Dự tuyển olympic tin học

Duyệt toàn bộ

Nội dung

- Duyệt toàn bộ là gì?
- Hàm đệ quy
- Phát sinh mọi dãy nhị phân
- Phát sinh mọi hoán vị
- Phát sinh các tập con

Duyệt toàn bộ là gì?

- Duyệt toàn bộ = Dùng sức mạnh của máy tính để xem xét mọi trường hợp có thể xảy ra
- Trong thực tế đây chính là tận dụng lợi thế của máy tính so với con người (làm việc đơn giản, không mệt mỏi, xét nhiều tình huống)
  - Chẳng hạn: tính tổng các số 1+2+3+…+2017
  - Máy tính:

```
for (i = 1, k = 0; i <= 2017; i++) k+=i;
```

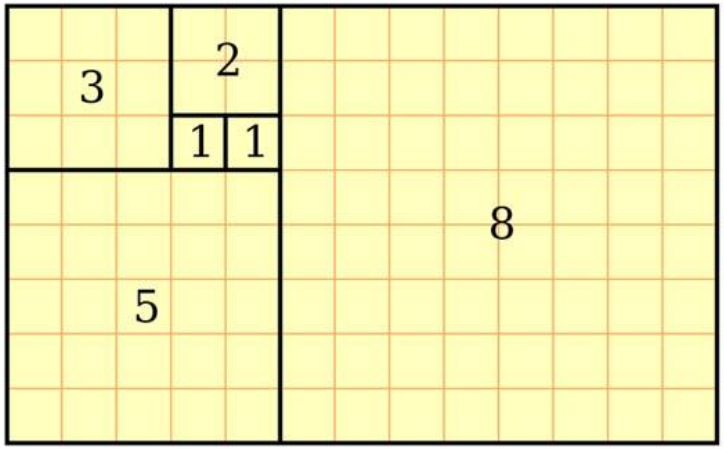
  - Con người:

```
k = 2017*(2017+1)/2;
```

- Thực tế cuộc sống số lượng bài toán có cách giải "tắt" (không duyệt toàn bộ) là tương đối ít

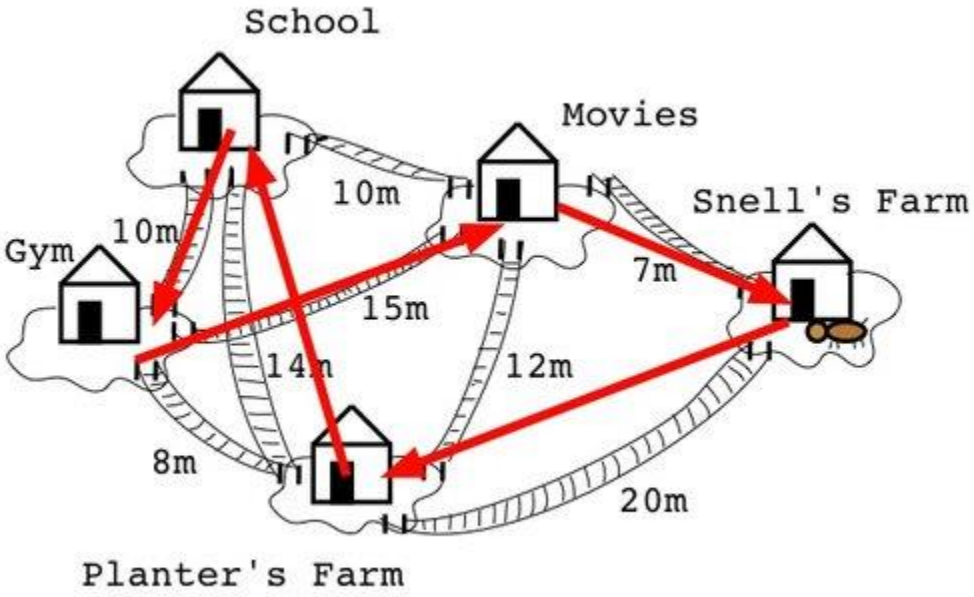
Duyệt toàn bộ là gì?

