Informatique théorique

- Séance 3
- Dénombrement et suites récurrentes

Plan du cours

1- Notion de dénombrement
2- Permutations
3- Arrangements
4- Combinaisons
5- Suites récurrentes

I- Dénombrement : notion

- L'étude des nombres de permutations, d'arrangements, de combinaisons ou de partitions, s'appelle traditionnellement en mathématiques l'analyse combinatoire.
- Elle est omniprésente en informatique :La résolution de nombreux problèmes consiste en
  - l'énumération exhaustive des possibilités pour ensuite
  - décider pour chacune si elle est solution ou non au problème.

  Et avant d'énumérer, il est prudent de dénombrer.
- Plusieurs autres raisons à cette omniprésence :
  - codage des données en binaire
  - propriétés combinatoires des structures de données
  - programmation itérative ou récursive
  - estimation du temps de calcul des algorithmes en fonction de la taille des entrées

I- Dénombrement: exemple du temps de calcul

- Par exemple, on considère un programme qui va lister toutes les parties d'un ensemble E à n éléments. Il y en a 2^n.
- Supposons pour simplifier que pour calculer et afficher une partie, l'ordinateur prenne une μ-seconde (10^{-6} s) :

|E|	0	10	20	30	40	50	60
Temps	1 µs	1,024 ms	1,048 s	17,9 mn	12,7 jours	35,7 ans	366 siècles

I- Dénombrement: exemple du voyage de commerce

- Toutes les villes d'une région sont reliées deux à deux.
- Le VRP habite dans l'une et doit visiter plusieurs clients, se trouvant dans chacune des villes voisines.
- On cherche le chemin le plus court lui permettant de parcourir toutes les villes avant de retourner dans la sienne.
- On pense à un algorithme naïf :
  - on énumère tous les parcours possibles
  - on sélectionne le plus court.
- Le seul hic est que pour n un peu grand, la réponse de l'ordinateur peut prendre … des années ou des siècles !

II- Permutations : définition

- Soit E un ensemble fini à n éléments. On appelle permutation p de E toute bijection de E dans E.
- Le nombre de permutations de E est égal à n!
  - Il y a (n>1):n choix pour le 1er élément de la permutation,
  - n-1 choix pour le 2nd
  - …
  - 2 choix pour l'avant dernier
  - 1 seul pour le dernier.
  - On multiplie le nombre de toutes ces possibilités, soit : n!
- Une permutation est aussi une suite ordonnée sans répétition ni omission d'éléments de E.

II- Permutations : Factorielle (2)

- L'ordre de grandeur de la factorielle est n^n.
- Une approximation de la factorielle est donnée par la formule de Stirling

$$n\,! = \sqrt{2\pi n}\,(n/e)^n\,(1+\,\theta(1/n))$$

- Exemple
  - L'algorithme précédent doit énumérer toutes les permutations des villes du voisinage.
  - Pour n villes, il y a n-1 villes voisines donc (n-1)! parcours, d'où un temps d'exécution excessif de l'algorithme.
- A ce jour, on ne connaît pas d'algorithme permettant de résoudre ce problème en un temps acceptable quand n est très grand.

III- Arrangement : notion

- Soit E un ensemble fini à n éléments. On appelle arrangement de p éléments de E avec p≤n toute suite ordonnée et sans répétition de p éléments de E.
- Nombre d'arrangements de p éléments (p≠0) de E :

$$A_n^p = \frac{n\,!}{(n-p)!}$$

  - Par convention, $A_n^0 = 1$
  - Un arrangement de p éléments pris parmi n peut être vu comme les p premiers éléments d'une permutation de n éléments.
  - Chaque arrangement de p éléments donne donc lieu à (n-p)! permutations.
- Exemple
  - Pour 12 chevaux au départ, il y a 1320 tiercés possibles.

III- Arrangement: avec répétition

- Soit E un ensemble fini à n éléments. On appelle arrangement avec répétition de p éléments de E toute application de {1,…,p} dans E.
- On en compte le nombre d'arrangements : $\mathbf{n^p}$
  - Il y a : n choix pour le 1er élément
  - n choix pour le 2d
  - …
  - n choix pour le p ième.
  - On multiplie le nombre de toutes ces possibilités, soit: n^p.
- Exemple : Combien existe-il d'octets ? 2^8.
- On peut en déduire le nombre d'entiers de type byte, short, int et long en Java, sachant qu'ils sont codés sur 1, 2, 4 et 8 octets.

