} (n-1, m-1)$. Alors c_1 et c_2 sont des chemins de poids maximaux de la case $(0,0)$ à (i,j) et de la case (i,j) à $(n-1, m-1)$.

$\rightsquigarrow$ Preuve !

Formulation récursive

- $p_{i,j}$: poids d'un chemin optimal de $(0,0)$ à (i,j).
- On veut $p_{n-1,m-1}$.
- Relation :

$$p_{i,j} = a_{i,j} + \begin{cases} 0 & \text{si } i = j = 0 \\ p_{i,j-1} & \text{si } i = 0, \text{ et } j > 0 \\ p_{i-1,j} & \text{si } j = 0, \text{ et } i > 0 \\ \max\{p_{i-1,j}, p_{i,j-1}\} & \text{sinon.} \end{cases}$$

Chemin de poids max dans une matrice

Sous-structure optimale

Si un chemin optimal c passe par la case (i, j),
$c = (0, 0) \overset{c_1}{\leadsto} (i, j) \overset{c_2}{\leadsto} (n-1, m-1)$. Alors c_1 et c_2 sont des chemins de poids maximaux de la case $(0, 0)$ à (i, j) et de la case (i, j) à $(n-1, m-1)$.

$\leadsto$ Preuve !

Formulation récursive

- $p_{i,j}$: poids d'un chemin optimal de $(0, 0)$ à (i, j).
- On veut $p_{n-1,m-1}$.
- Relation :

$$p_{i,j} = a_{i,j} + \begin{cases} 0 & \text{si } i = j = 0 \\ p_{i,j-1} & \text{si } i = 0, \text{ et } j > 0 \\ p_{i-1,j} & \text{si } j = 0, \text{ et } i > 0 \\ \max\{p_{i-1,j}, p_{i,j-1}\} & \text{sinon.} \end{cases}$$

$\leadsto$ Preuve !

Chemin de poids max dans une matrice

Calcul de *tous* les $p_{i,j}$

- Calcul récursif de $p_{n-1,m-1}$: hors de question !

Chemin de poids max dans une matrice

Calcul de *tous* les $p_{i,j}$

- Calcul récursif de $p_{n-1,m-1}$: hors de question !
- Calcul itératif, dans une matrice.

Chemin de poids max dans une matrice

Calcul de *tous* les $p_{i,j}$

- Calcul récursif de $p_{n-1,m-1}$: hors de question !
- Calcul itératif, dans une matrice.

Code

```
let calcul_p a=
  let n,m=Array.length a, Array.length a.(0) in
  let p=Array.make_matrix n m a.(0).(0) in
  for i=1 to n-1 do
    p.(i).(0) <- p.(i-1).(0) + a.(i).(0)
  done ;
  for j=1 to m-1 do
    p.(0).(j) <- p.(0).(j-1) + a.(0).(j)
  done ;
  for i=1 to n-1 do
    for j=1 to m-1 do
      p.(i).(j) <- a.(i).(j) + max p.(i-1).(j) p.(i).(j-1)
    done
  done ;
  p
;;
```

Calcul de $(p_{i,j})$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & 159 & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & 193 & 220 & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & 260 & 296 & 309 \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & 315 \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & 159 & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & 193 & 220 & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & 260 & 296 & 309 \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & 315 \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & 159 & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & 193 & 220 & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & 260 & 296 & 309 \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & \color{red}{315} \end{pmatrix}$$

