Les principes fondamentaux du dénombrement (suite)

11 septembre 2018

1. permutations
   - exemple
   - factorielle
   - exemple
   - arrangement

réponse à la question posée au cours précédent

26	$\times$		$\times$		$\times$		$\times$		$\times$	
1^{re} position		2^{e} pos.		3^{e} pos.		4^{e} pos.		5^{e} pos.		6^{e} pos.

Ce qui donne 3276000 numéros possibles.

réponse à la question posée au cours précédent

26	$\times$	25	$\times$		$\times$		$\times$		$\times$	
1re position		2e pos.		3e pos.		4e pos.		5e pos.		6e pos.

Ce qui donne 3276000 numéros possibles.

réponse à la question posée au cours précédent

26	$\times$	25	$\times$	10	$\times$		$\times$		$\times$	
1re position		2e pos.		3e pos.		4e pos.		5e pos.		6e pos.

Ce qui donne 3276000 numéros possibles.

réponse à la question posée au cours précédent

26	×	25	×	10	×	9	×		×	
1re position		2e pos.		3e pos.		4e pos.		5e pos.		6e pos.

Ce qui donne 3276000 numéros possibles.

réponse à la question posée au cours précédent

26	$\times$	25	$\times$	10	$\times$	9	$\times$	8	$\times$	
1re position		2e pos.		3e pos.		4e pos.		5e pos.		6e pos.

Ce qui donne 3276000 numéros possibles.

réponse à la question posée au cours précédent

26	$\times$	25	$\times$	10	$\times$	9	$\times$	8	$\times$	7
1re position		2e pos.		3e pos.		4e pos.		5e pos.		6e pos.

Ce qui donne 3276000 numéros possibles.

exemple

Dans un groupe de dix étudiants, cinq ont décidé de s'asseoir sur une rangée pour prendre une photo. De combien de manières peuvent-ils s'installer ?

première position $\times$ $\times$ $\times$ $\times$

exemple

Dans un groupe de dix étudiants, cinq ont décidé de s'asseoir sur une rangée pour prendre une photo. De combien de manières peuvent-ils s'installer ?

première position	$\times$	deuxième position	$\times$		$\times$		$\times$	

exemple

Dans un groupe de dix étudiants, cinq ont décidé de s'asseoir sur une rangée pour prendre une photo. De combien de manières peuvent-ils s'installer ?

première position	$\times$	deuxième position	$\times$	troisième position	$\times$		$\times$	

exemple

Dans un groupe de dix étudiants, cinq ont décidé de s'asseoir sur une rangée pour prendre une photo. De combien de manières peuvent-ils s'installer ?

première position	$\times$	deuxième position	$\times$	troisième position	$\times$	quatrième position	$\times$

exemple

Dans un groupe de dix étudiants, cinq ont décidé de s'asseoir sur une rangée pour prendre une photo. De combien de manières peuvent-ils s'installer ?

première position	$\times$	deuxième position	$\times$	troisième position	$\times$	quatrième position	$\times$	cinquième position

exemple

Dans un groupe de dix étudiants, cinq ont décidé de s'asseoir sur une rangée pour prendre une photo. De combien de manières peuvent-ils s'installer ?

10								
première position	$\times$	deuxième position	$\times$	troisième position	$\times$	quatrième position	$\times$	cinquième position

exemple

Dans un groupe de dix étudiants, cinq ont décidé de s'asseoir sur une rangée pour prendre une photo. De combien de manières peuvent-ils s'installer ?

10	$\times$	9	$\times$		$\times$		$\times$	
première position		deuxième position		troisième position		quatrième position		cinquième position

exemple

Dans un groupe de dix étudiants, cinq ont décidé de s'asseoir sur une rangée pour prendre une photo. De combien de manières peuvent-ils s'installer ?