IV- Combinaisons: notion

- On appelle combinaison de p éléments pris parmi n éléments dans E toute partie de E à p éléments.
  - Sauf avis contraire, les combinaisons sont sans répétition.
- Nombre de combinaisons de p éléments pris parmi n :

$$C_n^p = \frac{n\,!}{p!(n-p)!}$$

- Une combinaison ne tient pas compte de l'ordre des éléments ainsi un arrangement de p éléments pris parmi n donne lieu à p! combinaisons.
- Exemple
  - On pourrait vérifier que la somme du nombre des combinaisons pour les p variant de 0 à n redonne bien 2^n, le cardinal des parties de E.

IV- Combinaisons: propriétés

- Par convention, $C_n^p = 0$ pour tout p<0 ou p>n.
- Il est immédiat que $C_n^p = C_n^{n-p}$
- Triangle de Pascal

1							
1	1						
1	2	1					
1	3	3	1				
1	4	6	4	1			
1	5	10	10	5	1		
1	6	15	20	15	6	1	
1	7	21	35	35	21	7	1
...							

Le **triangle de Pascal** est un arrangement géométrique des coefficients binomiaux dans un triangle. À la ligne *i* et colonne *j* ($0 \leq j \leq i$) est placé le coefficient binomial

Exemple i=2 :
$(X + 1)^2 = 1X^2 + 2X + 1^2$

IV- Combinaisons: propriétés (2)

- Les C_n^p ont aussi appelés les coefficients binomiaux. Ils sont obtenus après développement du binôme de Newton
$$(a + b)^n = \Sigma_{0\leq i\leq n} C_n^i a^i b^{n-i}$$
- Preuve : par Récurrence sur n
  - n=0, $(a + b)^0 = C_0^0 = 1$
  - n=1, $a + b = C_1^0 a^0 b + C_1^1 a b^0$
  - on suppose l'égalité vraie à l'ordre n.
  - Calculons à l'ordre n+1 :
  - $(a+b)^{n+1} = (a+b)^n (a+b) = (\Sigma_{0\leq i\leq n} C_n^i a^i b^{n-i}) (a+b)$ par hyp.réc.
  $$= \Sigma_{0\leq i\leq n} C_n^i a^{i+1} b^{n-i} (a+b) + \Sigma_{0<i<n} C_n^i a^i b^{n+1-i}$$
  - Le coefficient de $a^i b^{n+1-i}$ pour $1\leq i\leq n$ est $C_n^{i-1} + C_n^i = C_{n+1}^i$
  - Celui de an+1 est $\underline{C_n^n = C_{n+1}^{n+1}}$
  - et celui de bn+1 est bien $\underline{C_n^0 = C_{n+1}^0}$

IV- Combinaisons: avec répétitions

- Soit E un ensemble fini à n éléments. On appelle combinaison avec répétition de p éléments pris parmi n éléments de E toute application f de E dans {1,…,p} qui vérifie f (e1) + … + f(en) = p
- On en dénombre C_{n+p-1}^{n-1}
- C'est aussi le nombre de solutions entières de l'équation $x_1 + \ldots + x_n = p$
- Exemple
  - Sur l'alphabet {a,b,c}, on associe à chaque mot l'ensemble des lettres qui le constituent.
  - Il y a 9 mots de 2 lettres : aa, ab, ac, ba, bb, bc, ca, cb, cc et ensembles associés [a,a],[a,b],[a,c],[b,b],[b,c] ,[c,c].
  - On considère les occurrences $|w|_a$, $|w|_b$ et $|w|_c$ de chaque lettre.
  - La somme des occurrences est bien égale à la longueur du mot.