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & 159 & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & 193 & 220 & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & 260 & 296 & \color{red}{309} \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & \color{red}{315} \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & 159 & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & 193 & 220 & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & 260 & 296 & 309 \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & 315 \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & 159 & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & 193 & 220 & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & 260 & 296 & 309 \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & 315 \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & 159 & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & 193 & \color{red}{220} & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & \color{red}{260} & \color{red}{296} & \color{red}{309} \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & \color{red}{315} \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & 159 & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & \color{red}{193} & \color{red}{220} & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & \color{red}{260} & \color{red}{296} & \color{red}{309} \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & \color{red}{315} \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & \color{red}{159} & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & \color{red}{193} & \color{red}{220} & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & \color{red}{260} & \color{red}{296} & \color{red}{309} \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & \color{red}{315} \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & {\color{red}149} & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & {\color{red}159} & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & {\color{red}193} & {\color{red}220} & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & {\color{red}260} & {\color{red}296} & {\color{red}309} \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & {\color{red}315} \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & \color{red}{102} & \color{red}{149} & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & \color{red}{159} & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & \color{red}{193} & \color{red}{220} & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & \color{red}{260} & \color{red}{296} & \color{red}{309} \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & \color{red}{315} \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & 159 & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & 193 & 220 & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & 260 & 296 & 309 \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & 315 \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & \color{red}{41} & \color{red}{53} & \color{red}{102} & \color{red}{149} & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & \color{red}{159} & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & \color{red}{193} & \color{red}{220} & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & \color{red}{260} & \color{red}{296} & \color{red}{309} \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & \color{red}{315} \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} \color{red}{2} & \color{red}{41} & \color{red}{53} & \color{red}{102} & \color{red}{149} & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & \color{red}{159} & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & \color{red}{193} & \color{red}{220} & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & \color{red}{260} & \color{red}{296} & \color{red}{309} \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & \color{red}{315} \end{pmatrix}$$

Chemin de poids max dans une matrice

Chemin optimal : remonter de $(n-1, m-1)$ vers $(0,0)$:

$$A = \begin{pmatrix} 2 & 39 & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & 34 & 27 & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & 40 & 36 & 13 \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & 6 \end{pmatrix}$$

$$P = \begin{pmatrix} 2 & 41 & 53 & 102 & 149 & 167 & 189 & 208 \\ 39 & 62 & 96 & 128 & 159 & 169 & 224 & 263 \\ 70 & 91 & 108 & 154 & 193 & 220 & 231 & 285 \\ 90 & 137 & 153 & 156 & 204 & 260 & 296 & 309 \\ 108 & 167 & 199 & 236 & 264 & 288 & 305 & 315 \end{pmatrix}$$

Chemin de poids max dans une matrice

En Caml : encodage d'un chemin comme une liste de ">" ou "v".

```
let max_chemin a=
  let n,m=Array.length a, Array.length a.(0) in
  let p=calcul_p a in
  let i=ref (n-1) and j=ref (m-1) and q=ref [] in
  while !i>0 && !j>0 do
    if p.( !i-1).( !j) > p.( !i).( !j-1) then
      begin q:= "v":: !q ; decr i end
    else
      begin q:= ">":: !q ; decr j end
  done ;
  while !i>0 do
    q:="v":: !q ; decr i
  done ;
  while !j>0 do
    q:=">":: !q ; decr j
  done ;
  !q ;;
```

Chemin de poids max dans une matrice

En Caml : encodage d'un chemin comme une liste de ">" ou "v".

```
let max_chemin a=
  let n,m=Array.length a, Array.length a.(0) in
  let p=calcul_p a in
  let i=ref (n-1) and j=ref (m-1) and q=ref [] in
  while !i>0 && !j>0 do
    if p.( !i-1).( !j) > p.( !i).( !j-1) then
      begin q:= "v":: !q ; decr i end
    else
      begin q:= ">":: !q ; decr j end
  done ;
  while !i>0 do
    q:="v":: !q ; decr i
  done ;
  while !j>0 do
    q:=">":: !q ; decr j
  done ;
  !q ;;
```

```
# max_chemin a ;;
- : string list = [">"; ">"; ">"; ">"; "v"; "v"; ">";
        "v"; ">"; ">"; "v"]
```

Principes de la prog. dynamique

Problème

- $f : \mathcal{U} \to \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \max_{y \in \mathcal{U}}\{f(y)\} = M_{f,\mathcal{U}}$

Démarche

Principes de la prog. dynamique

Problème

- $f : \mathcal{U} \to \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \max_{y \in \mathcal{U}}\{f(y)\} = M_{f,\mathcal{U}}$

Démarche

1. Identifier une *sous-structure optimale* : si l'on connaît $x \in \mathcal{U}$ tel que $f(x) = M_{f,\mathcal{U}}$, alors de x on déduit des solutions $(x_i)_{i \in I}$ à des sous-problèmes de la forme « trouver $x_i \in \mathcal{U}_i$, tel que $f_i(x_i) = \max_{y \in \mathcal{U}_i}\{f(y)\} = M_{f_i,\mathcal{U}_i}$ ».