- Có những bài toán duyệt toàn bộ rất dễ:
  - Tìm bạn điểm cao nhất trong lớp
  - Tìm dòng ngắn nhất trong file
- Có những bài toán duyệt toàn bộ không dễ:
  - Phát sinh mọi cách đặt cặp ngoặc vào biểu thức
    - Biểu thức: a*b+c*d
    - Các phương án: a*b+c*d, a*(b+c)*d, (a*b+c)*d,…
  - Hãy chỉ ra mọi cách chia một hình chữ nhật kích thước m x n nguyên thành các hình chữ nhật con độ dài cạnh nguyên



Duyệt toàn bộ là gì?

- Có những bài toán ngoài cách duyệt toàn bộ hiện nay người ta chưa biết cách nào khác
  - Bài toán người du lịch: xuất phát từ khách sạn, tìm đường đi ngắn nhất qua các điểm thăm quan trong thành phố rồi trở về khách sạn ban đầu



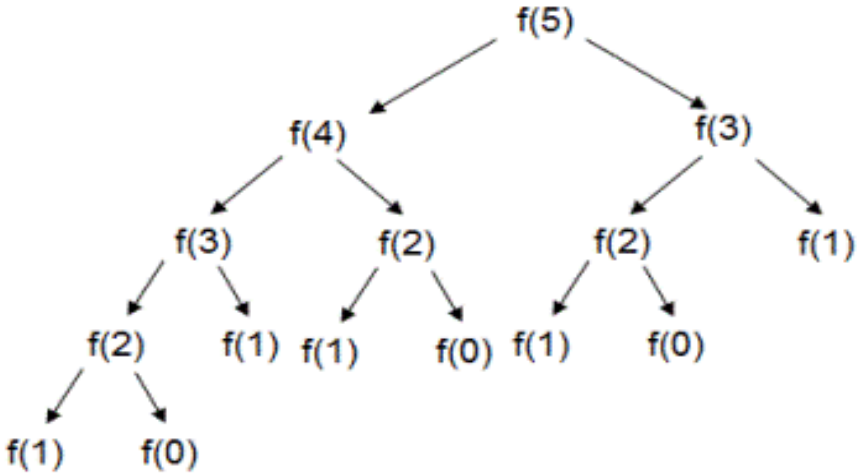
Hàm đệ quy

- Xuất phát từ ý tưởng chia bài toán lớn thành các bài toán con tương tự để giải
- Hàm đệ quy là một hàm trong quá trình thực thi có gọi chính lại nó
  - Gọi ngay chính nó: đệ quy trực tiếp
  - Gọi lòng vòng qua hàm khác: đệ quy gián tiếp
- Ví dụ về đệ quy trực tiếp: tính USCLN

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

Hàm đệ quy

- Tính số Fibonacci thứ n
  - f(0) = 0
  - f(1) = 1
  - f(n) = f(n-1) + f(n-2)



- Viết ở dạng đệ quy

```
int fibo(int n) {
    if (n < 2) return n;
    else return fibo(n-1) + fibo(n-2);
}
```

- Không phải đệ quy lúc nào cũng hay

Hàm đệ quy

- Hàm đệ quy có thể xem như lời giải của việc thực hiện rất nhiều vòng lặp lồng nhau, đặc biệt là khi ta chưa biết số lượng vòng lặp là bao nhiêu

```
void process(int depth) {
    if (depth == 0) {
        // xử lý
        return;
    }
    for (int i = 0; i < n; i++)
        process(depth - 1);
}
```

Phát sinh mọi dãy nhị phân

- Nhập 1 số n nguyên, hãy phát sinh mọi dãy nhị phân có độ dài n
  - N = 2: 00, 01, 10, 11
  - N = 3: 000, 001, 010, 011, 100, 101, 110, 111
  - …
- Ý tưởng:
  - Số thứ nhất có thể nhận giá trị 0 hoặc 1
  - Chọn số thứ n = 0, ta sinh mọi dãy con có độ dài n-1 và ghép với số thứ n là xong
  - Chọn số thứ n = 1, ta sinh mọi dãy con có độ dài n-1 và ghép với số thứ n là xong

