

Le triangle de Pascal est une présentation des coefficients binomiaux sous la forme d'un triangle. Les coefficients binomiaux notés $\binom{n}{k}$ ou C_n^k , sont définis pour tout entier naturel n et tout entier naturel $k \leq n$, par $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. Ils donnent le nombre de parties de k éléments dans un ensemble de n éléments.

Les objectifs de cette activité sont :

- Construire le triangle de Pascal.
- Étudier la complexité des algorithmes mis en œuvre.

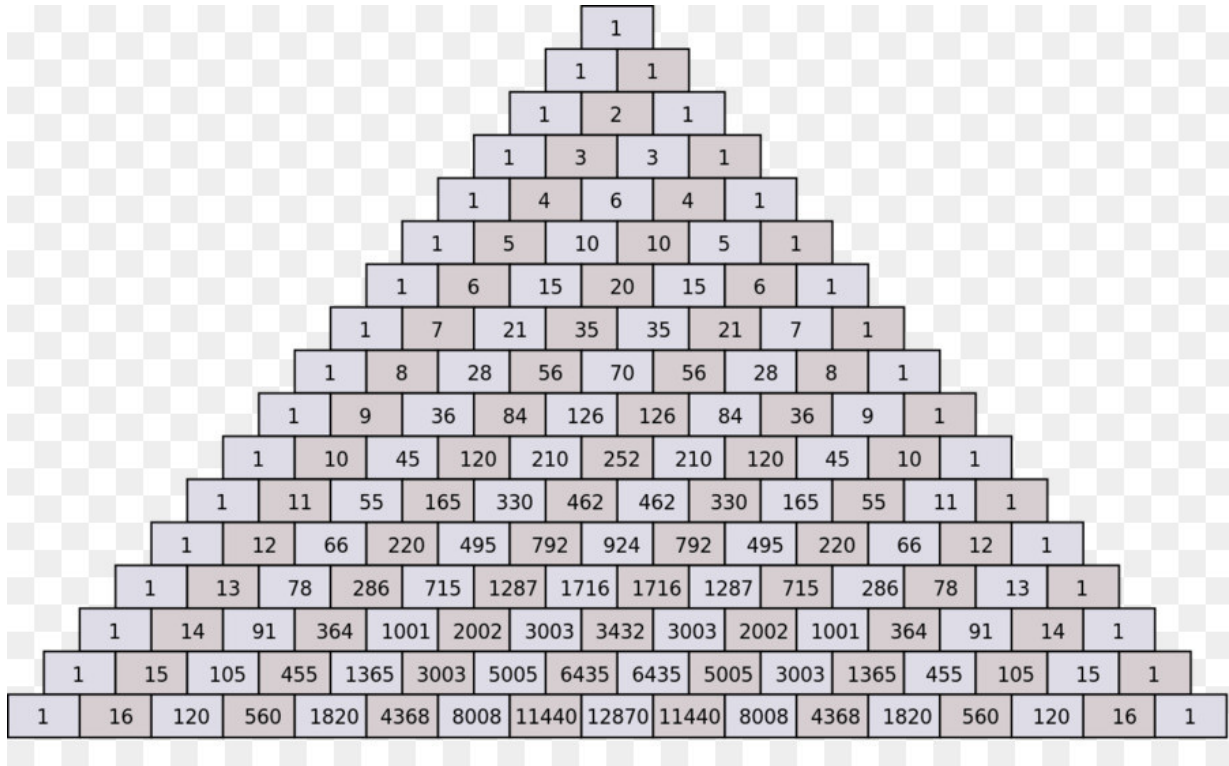


Figure 1 – Triangle de Pascal

- Factorielle :
 - Définir une fonction **fact(n)** qui prend en argument un entier positif ou nul et qui renvoie $n!$.
 - Justifier que l'algorithme est correct.
 - Quelle est la complexité temporelle de cet algorithme? et sa complexité en espace?
- Coefficients binomiaux :
 - Définir une fonction **binomial(n,k)** renvoyant $\binom{n}{k}$.
 - Quelle est la complexité temporelle de cet algorithme?
- Triangle de Pascal :
 - Définir une fonction **ListeCnk(n)** qui renvoie la liste des coefficients binomiaux pour n donné.
 - Quelle est la complexité temporelle de cet algorithme?
 - Définir une fonction **ListePascal(n)** qui renvoie une liste dont chaque élément est la liste des éléments d'une ligne du triangle de Pascal.
Exemple : **ListePascal(3)** renvoie `[[1], [1,1], [1,2,1], [1,3,3,1]]`.
- Durée de calcul :
 - Définir une fonction **moyenne(valeurs)** prend en argument une liste de valeurs (type 'list') et qui renvoie la moyenne arithmétique de ces valeurs.
 - Insérer la fonction **DureesFonction**, définie ci-après, dans votre script Python. L'importation de la bibliothèque `time` se fera par la commande :

```
from time import perf_counter as clock
```

```
def DureesFonction ( fonction , nmin , nmax , pas , moyennage=1):  
    """ documentation """  
    durees = []  
    n = []  
    for i in range ( int ( nmin ) , int ( nmax ) + 1 , int ( pas ) ) :  
        n . append ( i )  
        t = []  
        for k in range ( moyennage ) :  
            ti = clock ()  
            fonction ( i )  
            tf = clock ()  
            t . append ( tf - ti )  
        durees . append ( moyenne ( t ) )  
    return n , durees
```

- (c) Que fait la fonction `DureesFonction` ?
- (d) À l'aide de cette fonction, analyser graphiquement la durée des calculs pour exécuter les algorithmes `fact`, `ListeCnk` et `ListePascal` en fonction la valeur de n .