Principes de la prog. dynamique

Problème

- $f : \mathcal{U} \to \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \max_{y \in \mathcal{U}}\{f(y)\} = M_{f,\mathcal{U}}$

Démarche

1. Identifier une *sous-structure optimale* : si l'on connaît $x \in \mathcal{U}$ tel que $f(x) = M_{f,\mathcal{U}}$, alors de x on déduit des solutions $(x_i)_{i \in I}$ à des sous-problèmes de la forme « trouver $x_i \in \mathcal{U}_i$, tel que $f_i(x_i) = \max_{y \in \mathcal{U}_i}\{f(y)\} = M_{f_i,\mathcal{U}_i}$ ».
2. Déduire une relation récursive permettant le calcul de $M_{f,\mathcal{U}}$ à partir de certains $M_{f_i,\mathcal{U}_i}$.

Principes de la prog. dynamique

Problème

- $f : \mathcal{U} \to \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \max_{y \in \mathcal{U}}\{f(y)\} = M_{f,\mathcal{U}}$

Démarche

1. Identifier une *sous-structure optimale* : si l'on connaît $x \in \mathcal{U}$ tel que $f(x) = M_{f,\mathcal{U}}$, alors de x on déduit des solutions $(x_i)_{i \in I}$ à des sous-problèmes de la forme « trouver $x_i \in \mathcal{U}_i$, tel que $f_i(x_i) = \max_{y \in \mathcal{U}_i}\{f(y)\} = M_{f_i,\mathcal{U}_i}$ ».
2. Déduire une relation récursive permettant le calcul de $M_{f,\mathcal{U}}$ à partir de certains $M_{f_i,\mathcal{U}_i}$.
3. Un calcul récursif est en général inefficace : calculer itérativement en stockant certains $M_{f_i,\mathcal{U}_i}$.

Principes de la prog. dynamique

Problème

- $f : \mathcal{U} \to \mathbb{Z}$, $\mathcal{U}$ fini.
- But : trouver x tel que $f(x) = \max_{y \in \mathcal{U}}\{f(y)\} = M_{f,\mathcal{U}}$

Démarche

1. Identifier une *sous-structure optimale* : si l'on connaît $x \in \mathcal{U}$ tel que $f(x) = M_{f,\mathcal{U}}$, alors de x on déduit des solutions $(x_i)_{i \in I}$ à des sous-problèmes de la forme « trouver $x_i \in \mathcal{U}_i$, tel que $f_i(x_i) = \max_{y \in \mathcal{U}_i}\{f(y)\} = M_{f_i,\mathcal{U}_i}$ ».
2. Déduire une relation récursive permettant le calcul de $M_{f,\mathcal{U}}$ à partir de certains $M_{f_i,\mathcal{U}_i}$.
3. Un calcul récursif est en général inefficace : calculer itérativement en stockant certains $M_{f_i,\mathcal{U}_i}$.
4. Modifier le calcul pour trouver en même temps des x_i tels que $f_i(x_i) = M_{f_i,\mathcal{U}_i}$

Parenthèse : problèmes de combinatoire

Problème

Calculer le cardinal d'un certain ensemble $\mathcal{U}$. Pour certains problèmes, technique similaire.

Démarche

Problème

Calculer le cardinal d'un certain ensemble $\mathcal{U}$. Pour certains problèmes, technique similaire.