10	$\times$	9	$\times$	8	$\times$		$\times$	
première position		deuxième position		troisième position		quatrième position		cinquième position

exemple

Dans un groupe de dix étudiants, cinq ont décidé de s'asseoir sur une rangée pour prendre une photo. De combien de manières peuvent-ils s'installer ?

$$\underset{\text{première position}}{10} \times \underset{\text{deuxième position}}{9} \times \underset{\text{troisième position}}{8} \times \underset{\text{quatrième position}}{7} \times \underset{\text{cinquième position}}{6}$$

factorielle

definition

Pour tout entier $n \geq 0$, *factorielle n* (notée $n!$) est défini par

- $0! = 1$,
- Pour tout entier $n \geq 1$, $n! = (1)(2)(3)...(n-2)(n-1)(n)$.

$$n! = \prod_{i=1}^{n} i$$

de plus

$$(n+1)! = (n+1)n!$$

Un grand classique.

Dans vos études d'informatique, il sera difficile de dénombrer le nombre de fois que vous implanterez cette fonction !

une implantation en python


```
def factorielle(n):
    res=1
    for i in range(1,n+1):
      res = res * i
    return res
```

les dix premieres valeurs

```
>>> [factorielle(i) for i in range(10)]
[1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880]
```

Un autre trésor

OEIS : The On-Line Encyclopedia of Integer Sequences® (OEIS®)https://oeis.org/

retour à l'exemple précédent

le nombre

retour à l'exemple précédent

le nombre

$$\frac{10 \times 9 \times 8 \times 7 \times 6}{}$$

peut s'écrire $\frac{10!}{5!}$

retour à l'exemple précédent

le nombre

$$\frac{10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1}{5 \times 4 \times 3 \times 2 \times 1}$$

peut s'écrire $\frac{10!}{5!}$

definition

Étant donné une collection de n objets différents, tout arrangement utilisant l'ensemble de ces objets est appelé permutation de la collection.

On appelle arrangement de k objets différents choisis parmi n le nombre noté A_n^k et prononcé "a n k" egal à

$$\underset{\substack{\text{première}\\\text{position}}}{n} \times \underset{\substack{\text{deuxième}\\\text{position}}}{n-1} \times \underset{\substack{\text{troisième}\\\text{position}}}{n-2} \times \dots \times \underset{\substack{\text{k-ième}\\\text{position}}}{n-k+1} = \frac{n!}{(n-k)!}$$

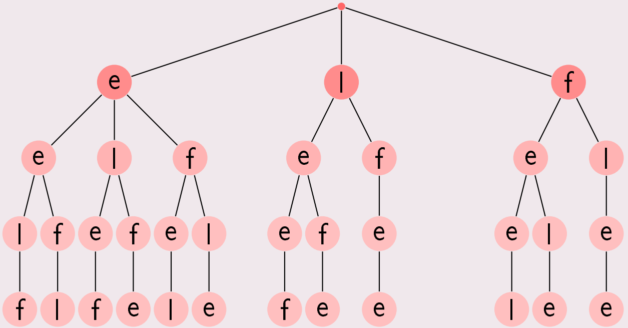
exemple

Le nombre de permutations des lettres du mot PORTABLE est 8! Si seulement quatre lettres du mot PORTABLE sont utilisées. Le nombre d'arrangements est A_8^4.

Si on s'autorise les répétitions...

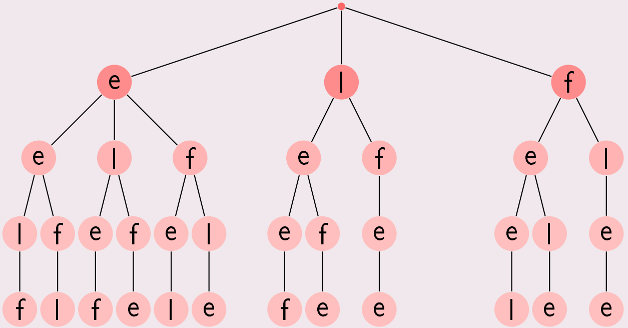
exemple

Le nombre de permutations des lettres du mot ELFE n'est pas 4 ! car il y a deux E qu'on ne doit pas distinguer.



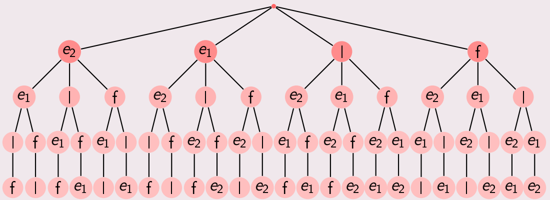
exemple

Le nombre de permutations des lettres du mot ELFE n'est pas 4 ! car il y a deux E qu'on ne doit pas distinguer.



exemple

Faisons le choix de distinguer les deux E, on obtient alors l'arbre suivant



Dans cet arbre, on s'aperçoit que lorsqu'on « oublie » les numéros chaque mot est obtenu deux fois

En Utilisant l'idée développée dans l'exercice précédent, on regarde le nombre de mots qu'on peut écrire en utilisant toutes lettres du mot LILLIAD

Une idée est de considerer que les lettres sont distinctes, c'est à dire les remplacer par L, I, L, L, I, A, D par $L_1, I_1, L_2, L_3, I_2, A, D$ de compter le nombre de permutations de ces 7 symboles distincts.
Après il faut « oublier » les numéros.
Seulement il y a plusieurs manières de réaliser le mot LILLIAD avec $L_1, I_1, L_2, L_3, I_2, A, D$ par exemple $L_2 I_2 L_1 L_3 I_1 AD$. de combien de façons peut on écrire LILLIAD à partir de $L_1, I_1, L_2, L_3, I_2, A, D$? .

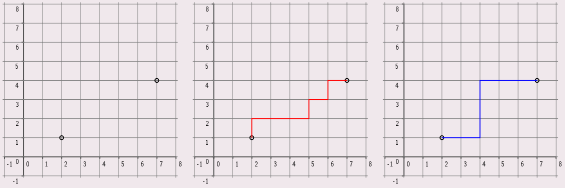
- $7!$ permutations des symboles $L_1, l_1, L_2, L_3, l_2, A, D$
- $3!$ permutations de L_1, L_2, L_3
- $2!$ permutations de l_1, l_2
- $$\frac{7!}{3!2!}$$

Plus généralement, s'il y a n objets n_1 d'un premier type, n_2 d'un second type,..., et s'il y a n_r d'un r[e]type où $n_1 + n_2 + ... + n_r = \text{n}$ alors il y a

$$\frac{n!}{n_1!n_2!...n_r!}$$

arrangements de ces objets
et qu'on ne peut pas distinguer les objets de même type

Nombre de chemins



Déterminez le nombre de chemins allant de $(2, 1)$ à $(7, 4)$ tel que

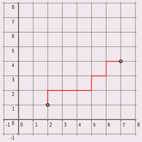
- l'abscisse soit croissante (en fonction du temps)
- l'ordonnée soit croissante
- toujours, on conserve entière l'une des coordonnées.

C'est à dire qu'on reste sur le quadrillage et qu'on fait des pas soit vers la droite soit vers le haut

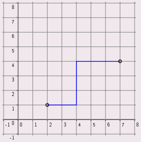
Grâce à un codage

On transforme le problème de comptage des chemins, en un problème de comptage de mots.

- on code les pas à droite par D,
- on code les pas vers le haut par H.



(a) 'HDDDHDHD'



(b) 'DDHHHDDD'

Cela revient donc à compter les mots de 8 lettres sur l'alphabet $\{D, H\}$ qui contiennent exactement 5 occurences de D et 3 occurences de H.

$$\frac{8!}{5!3!}$$

exercice de programmation

écrire une fonction python nommée `enum_chemin` paramétrée par les coordonnées des deux points (4 entiers) et qui renvoie la liste de tous les mots par exemple

```
>>> enum_chemin(2,1,3,4)
        ['DHHH', 'HDHH', 'HHDH', 'HHHD']
```

Indication (attendre d'avoir vu la récursivité en AP2