IV- Combinaisons: avec répétitions (2)

- Pour dénombrer les combinaisons avec répétition de p éléments pris parmi $\{e_1, \ldots ,e_n\}$, on a recours au calcul suivant :
  - On considère n+p-1 emplacements pour y placer n-1 marqueurs.
  - Le nombre d'emplacements vides entre les ième et (i+1)ème marqueurs indique le nombre de répétitions de l'élément e_{i+1}.
  - Celui avant le premier marqueur indique le nombre de e_1.
  - Celui après le (n-1)ème et dernier marqueur indique le nombre de e_n.
  - il y a bien C_{n+p-1}^{n-1} possibilités.
- Exemple : Les 6 groupes de 2 lettres [a,a],[a,b],[a,c],[b,b],[b,c] et [c,c] sur {a,b,c} se schématisent ainsi :
  - [a,a] _ _ √ √ [a,b] _ √ _ √ [a,c] _ √ √ _
  - [b,b] √ _ _ √ [b,c] √ _ √ _ [c,c] √ √ _ _

IV- Combinaisons: nombre de partitions

- Soit E un ensemble fini à n éléments. On effectue une partition de E en k ensembles non vides $(E_i)_{0 \le i \le k}$ de cardinaux respectifs $(n_i)_{0 \le i \le k}$.
- Nombre de partitions possible $\frac{n!}{n_1!\, n_2! \ldots n_k!}$

- Preuve par récurrence
  - k = 1 : (n!) / n1 C_n^m
  - k = 2 : (n!) / n1! n2!

  Si n1=m et n2=n-m : C_n^m = (n!) / m!(n-m)!, ok.
  - On suppose la propriété vraie à l'ordre 2 et à l'ordre k. A-t-on le nombre de partitions en k+1 classes $((E_i)_{0 \le i \le k+1}$ égal à (n!) / n_1! … n_k! n_{k+1}! ?
  - Considérons les partitions en (E_1, E_2 , … , $E_k \cup E_{k+1}$) il y en a, par hypothèse de récurrence :

  (n!) / x_1! … (x_k + x_{k+1})!
  - Par hypothèse de récurrence, le nombre de possibilités de répartir les éléments de $X_k \cup X_{k+1}$ en X_k et X_{k+1} est $(x_k + x_{k+1})! / x_k! x_{k+1}!$
  - Il reste à multiplier ces possibilités.

IV- Combinaisons: Coefficients multi-nominaux

On note $C_n^{n_1, n_2, \ldots, n_k}$ le nombre de partitions d'un ensemble E à n éléments en k ensembles non vides $(E_i)_{0 \le i \le k}$ de cardinaux respectifs $(n_i)_{0 \le i \le k}$.

$$C_n^{n_1, n_2, \ldots, n_k} = \frac{n!}{n_1!\, n_2! \ldots n_k!}$$

Ce nombre est appelé <u>coefficient multinomial</u>. Il apparaît en effet dans le développement :

$$(X_1 + X_2 + \ldots + X_k)^n = \Sigma_{n1+n2+\ldots nk=n} C_n^{n_1, n_2, \ldots, n_k} X_1^{n1} X_2^{n2} \ldots X_k^{nk}$$

IV- Combinaisons: Exemple

- Considérons 50 invités à un banquet. Combien y a-t-il de façons de les répartir par tables, sachant qu'il y a 3 tables de 8 personnes, 2 tables de 11 et 1 table de 4?
  - On cherche ici en fait le nombre de partitions d'un ensemble à 50 éléments en 6 parties non vides de cardinaux respectifs 8,8,8, 11, 11 et 4.
  - La répartition par tables achevée, on pourrait encore se demander combien y a-t-il de possibilités de disposer les invités…

V- Suites récurrentes : motivation

- Les relations de récurrence munies de conditions initiales permettent de définir des suites de nombres.
- En informatique, elles proviennent essentiellement :
  - des définitions inductives
  - des stratégies du type « diviser pour régner!» (divide and conquer) utilisant la récursivité.
- exemple : la dichotomie
  - Les équations de récurrence sont aux mathématiques discrètes ce que les équations différentielles sont aux mathématiques continues.
- On peut les résoudre par différentes techniques selon les types de récurrence.