Démarche

1. Partitionner $\mathcal{U}$ en sous-ensembles disjoints $(\widetilde{\mathcal{U}_i})_i$, les $(\mathcal{U}_i)$ étant des ensembles associés à des « sous-problèmes combinatoires » et les $(\widetilde{\mathcal{U}_i})$ des ensembles en bijection avec les $(\mathcal{U}_i)$.

Parenthèse : problèmes de combinatoire

Problème

Calculer le cardinal d'un certain ensemble $\mathcal{U}$. Pour certains problèmes, technique similaire.

Démarche

1. Partitionner $\mathcal{U}$ en sous-ensembles disjoints $(\widetilde{\mathcal{U}_i})_i$, les $(\mathcal{U}_i)$ étant des ensembles associés à des « sous-problèmes combinatoires » et les $(\widetilde{\mathcal{U}_i})$ des ensembles en bijection avec les $(\mathcal{U}_i)$.
2. Écrire que $\mathcal{U} = \cup_i \widetilde{\mathcal{U}_i}$ (l'union étant disjointe) fournit une relation de récurrence permettant de calculer $|\mathcal{U}|$, à savoir $|\mathcal{U}| = \sum_i |\mathcal{U}_i|$.

Parenthèse : problèmes de combinatoire

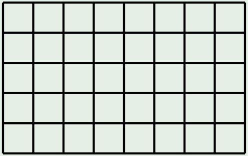
Problème

Calculer le cardinal d'un certain ensemble $\mathcal{U}$. Pour certains problèmes, technique similaire.

Démarche

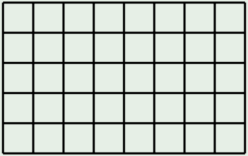
1. Partitionner $\mathcal{U}$ en sous-ensembles disjoints $(\widetilde{\mathcal{U}_i})_i$, les $(\mathcal{U}_i)$ étant des ensembles associés à des « sous-problèmes combinatoires » et les $(\widetilde{\mathcal{U}_i})$ des ensembles en bijection avec les $(\mathcal{U}_i)$.
2. Écrire que $\mathcal{U} = \cup_i \widetilde{\mathcal{U}_i}$ (l'union étant disjointe) fournit une relation de récurrence permettant de calculer $|\mathcal{U}|$, à savoir $|\mathcal{U}| = \sum_i |\mathcal{U}_i|$.
3. Calculer $|\mathcal{U}|$ itérativement, en faisant usage d'un tableau.

Nombre de chemins



Nombre de chemins partant du coin en haut à gauche jusqu'au coin en bas à droite, en suivant seulement les directions $\rightarrow$ et $\downarrow$?

Nombre de chemins

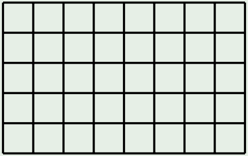


Nombre de chemins partant du coin en haut à gauche jusqu'au coin en bas à droite, en suivant seulement les directions $\rightarrow$ et $\downarrow$?

1. Indexation de la grille de $(0, 0)$ (en haut à gauche) à (n, m) (en bas à droite).

Parenthèse : problèmes de combinatoire

Nombre de chemins

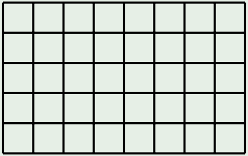


Nombre de chemins partant du coin en haut à gauche jusqu'au coin en bas à droite, en suivant seulement les directions $\rightarrow$ et $\downarrow$?

1. Indexation de la grille de $(0,0)$ (en haut à gauche) à (n,m) (en bas à droite).
2. $\mathcal{C}_{i,j}$ chemins de $(0,0)$ à (i,j)

Parenthèse : problèmes de combinatoire

Nombre de chemins

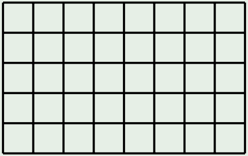


Nombre de chemins partant du coin en haut à gauche jusqu'au coin en bas à droite, en suivant seulement les directions $\rightarrow$ et $\downarrow$?

