Aspects mathématiques dans les concours de programmation

B. Meyer

Cours INF280, département INFRES

Contenu

En quelques mots

- Nombreuses occurrences de questions mathématiques (parfois noyées dans la narration).
- Réponse immédiate ou rapide si on connaît ses formules, sinon alternative en force brute.
- Favorise des compétiteurs cultivés.
- Difficile de faire un cours structuré.

Méthodes ad hoc

Thèmes fréquents :

- Recherche de formules ou de motifs,
  (Ex : UVa10161, UVa11231)
- Suites de nombres,
  (Ex : UVa10408)
- Systèmes de numération,
  (Ex : UVa443)
- Utilisation maligne de `sqrt`, `log` ou `exp`,
  (Ex : UVa701, UVa10916, UVa11847)
- Polynômes (produits, dérivations, évaluation).
  (Ex : UVa498, UVa10268, UVa10586)

Contenu

Adopter le bon type

Problèmes en apparence facile, mais calculs difficiles :

- Entiers de grosse taille :
  - → `BigInteger` en Java.
  - → Gérer à la main en C++.
- Grande précision demandée :
  - → Recoder une classe pour les rationnels.

Exponentiation rapide

Algorithme de type diviser pour régner :

Exponentiation rapide

Soit $(M, \cdot)$ un monoïde, $g \in M$ et n un entier

$$g^n = g^{\lfloor n/2 \rfloor} \cdot g^{\lfloor n/2 \rfloor} \cdot g^{n\%2}.$$

Fonctionne avec des *entiers*, des *matrices*, des *permutations*, des *chaînes de caractères*, etc.

UVa306, UVa10625, UVa10710

PGCD, relation de Bezout

- Algorithme d'Euclide :

```
int gcd(int a, int b) {
  if (b == 0) return a;
  return gcd(b, a%b);
}
```

- Relation de Bezout : $ua + vb = d$ où d est le PGCD :

```
pair<int, pair<int, int> > bezout(int a, int b) {
  if (b == 0) return make_pair(a, make_pair(1, 0));
  pair<int, pair<int, int> > p = bezout(b, a%b);
  int d = p.first;
  int u = p.second.first, v = p.second.second;
  return make_pair(d, make_pair(v, u - (a/b) * v));
}
```

- Application : $u = a^{-1} \mod b$ (inversion modulaire).

(Ex : UVa10407, UVa10090, UVa10673)

Restes chinois

Soient $n_1, n_2, \ldots, n_k$ des entiers deux à deux premiers entre eux.
Si :

$$\begin{cases} x & \equiv & b_1 & \mod n_1 \\ x & \equiv & b_2 & \mod n_2 \\ & \vdots & & \\ x & \equiv & b_k & \mod n_k \end{cases},$$

alors, en notant $c_i = (N/n_i)^{-1} \mod n_i$, on a :

$$x \equiv b_1 \frac{c_1 N}{n_1} + b_2 \frac{c_2 N}{n_2} + \cdots + b_k \frac{c_k N}{n_k} \mod N,$$

où $N = n_1 n_2 \cdots n_k$.

Nombres premiers

- On génère la liste des premiers par le crible d'Ératosthène :

```
bitset<10000001> P;
vector<long long> premiers;
P.set(); // initialisation
P[0] = P[1] = 0;
for (long long i = 2; i < pmax; i++)
  if (P[i]) {
    for (long long j=i*i; j < pmax; j+=i)
      P[j]=0;
    premiers.push_back(i);
  }
```

- Factorisation d'un entier n : par divisions successives, chercher les diviseurs de n parmi les premiers $\leq \sqrt{n}$.

(Ex : UVa10140)

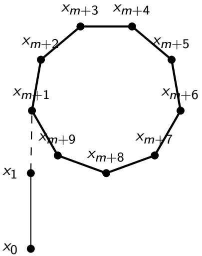
Suites récurrentes ultimement périodiques

Soit S un ensemble fini, $f : S \to S$ et $(x_n)_{n\in\mathbb{N}}$ une suite récurrente définie par $x_{n+1} = f(x_n)$.
Trouver λ et μ minimaux tel que $x_{\lambda+\mu} = x_\mu$.

Algorithme du lièvre et de la tortue

Poser $l \leftarrow f \circ f(x_0)$; $t \leftarrow f(x_0)$.
Tant que $(l \neq t)$, faire
  $l \leftarrow f \circ f(l)$; $t \leftarrow f(t)$.
$\mu \leftarrow 0$
$t \leftarrow x_0$
Tant que $(l \neq t)$, faire
  $l \leftarrow f(l)$; $t \leftarrow f(t)$; $\mu \leftarrow \mu + 1$.
$\lambda \leftarrow 1$
Tant que $(l \neq t)$, faire
  $l \leftarrow f(l)$; $\lambda \leftarrow \lambda + 1$.

Renvoyer λ (période) et μ (pré-période)



(Ex : UVa350, UVa11053)

Contenu

Fibonacci, binomiaux

- Suite de Fibonacci (Ex : UVa763, UVa10334, UVa10518)

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

$$F_n = \frac{\phi^n + (-\phi)^{-n}}{\sqrt{5}}, \quad \text{où } \phi = (1+\sqrt{5})/2.$$

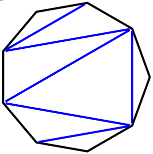
- Coefficients binomiaux (Ex : UVa10219, UVa10541)

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

Utiliser la formule directe pour un seul coefficient ou de la programmation dynamique pour beaucoup de coefficients.