Phát sinh mọi dãy nhị phân

- Chương trình:

```
void sinh(int n) {
  if (n == 0) { /* sinh xong, in ra */ }
  else {
    a[n] = 0; sinh(n-1);
    a[n] = 1; sinh(n-1);
  }
}
```

- Có thể dùng ý tưởng tương tự để sinh mọi dãy tam phân, tứ phân,…

Phát sinh mọi dãy nhị phân

- Việc sinh dãy nhị phân, tam phân,… có vẻ là một ý tưởng ngớ ngẩn, sẽ dùng vào việc gì?
- Bản chất là việc xét duyệt mọi tập con!
  - Tập cha: {0,1,2}
  - Các tập con: Ø, {0}, {1}, {2}, {0,1}, {0,2}, {1,2} và {0,1,2}
- Một lớp có n học sinh, được chia thành 2 dãy, hãy chỉ ra mọi cách chia có thể.
  - Những người thuộc dãy thứ nhất đánh số 0.
  - Những người thuộc dãy thứ hai đánh số 1.
  - Như vậy việc phân dãy thực chất là đánh số 01 cho tất cả n học sinh trong lớp.
  - Chỉ ra mọi cách chia dãy ~ chỉ ra mọi cách đánh số có thể

Phát sinh mọi dãy nhị phân

- Từ ý tưởng duyệt mọi dãy nhị phân hay phát sinh mọi tập con, ta có thể có phương pháp không đệ quy đơn giản sau đây để xuyệt toàn bộ

```
for (int a = 0; a < (1 << n); a++) {
    // xử lý theo a
}
// phiên bản đầy đủ hơn
for (int a = 0; a < (1 << n); a++) {
    vector<int> subset;
    for (int i = 0; i < n; i++) {
        if (a & (1 << i)) subset.push_back(i);
    }
}
```

Phát sinh mọi dãy nhị phân

1. Có n tờ tiền có mệnh giá a_0, a_1, ..., a_{n-1}. Hãy chỉ ra mọi cách chọn từ n tờ tiền đó một số tờ để tổng số tiền là X
   - Mã hóa: được chọn là 1, không được chọn là 0
   - Phát sinh mọi dãy nhị phân độ dài n, với mỗi cách phát sinh ta tính xem tổng số tiền được chọn có là X hay không? Nếu đúng bằng X thì in ra cách đó.
   - Chú ý: lời giải này chưa hay lắm
2. Cho xâu 123456789, hãy in ra mọi cách đặt các dấu + hoặc - vào giữa xâu để được biểu thức có giá trị đúng bằng 17.

Phát sinh mọi dãy nhị phân

3. Một bàn ăn có n người ngồi quanh, người ta dọn cho mỗi người một món ăn, thực đơn chỉ có 3 món A, B và C. Hãy chỉ ra mọi cách dọn đồ ăn thỏa mãn điều kiện: không có 3 người nào liên tiếp có đồ ăn giống nhau.
4. Một người lái xe thồ cần chở n món hàng nặng a_1, a_2,...a_n kilogram. Để dễ dàng di chuyển, người đó phải chia n món hàng sang 2 phía của xe thồ sao cho chênh lệch cân nặng giữa 2 bên càng nhỏ càng tốt. Hãy chỉ ra phương án tốt nhất.

Duyệt nhanh hơn?

- Muốn tăng tốc độ của các bài toán duyệt, có 2 chiến lược cơ bản
  - Không duyệt thừa (tính cái không cần thiết)
  - Không duyệt lại (tính cái đã tính từ trước)
- Không duyệt lại = dùng bộ nhớ để lưu trữ các kết quả đã tính = đệ quy có nhớ
- Không duyệt thừa: rất nhiều chiến lược
- Các chiến lược dựa trên quay lui:
  - Cắt nhánh: không đi những nhánh mà ta biết chắc là không chứa nghiệm
  - Nhánh cận: không đi những nhánh mà ta biết xác xuất có nghiệm khá thấp (hoặc bằng 0)