1. Indexation de la grille de $(0,0)$ (en haut à gauche) à (n,m) (en bas à droite).
2. $\mathcal{C}_{i,j}$ chemins de $(0,0)$ à (i,j)
3. Relation : $\mathcal{C}_{i,j} = \widetilde{\mathcal{C}_{i-1,j}} \cup \widetilde{\mathcal{C}_{i,j-1}}$
   où $\widetilde{\mathcal{C}_{i-1,j}}$ est l'ensemble des chemins de $\mathcal{C}_{i-1,j}$, complétés par le segment $(i-1,j) \rightarrow (i,j)$, et de même pour $\widetilde{\mathcal{C}_{i,j-1}}$.

Parenthèse : problèmes de combinatoire

Nombre de chemins

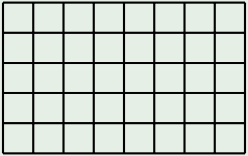


Nombre de chemins partant du coin en haut à gauche jusqu'au coin en bas à droite, en suivant seulement les directions $\rightarrow$ et $\downarrow$?

1. Indexation de la grille de $(0,0)$ (en haut à gauche) à (n,m) (en bas à droite).
2. $\mathcal{C}_{i,j}$ chemins de $(0,0)$ à (i,j)
3. Relation : $\mathcal{C}_{i,j} = \widetilde{\mathcal{C}_{i-1,j}} \cup \widetilde{\mathcal{C}_{i,j-1}}$
   où $\widetilde{\mathcal{C}_{i-1,j}}$ est l'ensemble des chemins de $\mathcal{C}_{i-1,j}$, complétés par le segment $(i-1,j) \rightarrow (i,j)$, et de même pour $\widetilde{\mathcal{C}_{i,j-1}}$.
4. Avec $N_{i,j} = |\mathcal{C}_{i,j}|$, relation $N_{i,j} = N_{i-1,j} + N_{i,j-1}$, valable pour $i,j > 0$, sinon $N_{i,0} = N_{0,j} = 1$.

Parenthèse : problèmes de combinatoire

Nombre de chemins

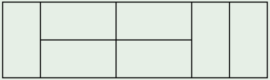


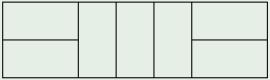
Nombre de chemins partant du coin en haut à gauche jusqu'au coin en bas à droite, en suivant seulement les directions $\rightarrow$ et $\downarrow$?

1. Indexation de la grille de $(0,0)$ (en haut à gauche) à (n,m) (en bas à droite).
2. $\mathcal{C}_{i,j}$ chemins de $(0,0)$ à (i,j)
3. Relation : $\mathcal{C}_{i,j} = \widetilde{\mathcal{C}_{i-1,j}} \cup \widetilde{\mathcal{C}_{i,j-1}}$
   où $\widetilde{\mathcal{C}_{i-1,j}}$ est l'ensemble des chemins de $\mathcal{C}_{i-1,j}$, complétés par le segment $(i-1,j) \rightarrow (i,j)$, et de même pour $\widetilde{\mathcal{C}_{i,j-1}}$.
4. Avec $N_{i,j} = |\mathcal{C}_{i,j}|$, relation $N_{i,j} = N_{i-1,j} + N_{i,j-1}$, valable pour $i,j > 0$, sinon $N_{i,0} = N_{0,j} = 1$.
5. On peut tabuler les $N_{i,j}$ dans un tableau de taille $(n+1) \times (m+1)$.

Parenthèse : problèmes de combinatoire

Nombre de pavages ?

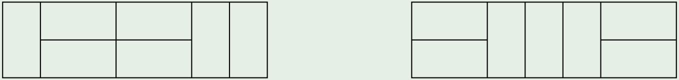




Nombre de pavages d'un rectangle $2 \times n$ avec des dominos 2×1 ?

1. F_n nombre de pavages possibles d'un rectangle $2 \times n$;

Nombre de pavages ?

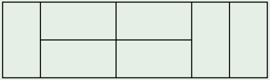


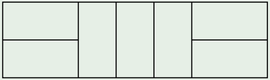
Nombre de pavages d'un rectangle $2 \times n$ avec des dominos 2×1 ?