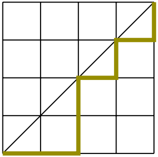
Nombres de Catalan

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \binom{2n}{n} - \binom{2n}{n+1} = \frac{2n(2n-1)}{(n+1)n} C_{n-1}.$$



Permet de compter :

- arbres binaires distincts à n sommets,
- mots de $\{(,)\}^*$ bien parenthésés,
- triangulations d'un polygone,
- chemins sous-diagonaux dans une grille carrée.



Récurrence :

$$C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}.$$

(Ex : UVa991, UVa10007, UVa10312)

Dérangements

Définition

Un dérangement est une permutation sans point fixe.

Théorème (OEIS A000166)

Le nombre D_n de dérangements de $\{1, \ldots, n\}$ satisfait

$$D_1 = 0, \; D_2 = 1, \quad D_{n+2} = (n+1)(D_{n+1} + D_n).$$

(Ex : UVa12024)

Arbres couvrants de graphes complets

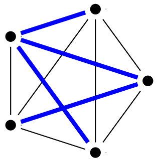
Théorème (formule de Cayley)

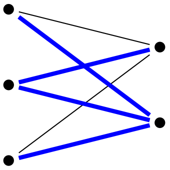
Il y a n^{n-2} arbres couvrants dans le graphe complet K_n.

Théorème

Il y a $n^{m-1}m^{n-1}$ arbres couvrants dans le graphe biparti complet $K_{m,n}$.

(Ex : UVa10843, UVa1179)





Formule d'Euler pour les graphes planaires

Théorème (Euler)

Un graphe planaire avec s sommets, a arêtes et f faces satisfait :

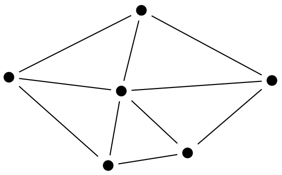
$$s - a + f = 2.$$

NB : La face extérieure est comptée.

(Ex : UVa10178)

Exemple : $s = 6$, $a = 10$, $f = 6$

$6 - 10 + 6 = 2$



Division du disque par des cordes

Théorème (Cercle de Moser, OEIS A000127)

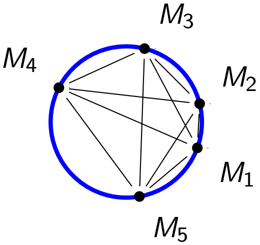
Un disque partagé par n cordes en position générique (i.e. 3 cordes ne sont pas concourantes) possède M_n parts avec :

$$M_n = \binom{n}{4} + \binom{n}{2} + 1.$$

(Ex : UVa10213)

Exemple : $n = 5$

$$M_5 = \binom{5}{4} + \binom{5}{2} + 1 \text{ parties}$$



Contenu

Problèmes de dénombrement

Approches :

- Si possible, aborder la question comme un problème de mathématiques et trouver une formule close,
- Sinon, chercher des relations de récurrence et utiliser de la programmation dynamique,
- Sinon, *force brute* (vérifier auparavant si la taille le permet).

Quelques outils

Principe d'inclusion-exclusion

Soient A et B deux ensembles, alors :

$|A \cup B| = |A| + |B| - |A \cap B|$

Généralise à de plus grandes unions ; aussi formule d'inversion de Möbius.

(Ex : UVa10882, UVa11806)

Formule de Burnside

Soit G un groupe agissant sur un ensemble Ω, alors :

$$|\text{Orbites}| = \frac{1}{|G|} \sum_{g \in G} |\text{Stab}_g(\Omega)| \,.$$

Coloration de collier ou de polyèdres réguliers à permutation près.

(Ex : UVa12387)

Problèmes de probabilités

Approches :

- Par du dénombrement :

$$\mathbb{P} = \frac{\sharp \text{ cas favorable}}{\sharp \text{ cas total}}.$$

- Probabilités conditionnelles parfois plus faciles à calculer.
- Processus aléatoire : calculer les paramètres puis simuler le processus sur un nombre fini suffisamment grand d'étapes.

Contenu

Stratégie Min-max

Jeux à deux joueurs, à somme nulle et à information complète.

- Arbre de décision :
  nœud = position du jeu,
  arête = coup qu'un joueur peut jouer.
- Score calculé récursivement des feuilles vers la racine :

$$\mathrm{score}(p) = \begin{cases} \max\{\mathrm{score}(f); f \text{ fils de } p\} & \text{si joueur joue,} \\ \min\{\mathrm{score}(f); f \text{ fils de } p\} & \text{si adversaire joue.} \end{cases}$$

Ex : UVa10368, UVa10111

Jeu de Nim

Des pièces sont entassées en k tas. Deux joueurs jouent à tour de rôle. À chaque tour, le joueur ou son adversaire doit retirer une ou plusieurs pièces choisies dans un même tas et perd sinon.
Solution : Soit n_i le nombre de pièces dans le tas i. Le (premier) joueur a une stratégie gagnante si

$$n_1 \oplus n_2 \cdots \oplus n_k \neq 0$$

où $\oplus$ est le XOR.

Ex : UVa11311