IFT436 - Série d'exercices #1 : un peu de mathématiques discrètes

Manuel Lafond

Exercices mathématiques

Exercice 1. Démontrez l'identité $\binom{n}{2} = \binom{n-1}{2} + \binom{n-1}{1}$. Essayez d'utiliser un argument purement combinatoire avec pratiquement aucun calcul. On peut le prouver avec l'expression qui utilise les factoriels, mais cette voie n'est pas recommandée car vous n'aprendrez rien.

Exercice 2. Montrez que pour tout $n \in \mathbb{N}$ avec $n > 1$, il existe un nombre premier p tel que p divise n.

Exercice 3. Montrez que pour tout entier $n \geq 1$, $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

Exercice 4. Montrez que $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$.

Exercice 5. Soit X un ensemble contenant n éléments. Combien existe-t-il de sous ensembles différents de X?

Exercice 6. Montrez que $\sum_{i=0}^{n} \binom{n}{i} = 2^n$. On peut probablement le faire par induction, mais je recommande d'utiliser des arguments logiques dans une preuve directe.

Exercice 7. Nous allons prouver la forme close pour exprimer $\binom{n}{k}$. Commençons par nous échauffer avec l'exemple de la Loto.

1. À la Loto 6/49, une tirage résulte en un ensemble de 6 nombres différents choisis au hasard parmi 49. On pense souvent que le nombre de combinaisons possibles est $49 \cdot 48 \cdot 47 \cdot 46 \cdot 45 \cdot 44$ car il y a 49 choix pour le premier numéro, puis 48 pour le second, et ainsi de suite. Mais c'est en fait

$$\frac{49 \cdot 48 \cdot 47 \cdot 46 \cdot 45 \cdot 44}{6!}$$

   Pourquoi?

2. Montrez que $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. On peut faire une preuve directe en n'utilisant que des arguments en mot.

Exercice 8. Dans cet exercice, nous allons expliquer pourquoi $\binom{n}{k}$ s'appelle le coefficient binomial.

1. Donnez la valeur de $\binom{3}{0}, \binom{3}{1}, \binom{3}{2}$ et $\binom{3}{3}$.

2. Calculez l'expansion de $(x+1)^3$, c'est-à-dire, donnez une expression sous la forme $ax^3 + bx^2 + cx^1 + dx^0$. Quels sont les valeurs des coefficients a, b, c et d? Il devrait y avoir un lien avec la question a).

3. Donnez une preuve directe de l'égalité

$$(x+1)^n = \sum_{i=0}^{n} \binom{n}{i} x^i$$

en généralisant vos observations ci-haut.

4. Pendant que vous y êtes, convainquez-vous que vous pourriez généraliser votre argument pour prouver que $(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^i y^{n-i}$.

Exercice 9. Montrez que pour deux ensembles X et Y, $|X \cup Y| = |X| + |Y| - |X \cap Y|$.

Exercice 10. Dans cet exercice, nous allons compter des fonctions.

1. Une fonction de décision sur les n premiers entiers est une fonction $f : \{0, 1, \ldots, n\} \to \{T, F\}$ qui associe chaque entier entre 0 et n à T ou à F (*true* ou $false$). Combien existe-t-il de fonctions de décision différentes sur les n premiers entiers?

2. On dénote par $\{0,1\}^n$ l'ensemble de toutes les séquences de n bits. Une fonction booléenne $f : \{0,1\}^n \to \{0,1\}$ associe chaque séquence de n bits à 0 ou à 1. Combien existe-t-il de fonctions booléennes différentes sur n bits?

3. Une fonction $f : X \to Y$ de X vers Y est *bijective* si (1) pour tout $x_1, x_2 \in X$, $f(x_1) \neq f(x_2) \Leftrightarrow x_1 \neq x_2$ et si (2)pour tout $y \in Y$, il existe $x \in X$ tel que $f(x) = y$. En mots, chaque élément de X "pointe" un élément de Y différent, et chaque élément de Y est "pointé" par un élément.

   Combien y a-t-il de fonctions bijectives par rapport à $|X|$?

Exercice 11. Montrez que si sept nombres distincts sont choisis parmi les nombres $1, 2, 3, \ldots, 11$, alors la somme de deux de ces nombres est 12.

Exercice 12. Supposez que $n \in \mathbb{N}$ n'est pas un nombre premier. Montrez par contradiction que n a un diviseur d tel que $d \leq \sqrt{n}$.

Exercice 13. Montrez que dans une fête avec 6 personnes, il y a un groupe de 3 personnes qui sont amies mutuellement, ou bien un groupe de 3 personnes qui ne sont pas amies mutuellement.

Exercices algorithmiques

Exercice 14. Montrez que le tri par insertion trie correctement.

Exercice 15. Donnez une preuve par contre-exemple de l'énoncé suivant: le tri Sous-gradué présenté en classe ne résout pas le problème du tri correctement.

Exercice 16. On peut exprimer le temps requis par le tri fusion par une fonction de récurrence. Lorsque la fonction $triFusion(T)$ est appelée avec un tableau de n éléments, calculer le temps requis demande de sommer le temps requis par les deux appels récursifs sur les moités de T, plus un temps $O(n)$ pour faire la fusion. Ce temps $t(n)$ peut s'approximer avec la fonction suivante:

$$t(n) = \begin{cases} 2 & \text{si } n = 2 \\ 2t(n/2) + n & \text{si } n = 2^k, k > 1 \end{cases}$$

Montrez que $t(n) = n \log n$ pour tout $n = 2^k$, avec $k \geq 1$ un entier.

Exercice 17. Considérons le problème de recherche suivant.
Entrée : Une séquence triée de nombres $A = (a_1, a_2, \ldots, a_n)$ et un nombre v.
Sortie : $true$ si A contient la valeur v, et $false$ sinon.

La recherche dichotomique résout ce problème en temps $O(\log n)$. Voici la version récursive de l'algorithme.

```
fonction rechercheDicho(A, v)
    if |A| == 0 then
        return false;
    mid = n/2 + 1 ;                                  // n = taille de A
    if A[mid] == v then
        return true;
    if A[mid] < v then
        return rechercheDicho(A[mid..n], v);
    else
        return rechercheDicho(A[1..mid − 1], v);
```

On suppose que n est une puissance de 2, i.e. $n = 2^k$ pour un certain k.

Une *comparaison* est une opération qui compare un élément de A avec v. L'instruction $A[mid] == v$ est une comparaison, et l'instruction $A[mid] < v$ est aussi une comparaison. Les autres instructions ne sont pas des comparaisons.

Soit $c(n)$ le nombre de comparaisons effectuées par cet algorithme sur un tableau de taille n, dans le pire cas.

Montrez que $c(n) = 2\log(n) + 1$.