V- Suites récurrentes : Exemples

- Définition inductive de l'ensemble E des mots sur l'alphabet {0,1} n'ayant pas de 0 consécutifs.
  - (B) $\varepsilon \in E$, $0 \in E$
  - (I) si $m \in E$ alors $m1 \in E$ et $m10 \in E$
- Remarque : cette définition n'est pas ambiguë !
- On cherche la fonction de comptage des mots de longueur n.
  - Pour obtenir un mot de longueur n avec n>1,
    - soit on ajoute un 1 à la fin d'un mot de longueur n-1
    - soit on ajoute 10 en fin d'un mot de longueur n-2
  - Donc pour n>1 : f(n) = f(n-1) + f(n-2)
    - Conditions initiales : f(0) = 1, f(1) = 2.
    - Il reste à résoudre l'équation de récurrence…

V- Suites récurrentes : Exemples (2)

- Ensemble A des additions booléennes sur l'alphabet {0,1,+} :
  - (B) $0 \in A$, $1 \in A$
  - (I) si $u,v \in A$ alors $u+v \in A$
- Remarque : cette définition est ambiguë ! Attention !

- Fonction de comptage g. Si n est pair, g(n) = 0 (par induction).
  - On a la condition initiale g(1) = 2.
  - Pour construire un mot de longueur n, n=2k+1 et k>1 :
    - Soit on prend un u de longueur 1 et un v de longueur 2k-1
    - Soit on prend un u de longueur 3 et un v de longueur 2k-3
    - (…)
    - Soit on prend un u de longueur 2k-1 et un v de longueur 1
  - On a envie d'écrire que $g(n) = \Sigma_{1 \leq i \leq k}\ g(2i-1)\ g(n-2i)$
  - C'est faux !
- On a voulu «coller» à la définition inductive, mais elle est ambiguë et du coup, on a compté plusieurs fois certains mots…

V- Suites récurrentes : Exemples (3)

- rappelle de définition ambiguë de l'ensemble A des additions binaires sur l'alphabet {0,1,+} :
  - (B) $0 \in A$, $1 \in A$
  - (I) si $u,v \in A$ alors $u+v \in A$
- • Donnons une autre définition inductive, non ambiguë cette fois :
  - (B') $0 \in A$, $1 \in A$
  - (I') si $u \in A$ alors $u+0 \in A$ et $u+1 \in A$
- Quelle est la fonction de comptage g ?
  - conditions initiales :

  $g(0) = 0$

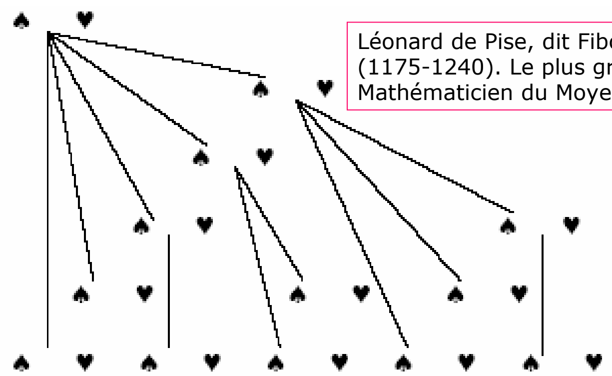
  $g(1) = 2$
  - pour $n>1$, $g(n) = 2.g(n-2)$

  on reconnaît une suite géométrique: $n=2k+1$, $k>0$, $g(n)=2k+1$

V. Suite(s) de Fibonacci

Léonard de Pise, dit Fibonacci (1175-1240). Le plus grand Mathématicien du Moyen-Age.

- $F_1=F_2= 2$
- $F_3 = 4$
- $F_4 = 6$
- $F_5 = 10$
- $F_6 = 16$
- $F_7 = 26$



- en fait, pour $\boxed{n>2,\ F_n = F_{n-1} + F_{n-2}}$

V. Suite(s) de Fibonacci (2)

- Si F1=0 et F1=1, on a pour tout n :

$$F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right)$$

- Ceci est le terme général de la suite, c'est-à-dire la valeur de F_n en fonction de n seulement. Pour l'obtenir, il faut résoudre l'équation de récurrence.
- Comment ? Ca dépend du type de l'équation. On distingue:
  - les équations linéaires, comme dans le cas des deux exemples présentés
  - les équations polynomiales
  - les équations non polynomiales.