1. F_n nombre de pavages possibles d'un rectangle $2 \times n$;
2. $F_0 = F_1 = 1$;

Parenthèse : problèmes de combinatoire

Nombre de pavages ?



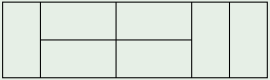
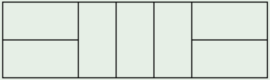


Nombre de pavages d'un rectangle $2 \times n$ avec des dominos 2×1 ?

1. F_n nombre de pavages possibles d'un rectangle $2 \times n$;
2. $F_0 = F_1 = 1$;
3. $F_n = F_{n-1} + F_{n-2}$ pour $n \geq 2$ $\rightsquigarrow$ Preuve !

Parenthèse : problèmes de combinatoire

Nombre de pavages ?

Nombre de pavages d'un rectangle $2 \times n$ avec des dominos 2×1 ?

1. F_n nombre de pavages possibles d'un rectangle $2 \times n$;
2. $F_0 = F_1 = 1$;
3. $F_n = F_{n-1} + F_{n-2}$ pour $n \geq 2 \rightsquigarrow$ Preuve !

Plus dur : rectangle $3 \times n$?

Glouton vs Dynamique

Retour sur le problème du chemin de poids max.

$$\text{Optimal :} \begin{pmatrix} \mathbf{2} & \mathbf{39} & \mathbf{12} & \mathbf{49} & \mathbf{47} & 18 & 22 & 19 \\ 37 & 21 & 34 & 26 & \mathbf{10} & 2 & 35 & 39 \\ 31 & 21 & 12 & 26 & \mathbf{34} & \mathbf{27} & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & \mathbf{40} & \mathbf{36} & \mathbf{13} \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & \mathbf{6} \end{pmatrix}$$

$$\text{Glouton :} \begin{pmatrix} \mathbf{2} & \mathbf{39} & 12 & 49 & 47 & 18 & 22 & 19 \\ 37 & \mathbf{21} & \mathbf{34} & \mathbf{26} & 10 & 2 & 35 & 39 \\ 31 & 21 & 12 & \mathbf{26} & \mathbf{34} & \mathbf{27} & 7 & 22 \\ 20 & 46 & 16 & 2 & 11 & \mathbf{40} & \mathbf{36} & \mathbf{13} \\ 18 & 30 & 32 & 37 & 28 & 24 & 9 & \mathbf{6} \end{pmatrix}$$

Meilleure complexité, mais ... non optimal pour ce problème.

Glouton vs Dynamique

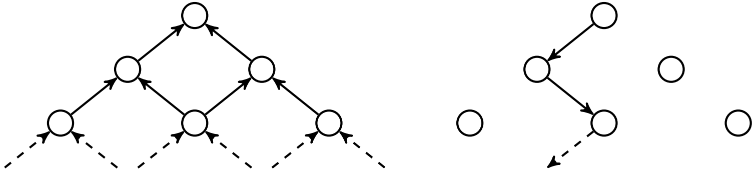
Principe des algorithmes gloutons

1. Identifier une sous-structure optimale ;
2. Trouver une relation de récurrence sur les $M_{f_i,\mathcal{U}_i}$;
3. Faire un choix localement optimal ramenant le calcul de $M_{f,\mathcal{U}}$ à un seul $M_{f',\mathcal{U}'}$.

Glouton vs Dynamique

Principe des algorithmes gloutons

1. Identifier une sous-structure optimale ;
2. Trouver une relation de récurrence sur les $M_{f_i,\mathcal{U}_i}$;
3. Faire un choix localement optimal ramenant le calcul de $M_{f,\mathcal{U}}$ à un seul $M_{f',\mathcal{U}'}$.



