

Ensemble \mathcal{C}_n^k des parties à k éléments
de l'ensemble $\mathcal{E}_n = \{1, 2 \dots n\}$.

partie = sous-ensemble

$$\mathcal{E}_4 = \{ \quad , \quad , \quad , \quad \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \quad , \quad , \quad , \quad , \quad , \quad \}$$

$$\mathcal{E}_4 = \{1, 2, 3, 4\}$$

$$\mathcal{C}_4^2 = \{ \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\} \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,1\}, \quad , \quad , \quad , \quad , \quad \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\} \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \quad , \quad , \quad , \quad , \quad \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1,3\}, \{1,4\}, \quad , \quad , \quad \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1, 3\}, \{1, 4\}, \{2,}, \quad , \quad \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1, 3\}, \{1, 4\}, \{2, 1\}, \quad , \quad \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \quad , \quad \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \}$$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \}$$

La complexité de l'algorithme naïf qui construit toutes les parties à **deux** éléments d'un ensemble à n éléments est dans

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \}$$

La complexité de l'algorithme naïf qui construit toutes les parties à **deux** éléments d'un ensemble à n éléments est dans $\Theta(n^2)$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \}$$

La complexité de l'algorithme naïf qui construit toutes les parties à **trois** éléments d'un ensemble à n éléments est dans $\Theta()$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \}$$

La complexité de l'algorithme naïf qui construit toutes les parties à **trois** éléments d'un ensemble à n éléments est dans $\Theta(n^3)$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \}$$

La complexité de l'algorithme naïf qui construit toutes les parties à k éléments d'un ensemble à n éléments est dans $\Theta()$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \}$$

La complexité de l'algorithme naïf qui construit toutes les parties à k éléments d'un ensemble à n éléments est dans $\Theta(n^k)$

$$\mathcal{E}_4 = \{ 1, 2, 3, 4 \}$$

$$\mathcal{C}_4^2 = \{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \}$$

La complexité de l'algorithme naïf qui **imprime** toutes les parties à k éléments d'un ensemble à n éléments est dans $\Theta(n^k \times k)$

Peut-on faire mieux ?

Quel est le coût **minimum** de l'algorithme d'impression ?

Peut-on faire mieux ?

Quel est le coût **minimum** de l'algorithme d'impression ?

Coût d'impression d'un élément

×

Nombre d'éléments

Peut-on faire mieux ?

Quel est le coût **minimum** de l'algorithme d'impression ?

Coût d'impression d'un élément $\Theta(k)$

×

Nombre d'éléments

Peut-on faire mieux?

Quel est le coût **minimum** de l'algorithme d'impression?

Coût d'impression d'un élément $\Theta(k)$

×

Nombre d'éléments C_n^k

Peut-on faire mieux?

Quel est le coût **minimum** de l'algorithme d'impression ?

Coût d'impression d'un élément $\Theta(k)$

\times

Nombre d'éléments C_n^k

$$\Theta(k.C_n^k)$$

Peut-on faire mieux ?

Quel est le coût **minimum** de l'algorithme d'impression ?

Coût d'impression d'un élément $\Theta(k)$

×

Nombre d'éléments C_n^k

$$\Theta(k.C_n^k)$$

L'algorithme naïf est en $\Theta(k \times n^k)$, le coût minimum est en $\frac{n!}{(k-1)!(n-k)!}$

algorithme naïf : $\Theta(k \times n^k)$

coût minimum : $\frac{n!}{(k-1)!(n-k)!}$

$$\frac{n!}{(n-k)!} =$$

algorithme naïf : $\Theta(k \times n^k)$

coût minimum : $\frac{n!}{(k-1)!(n-k)!}$

$$\frac{n!}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{(n-k)!} =$$

algorithme naïf : $\Theta(k \times n^k)$

coût minimum : $\frac{n!}{(k-1)!(n-k)!}$

$$\frac{n!}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{1 \times 2 \times \dots \times (n-k)}$$

algorithme naïf : $\Theta(k \times n^k)$

coût minimum : $\frac{n!}{(k-1)!(n-k)!}$

$$\frac{n!}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{1 \times 2 \times \dots \times (n-k)}$$