V. Suite(s) de Fibonacci (3)

- Dans une relation de récurrence, le terme T(n) est fonction d'un certain nombre de termes précédents :

$$T(n) = f(\{T(p),\ n_0 \leq p < n\})$$

- Si n_0 = n-k, on dit que la relation de récurrence est d'ordre k.
  - Si T(n) ne dépend que de T(n-1), on parle d'ordre 1.
  - Si T(n) dépend de T(n-1) et T(n-2), on parle d'ordre 2.
- Si pour tous les termes T(p), p<n interviennent dans T(n), on parle de relation de récurrence complète.
- Quand f est une combinaison linéaire des T(p), on parle de relation de récurrence linéaire.

V. Simplification d'une relation de récurrence

- Une équation de récurrence non linéaire peut être ramenée à une relation linéaire par passage au logarithme.
- Par exemple, à l'ordre 1, $T(n) = a(n) (T(n-1))^p$ devient: $S(n) = \log_2 (T(n))$
  - $S(n) = \log_2 (a(n) (T(n-1))p)$
  - $S(n) = \log_2 (a(n)) + p \log_2 (T(n-1))$
  - $S(n) = \log_2 (a(n)) + p\, S(n-1)$
- Il ne reste plus qu'à résoudre l'équation linéaire puis à,effectuer l'opération inverse: $T(n) = 2^{S(n)}$

V. Simplification d'une relation de récurrence (2)

- L'idée pour résoudre une équation de récurrence complète est de réduire son ordre.

  $T(n) = a + b(n) + c(n) \Sigma_{0\leq i<n} T(i)$
- Une telle relation peut être ramenée à une relation linéaire en effectuant la différence des termes T(n) et T(n-1).

  $T(n) - T(n-1) = a + b(n) + c(n) \Sigma_{0\leq i<n} T(i) - a - b(n-1) - c(n-1) \Sigma_{0\leq i<n-1} T(i)$
- Si c(n) est constant et égal à c, cela donne :
  - $T(n) - T(n-1) = b(n) - b(n-1) + c\, T(n)$
  - $T(n) (1-c) = b(n) - b(n-1) + c\, T(n-1)$

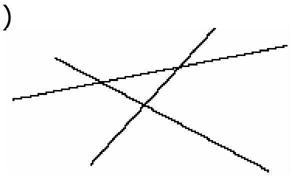
V- Equations linéaires d'ordre 1

$$T(n) = a(n)\ T(n-1) + b(n)$$

- On utilise la technique de sommation de plusieurs équations de récurrence.
- Après calcul et simplification, on obtient le terme général.
- Différents cas de figures se présentent, selon que:
  - a(n) est constant et b est nul, on a alors affaire à une suite géométrique
  - a(n) = 1, a(n) est constant ou non
  - b(n) est constant ou non.
- On va étudier cette technique directement sur des exemples. Ils ont valeur de généralité.

V- Résolution d'une équations linéaires d'ordre 1

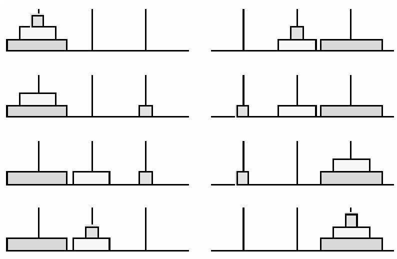
- On trace n droites (2 à 2 sécantes) dans le plan. Elles délimitent un ensemble de régions finies ou infinies.
- Quel est le nombre de régions T(n) ainsi délimitées ?
  - T(0) = 1, T(1) = 2, T(2) = 4, T(3) = 7…
  - On réalise que T(n) = T(n-1) + n.
  - (cas a(n) = 1 et b(n) dépend de n)
  - T(n) = T(n-1) + n
  - T(n-1) = T(n-2) + n-1
  - T(n-2) = T(n-3) + n-2
  - …
  - T(1) = T(0) +1



