Combinatoire2020-Eleves

November 30, 2020

1 Exemples du chapitre Combinatoire

- Lien vers le site de M.Junier http://www.frederic-junier.org/TS2021/Progression/TS_2021.html
- Lien vers l'activité Geogebra sur les listes : https://www.geogebra.org/m/phsnu6s3

1.1 Capacité 5 : k-uplets distincts

[1]:
```
def kuplets_distincts(n, k):
    """Retourne le nombre de k-uplets distincts
    d'un ensemble à n éléments"""
    c = 1
    for i in range(0, k):
        c =c * (n - i)
    return c
```

[2]:
```
kuplets_distincts(8, 3)
```

[2]:
```
336
```

1.2 Algorithmique 1 : factorielle de n

[2]:
```
def fact(n):
    f = 1
    for k in range(1, n + 1):
        f = f * k
    return f
```

[3]:
```
[[n, fact(n)] for n in range(0, 7)]
```

[3]:
```
[[0, 1], [1, 1], [2, 2], [3, 6], [4, 24], [5, 120], [6, 720]]
```

1.3 Manipulations de listes en Python, exemples du cours

Voir activité Geogebra https://www.geogebra.org/m/phsnu6s3

1.4 Algorithmique 2 : tirage aléatoire d'une permutation

Version du cours.

[28]:
```python
from random import randint

def generer_perm(n):
    """Tirage aléatoire d'une permutation de [1;n]"""
    perm = []
    urne = list(range(1, n + 1))
    for k in range(n):
        index_aleatoire = randint(0, len(urne) - 1)
        choix = urne.pop(index_aleatoire)
        perm.append(choix)
    return perm
```

[29]:
```python
generer_perm(6)
```

[29]:
```
[6, 3, 2, 1, 5, 4]
```

Version du TP1 Indice p.40

[23]:
```python
from random import randint

def generer_perm_Indice(n):
    """Génère une permutation de l'ensemble [1,n]
    On simule n tirages successifs sans remise
    """
    perm = []
    urne = list(range(1, n + 1))
    for k in range(n):
        choix = urne.pop(randint(0, n - 1 - k))
        perm.append(choix)
    return perm
```

[24]:
```python
generer_perm_Indice(6)
```

[24]:
```
[4, 2, 1, 3, 6, 5]
```

Générer les parties à 2 ou 3 éléments d'un ensemble fini

- Indice TP2 page 40
- Hyperbole exo 14 page 64

[231]:
```python
def generer_partie2(n):
    L = []
    for i in range(1, n):
        for j in range(i + 1, n + 1):
            L.append([i, j])
    return L
```

[232]:
```
generer_partie2(4)
```

[232]:
```
[[1, 2], [1, 3], [1, 4], [2, 3], [2, 4], [3, 4]]
```

[233]:
```
def generer_partie3(n):
    L = []
    for i in range(1, n - 1):
        for j in range(i + 1, n):
            for k in range(j + 1, n + 1):
                L.append([i, j, k])
    return L
```

[234]:
```
generer_partie3(4)
```

[234]:
```
[[1, 2, 3], [1, 2, 4], [1, 3, 4], [2, 3, 4]]
```

[238]:
```
def generer_partie(n, p):
    """Retourne la liste de parties à p éléments de [1,n]
    Version récursive"""
    if p > n or p  < 0:
        return []
    if n == 0 or p == 0:
        return [[]]
    return generer_partie(n-1,p) + [partie + [n] for partie in generer_partie(n␣
 ↪- 1,p-1)]
```

[236]:
```
generer_partie(4, 3)
```

[236]:
```
[[1, 2, 3], [1, 2, 4], [1, 3, 4], [2, 3, 4]]
```

[237]:
```
generer_partie(4, 2)
```

[237]:
```
[[1, 2], [1, 3], [2, 3], [1, 4], [2, 4], [3, 4]]
```

Coefficients binomiaux

Différentes fonctions qui renvoie la liste des coefficients binomiaux $\binom{n}{k}$

[8]:
```
def blaise(n):
    L = [1, 1]
    for i in range(2, n + 1):
        M = L + [1]
        for k in range(i - 1):
            M[k+1] = L[k+1] + L[k]
        L = M
    return L
```

[9]:
```
blaise(7)
```

[9]:
```
[1, 7, 21, 35, 35, 21, 7, 1]
```

[10]:
```
def binom(n):
    """Voir Hyperbole exo 11 page 63"""
    L = [1]
    for i in range(1, n + 1):
        M = L
        M.append(1)
        for j in range(i-1, 0, -1):
            M[j] = L[j-1] + L[j]
        L = M
    return L
```

[11]:
```
binom(7)
```

[11]:
```
[1, 7, 21, 35, 35, 21, 7, 1]
```

[12]:
```
def binomHyperbole(n):
    """Voir Hyperbole exo 11 page 63"""
    L = [1]
    for i in range(1, n + 1):
        M = L[:]
        M.append(1)
        #les 2 lignes précédentes sont équivalentes à M = L + [1]
        for j in range(1, i):
            M[j] = L[j-1] + L[j]
        L = M
    return L
```

[13]:
```
binomHyperbole(7)
```

[13]:
```
[1, 7, 21, 35, 35, 21, 7, 1]
```

[14]:
```
def binomHyperbole2(n):
    """Voir Hyperbole exo 11 page 63, c'est simple"""
    L = [1]
    for i in range(1, n + 1):
        M = L + [1] #copie de la ligne précédente à laquelle on ajoute 1
        for k in range(i - 1):
            M[k+1] = L[k+1] + L[k]
        L = M
    return L
```

[15]:
```
binomHyperbole2(7)
```

[15]: [1, 7, 21, 35, 35, 21, 7, 1]

[16]:
```python
def binomIndice(n):
    """Voir Indice Capacité 11 page 23, trop compliqué"""
    L = [1]
    for i in range(1, n + 1):
        L.append(1)
        a = L[0]
        b = L[1]
        for j in range(1, i):
            L[j] = a + b
            a = b
            b = L[j + 1]
    return L
```

[18]:
```python
binomIndice(7)
```

[18]: [1, 7, 21, 35, 35, 21, 7, 1]