$$= \frac{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k}) \times (n-k+1) \times \dots \times n}{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k})} =$$

algorithme naïf : $\Theta(k \times n^k)$

coût minimum : $\frac{n!}{(k-1)!(n-k)!}$

$$\frac{n!}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{1 \times 2 \times \dots \times (n-k)}$$

$$= \frac{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k}) \times (n-k+1) \times \dots \times n}{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k})} = \prod_{i=0}^{k-1} (n-i)$$

algorithme naïf : $\Theta(k \times n^k)$

coût minimum : $\frac{n!}{(k-1)!(n-k)!}$

$$\frac{n!}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{1 \times 2 \times \dots \times (n-k)}$$

$$= \frac{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k}) \times (n-k+1) \times \dots \times n}{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k})} = \prod_{i=0}^{k-1} (n-i)$$

n^k est beaucoup plus grand que $\prod_{i=0}^{k-1} (n-i)$

algorithme naïf : $\Theta(k \times n^k)$

coût minimum : $\frac{n!}{(k-1)!(n-k)!}$

$$\begin{aligned}\frac{n!}{(n-k)!} &= \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{1 \times 2 \times \dots \times (n-k)} \\ &= \frac{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k}) \times (n-k+1) \times \dots \times n}{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k})} = \prod_{i=0}^{k-1} (n-i)\end{aligned}$$

n^k est beaucoup plus grand que $\prod_{i=0}^{k-1} (n-i)$

Donc le rapport du coût de l'algorithme naïf au coût minimal

algorithme naïf : $\Theta(k \times n^k)$

coût minimum : $\frac{n!}{(k-1)!(n-k)!}$

$$\begin{aligned}\frac{n!}{(n-k)!} &= \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{1 \times 2 \times \dots \times (n-k)} \\ &= \frac{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k}) \times (n-k+1) \times \dots \times n}{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k})} = \prod_{i=0}^{k-1} (n-i)\end{aligned}$$

n^k est beaucoup plus grand que $\prod_{i=0}^{k-1} (n-i)$

Donc le rapport du coût de l'algorithme naïf au coût minimal

$$\frac{k \times n^k}{\frac{\prod_{i=0}^{k-1} (n-i)}{(k-1)!}} =$$

algorithme naïf : $\Theta(k \times n^k)$

coût minimum : $\frac{n!}{(k-1)!(n-k)!}$

$$\begin{aligned} \frac{n!}{(n-k)!} &= \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{1 \times 2 \times \dots \times (n-k)} \\ &= \frac{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k}) \times (n-k+1) \times \dots \times n}{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k})} = \prod_{i=0}^{i=k-1} (n-i) \end{aligned}$$

n^k est beaucoup plus grand que $\prod_{i=0}^{i=k-1} (n-i)$

Donc le rapport du coût de l'algorithme naïf au coût minimal

$$\frac{\frac{k \times n^k}{\prod_{i=0}^{i=k-1} (n-i)}}{(k-1)!} = \frac{n^k}{\prod_{i=0}^{i=k-1} (n-i)} \times k! \text{ est supérieur à } k!$$

algorithme naïf : $\Theta(k \times n^k)$

coût minimum : $\frac{n!}{(k-1)!(n-k)!}$

$$\begin{aligned} \frac{n!}{(n-k)!} &= \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{(n-k)!} = \frac{1 \times 2 \times \dots \times (n-k) \times (n-k+1) \times \dots \times n}{1 \times 2 \times \dots \times (n-k)} \\ &= \frac{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k}) \times (n-k+1) \times \dots \times n}{\cancel{1} \times \cancel{2} \times \dots \times (\cancel{n-k})} = \prod_{i=0}^{k-1} (n-i) \end{aligned}$$

n^k est beaucoup plus grand que $\prod_{i=0}^{k-1} (n-i)$

Donc le rapport du coût de l'algorithme naïf au coût minimal

$$\frac{k \times n^k}{\frac{\prod_{i=0}^{k-1} (n-i)}{(k-1)!}} = \frac{n^k}{\prod_{i=0}^{k-1} (n-i)} \times k! \text{ est supérieur à } k!$$

Ça vaut le coup de tenter quelque chose par exemple quand $k \approx \frac{n}{2}$