

Thuật toán ứng dụng

Bài thực hành số 1.1: Nhập môn

TS. Đinh Viết Sang, TA. Đặng Xuân Vương



Trường Đại học Bách khoa Hà Nội
Viện Công nghệ thông tin và Truyền thông

Ngày 20 tháng 12 năm 2020

Mục lục

1 ADD

2 SUBSEQMAX

Mục lục

1 ADD

2 SUBSEQMAX

01. ADD

- Cho hai số a và b , hãy viết chương trình bằng C/C++ tính số $c = a + b$
- Lưu ý giới hạn: $a, b < 10^{19}$ dẫn đến c có thể vượt quá khai báo `long long`

- Chỉ cần khai báo a, b, c kiểu `unsigned long long`, trường hợp tràn số chