

1 Valeurs des coefficients binomiaux $\binom{n}{p}$

Donner tous les coefficients binomiaux pour une valeur de n

1.1 L'idée mathématique

On utilise la relation (ou le triangle de Pascal) .

1.2 La mise en algorithme

Variables

M ,L : listes

i , k : entiers

Début de l'algorithme

$M \leftarrow [1, 1]$

$L \leftarrow [1, 1]$

Pour i allant de 2 à n Faire

$M[i + 1] \leftarrow 1$

Pour k allant de 0 à i - 2 Faire

$M[k + 1] \leftarrow L[k + 1] + L[k]$

FinPour

$L \leftarrow M$

FinPour

Sorties :

Afficher L

```

1 def coeffbinom(n):
2     L=[1,1]
3     M=L
4     for i in range(2,n+1):
5         M=M+[1]
6         for k in range(0,i-1):
7             M[k+1]=L[k+1]+L[k]
8         L=M
9     return L
    
```



Attention

- Quand on note $L[k]$, on donne le $k + 1$ ème terme de la liste .

2 Générer des permutations

On donne une liste et on veut obtenir toutes les permutations possibles

2.1 Le principe

On utilise en python une fonction récursive , c'est à dire qu'on utilise la fonction qu'on fabrique au rang inférieur .

Si on prend la liste [1,2,3] , on fixe 1 et on cherche les permutations de [2,3] , puis on fixe 2 et on cherche les permutations de [1,3]

Les deux premières lignes traitent la liste vide et la liste à un seul élément

2.2 La mise en algorithme

```
1 def perm(L):
2   →if not L:
3   → →return []
4   →elif len(L)==1:
5   → →return [list(L)]
6   →else:
7   → →return [[L[i]]+P for i in range(len(L)) for P in perm(L[:i]+L[i+1:])]
```



A retenir

L[:i] signifie les ièmes premiers termes de L et L[i+1 :] tous les termes de L sauf les (i + 1)èmes .

3 Déterminer des parties à 2 éléments d'un ensemble à n éléments

On veut déterminer , à partir d'une liste de n éléments , toutes les paires possibles

Par exemple , [1,2,3] donnera : [1,2] , [1,3] , [2,3]

On fixe le premier terme , puis on lui associe tous les autres termes possibles et ainsi de suite et comme l'ordre ne compte pas , on démarre en $i + 1$

3.1 La mise en algorithme

Variables

P , L : listes

i , k , n : entiers

Début de l'algorithme

Saisir L

$n \leftarrow$ longueur de la liste L

$P \leftarrow [0, 0]$

Pour i allant de 1 à n - 1 **Faire**

$P[0] \leftarrow L[i - 1]$

Pour k allant de i + 1 à n **Faire**

$P[1] \leftarrow L[k - 1]$

 Afficher P

FinPour

FinPour

```
1 def paires(L):
2     n=len(L)
3     P=[0,0]
4     for i in range (1,n):
5         P[0]=L[i-1]
6         for k in range(i+1,n+1):
7             P[1]=L[k-1]
8             print(P)
9     return
```