Stratégie dynamique   Stratégie gloutonne

Problème de la plus longue sous-séquence commune

Le problème

- si s chaîne de caractères, sous-séquence de s obtenue en prenant certains caractères de s en conservant l'ordre ;

Problème de la plus longue sous-séquence commune

Le problème

- si s chaîne de caractères, sous-séquence de s obtenue en prenant certains caractères de s en conservant l'ordre ;
- si s de taille n, 2^n sous-séquences sans éliminer les doublons ;

Problème de la plus longue sous-séquence commune

Le problème

- si s chaîne de caractères, sous-séquence de s obtenue en prenant certains caractères de s en conservant l'ordre ;
- si s de taille n, 2^n sous-séquences sans éliminer les doublons ;
- Problème : s et t deux chaînes, trouver une sous-séquence commune de taille maximale (plssc).

Problème de la plus longue sous-séquence commune

Le problème

- si s chaîne de caractères, sous-séquence de s obtenue en prenant certains caractères de s en conservant l'ordre ;
- si s de taille n, 2^n sous-séquences sans éliminer les doublons ;
- Problème : s et t deux chaînes, trouver une sous-séquence commune de taille maximale (plssc).

1. Sous-structure optimale : pour s et t non vides, et x une plssc :
    - si s et t terminent par le même caractère ...
    - sinon ...

Problème de la plus longue sous-séquence commune

Le problème

- si s chaîne de caractères, sous-séquence de s obtenue en prenant certains caractères de s en conservant l'ordre ;
- si s de taille n, 2^n sous-séquences sans éliminer les doublons ;
- Problème : s et t deux chaînes, trouver une sous-séquence commune de taille maximale (plssc).

1. Sous-structure optimale : pour s et t non vides, et x une plssc :
   - si s et t terminent par le même caractère ...
   - sinon ...
2. $\ell_{i,j}$: longueur de la plssc des préfixes de s et t de tailles i et j.

$$\ell_{i,j} = \begin{cases} 0 & \text{si } i = 0 \text{ ou } j = 0\,; \\ 1 + \ell_{i-1,j-1} & \text{si } s_{i-1} = t_{j-1}; \\ \max\{\ell_{i-1,j}, \ell_{i,j-1}\} & \text{sinon.} \end{cases}$$

Problème de la plus longue sous-séquence commune

Le problème

- si s chaîne de caractères, sous-séquence de s obtenue en prenant certains caractères de s en conservant l'ordre ;
- si s de taille n, 2^n sous-séquences sans éliminer les doublons ;
- Problème : s et t deux chaînes, trouver une sous-séquence commune de taille maximale (plssc).

1. Sous-structure optimale : pour s et t non vides, et x une plssc :
   - si s et t terminent par le même caractère ...
   - sinon ...
2. $\ell_{i,j}$: longueur de la plssc des préfixes de s et t de tailles i et j.

$$\ell_{i,j} = \begin{cases} 0 & \text{si } i = 0 \text{ ou } j = 0\,; \\ 1 + \ell_{i-1,j-1} & \text{si } s_{i-1} = t_{j-1}; \\ \max\{\ell_{i-1,j}, \ell_{i,j-1}\} & \text{sinon.} \end{cases}$$

3. Calcul itératif des $\ell_{i,j}$;

Problème de la plus longue sous-séquence commune

Le problème

- si s chaîne de caractères, sous-séquence de s obtenue en prenant certains caractères de s en conservant l'ordre ;
- si s de taille n, 2^n sous-séquences sans éliminer les doublons ;
- Problème : s et t deux chaînes, trouver une sous-séquence commune de taille maximale (plssc).

1. Sous-structure optimale : pour s et t non vides, et x une plssc :
   - si s et t terminent par le même caractère ...
   - sinon ...
2. $\ell_{i,j}$: longueur de la plssc des préfixes de s et t de tailles i et j.

$$\ell_{i,j} = \begin{cases} 0 & \text{si } i = 0 \text{ ou } j = 0\,; \\ 1 + \ell_{i-1,j-1} & \text{si } s_{i-1} = t_{j-1}; \\ \max\{\ell_{i-1,j}, \ell_{i,j-1}\} & \text{sinon.} \end{cases}$$

3. Calcul itératif des $\ell_{i,j}$;
4. Calcul effectif d'une plssc.