- $T(n) = T(0) + \Sigma_{1 \le i \le n}\ i = 1 + n(n+1)/2$

V- Résolution d'une équations linéaires d'ordre 1

- Jeu des tours de Hanoï



$$
\begin{aligned}
T(n) &= 2\,T(n-1) + 1 \\
2 \times T(n-1) &= (2\,T(n-2) + 1) \times 2 \\
2^2 \times T(n-2) &= (2\,T(n-3) + 1) \times 2^2 \\
&\dots \\
2^{n-1} \times T(1) &= (2\,T(0) + 1) \times 2^{n-1} \\
\hline
T(n) &= 2^n\,T(0) + \Sigma_{0 \le i \le n-1} 2^i \\
&= 2^n - 1
\end{aligned}
$$

- $T(0) = 0$, $T(1) = 1$, $T(2) = 3$, $T(3) = 7$…
- On réalise que $n>0$, $T(n) = 2\,T(n-1) + 1$.

(cas $a(n) = 2$, $b(n) = 1$, $a(n)$, $b(n)$ ne dépendent pas de n)

V- Résolution d'une équation de partition

- Appelons $T(n)$ le nombre de comparaisons nécessaires à la recherche dichotomique d'un élément dans un tableau trié de taille n.
    - On suppose $n=2^k$. $T(1) = 1$
    - pour $n>1$, $T(n) = 1 + T(n/2)$
  - On effectue un changement de variable et passe de n à k.
    - $T(2^0) = 1$
    - $k>0$, $T(2^k) = 1 + T(2^{k-1})$
  - On renomme la relation de récurrence concernant k :
    - $S(0) = 1$
    - pour $k>0$, $S(k) = 1 + S(k-1)$
  - On résout cette équation. Ici, c'est une simple suite arithmétique: $S(k) = k+1$
- Il reste à opérer le changement de variable inverse :

$T(n) = 1+\log_2(n)$

V- Equations linéaires d'ordre 2

$T(n) = a(n)\, T(n-1) + b(n)\, T(n-2) + c(n)$

- Dans le cas d'une équation homogène ($c(n)=0$) et à coefficients

constants ($a(n)$ et $b(n)$ sont constants), on utilise la technique de l'équation caractéristique, issue de la résolution des équations différentielles : $T(n) = a\, T(n-1) + b\, T(n-2)$

- On admet que l'ensemble des solutions forment un espace vectoriel de dimension inférieure à 2 et qu'une solution est de la forme α^n.
  - $\alpha^n = a\, \alpha^{n-1} + b\, \alpha^{n-2}$
  - On divise par α^{n-2}, on obtient l'équation d'inconnue α : $\alpha^2 = a\, \alpha + b$
- Dans le cas d'une racine réelle double, les solutions sont du type : $T(n) = \alpha^n (c_1 + c_2\, n)$
- Dans le cas de deux racines réelles, on a la solution :

$T(n) = c_1\, \alpha_1^{\,n} + c_2\, \alpha_2^{\,n}$

V- Equations linéaires d'ordre 2 : exemple

Suite de Fibonacci :

$F_0 = 0, \quad F_1 = 1, \qquad F_n = F_{n-1} + F_{n-2}$

Equation caractéristique :

$$\alpha^2 - \alpha - 1 = 0$$

$$\Delta = (-1)^2 - 4(-1) = 5 > 0$$

Les deux racines réelles sont :

$$\alpha_1 = \frac{1 + \sqrt{5}}{2}$$

$$\alpha_2 = \frac{1 - \sqrt{5}}{2}$$

Ensemble des solutions de la forme :

$$F_n = c_1\, \alpha_1^{\,n} + c_2\, \alpha_2^{\,n}$$

Il reste à déterminer c_1 et c_2 …

V- Equations linéaires d'ordre 2 : exemple (2)

Conditions initiales :

$$F_0 = 0 = c_1 + c_2$$
$$F_1 = 1 = c_1\,\alpha_1 + c_2\,\alpha_2$$

Calcul des c_1 et c_2 :

$$c_1 = -\,c_2$$
$$1 = c_1\,(\alpha_1 - \alpha_2)$$
$$1 = c_1\sqrt{5}$$
$$c_1 = 1/\sqrt{5} \text{ et } c_2 = -1/\sqrt{5}$$

Ensemble des solutions : pour tout n,

$$F_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$