

Combinaisons

Objectif

Exemple pour des combinations de 3 objets pris 2 à 2. Soit une liste d'objets [12,3]:

on veut connaître toutes les fuçons de les présenter 2 à 2, sans tenir compte de l'ordre: [[1,2], [1,3], [2,3]].

en temant compte si nécessaires de répétitions

• et on veut savoir combien il y en a.

NB: Contrairement aux permutatio

Nombre de combinaisons de n objets pris k à k

Le calcul du nombre de combinaisons de n objet

 $Cnk = \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2)...(n-k+1)}{k!}$

Soit, en Python (si fact(n)=factorielle de n): $Cnk = fact(n)((fact(k)^*|fact(n-k)) = n^*(n-1)^*(n-2)^*...^*(n-k-1)/fact(k)$

Puisque nous sommes sous Python qui sait calculer des nombres entiers énormes, nous po

Il est difficile de faire plus simple...

Mais le plus efficace est quand même le suivant, qui calcule progressivement le résultat sans calculer des grands

Madia le prince Chi spenie de la codificación de la

Attention à ne pas utiliser dans ce code le ra

Pour que cette fonction soit utilisable avec de très grands nombres, j'ai tenu compte des points suivants:

If aut utiliser while au lieu de for, car xrange() ne supporte pas les types 'long' (en contrepartie de sa ra
si k-n/2, alors combin(n,k) est remplacé par combin(n, n-k) qui donne le même résultat avec moins de c

print combin(3,2) # affiche 3 print combin(50,20) # affiche 4712921224

Coccole est tre in public pare exemple, le nombre de combinations de 100000 (cent mille) objets pris 100 à 100 donne en moiss d'1/1000 de seconde un nombre de 543 chiffres.

101976/1186808235416461754118605454886051591 68371777912242211176707023464150011986189177544218070341757002179101792692712005500177548407807975484517019250

Solution récursive

La solution récursive est particulièrement simple:

The continuous continuous area of the property of the continuous area of

off conducts, by:

""month of conducts are made on a diplets pris k a k (non record), made setted be record)?"

""by a ([0,4])

"in the setted of conductions do n objets pris k a k (non record), made setted be record)?"

"bill a ([0,5])

"bill pile.append([i-1,j-1])
pile.append([i-1,j])
return r

- A chaque boucle while, on dépile le dernier couple [i,j]. Les 2 seuls couples dont on c remplace le couple [i,j] par les 2 couples [i-1,j-1] et [i-1,j] qu'on empile.

La boucle s'arrête quand la pile est vide! et le

La boucle s'urrête quand la pile est videt et le résultet r est remoye Woyons cela sur un exemple. On ajoute juste après le while la ligne print r, sile en aliance l'exemple: print contact, 3.) Ce qui affiche:

Ce qui opirione.

0 [15, 31]
0 [14, 21, 14, 23]
1 [16, 21, 22, 22, 23, 31]
1 [16, 21, 22, 23]
1 [16, 21, 22, 23]
2 [16, 21, 22, 23]
2 [16, 21, 17, 02, 11, 11]
2 [16, 21, 22, 23]
2 [16, 21, 17, 02, 11, 11]
4 [16, 21]
4 [16, 21]
4 [16, 21]
5 [16, 21, 17, 02, 21]
6 [17, 11, 12, 23]
6 [17, 11, 12, 23]
6 [17, 11, 12, 13]
6 [17, 11, 12, 13]
6 [17, 11, 12, 13]
6 [17, 11, 12, 13]
6 [17, 11, 12, 13]
6 [17, 11, 12, 13]
6 [17, 11, 12, 13]
6 [17, 11, 11, 13]
6 [17, 11, 11, 13] - (10, 1)]
7 [[2, 0], [2, 1]]
7 [[2, 0], [1, 0], [1, 1]]
8 [[2, 0], [1, 0]]
9 [[2, 0]]
10

Le résultat final est ok, mais par rapport au code non-récursif précédent, celui-ci donne une lenteur __ afflig A ne conserver que pour des considérations théoriques.

Liste des combinaisons d'une liste de n objets pris k à k

Nous savons maintenant calculer le **nombre** de combinaisons, nous voulons maintenant en établir la **liste**.

Par exemple, nous voulons trouver la liste de toutes les combinaisons de de 3 objets [1,2,3] pris 2 à 2, qui est: [[1,2], [1,3], [2,3]]

Per exemple, nous vaulours remover la liste de toutes les combonisons de de 3 objets l' Nous allours reprode text simplement le principe de Tensemble des parties d'un en Cela domerta comme code: or contestierates, 1): . 1, 1000 e 0, 27000(40):11 . 1, 1000 e 0, 27000(40):11

Exemples d'utilisa

print conductant([W,W,C],2) # wittine [[W, W], [W, C], [W, C], [W, C]]
print conductant([W,W,C],2) # wittine [[W, W], [W, C], [W, C], [W, C]]
print conductant([W,W,C,C],2) # wittine [[W, W], [W, C], [W, C],

Liste des combinaisons d'une chaîne de n caractères pris k à k

C'est le même principe, à part que la donnée est une chaîne de n caractères, et qu'on cherche toutes les con

La solution la plus simple est d'utiliser la fonction précédente: def combinchaine(ch,k):
 return [''.join(x) for x in combinliste(list(ch),k)]

Exemple d'utilisation

print combinchaine('ABCD',3) ['ACD', 'ABD', 'ABC', 'BCD']

Combinaisons de n objets distincts pris k à k avec répétition

Pour la théorie, voir par exemple: http://fr.wikipedia.org/wiki/Combination_avec_ris_C7%A5piAC5%A5tition_ ict_ on a n objets distincts, mais on west se voir répérés dans les résultats des combinations de ces n objets pr Pur exemple, [1,2,3] ~ [[1,1,1,1,2,1,3,3,1,2,3,[2,3,3]] alors que sans répétitions: [[1,2,1,1,3,1,2,3]]

Nombre de combinaisons de n objets distincts pris k à k avec répétition

Ce nombre est égal à la combinaison de (n+k-1) objets pris k à k.

D'ou la fonction:

Exemple pour [1,2,3] pris 2 à 2 avec répétition:

-Liste des combinaisons de n objets distincts pris k à k avec répétiti

expansion de la liste dutiale, de sorte que chacum de su objets soit répéte k fois. Par exemple pour [1,23] et k-2 ~ [1,12,2,33]

 calcul de la liste des combinaisons "normales" de cette nouvelle liste comportant k*n objets pris k à k

oortant k*n objets pris k à k

Could donne commes code:

of conditioning on the conditioning new reportion due days to say pris & & ***

***Tenness is listed on conditioning new reportion due days to say pris & & ***

***Tenness is listed on conditioning of the conditioning of

print combinlisterep([1,2,3],2) [[1, 1], [1, 2], [1, 3], [2, 2], [2, 3], [3, 3]]

```
Liste des combinaisons d'une chaîne de n caractères pris k à k avec répétition
 On va simplement utiliser la fonction précédente:

def combinchainerep(ch,k):

return [''.join(k) for x in combinisterep(list(ch),k)]
Combinaisons de n objets non tous distincts pris k à k avec ou sans répétition
```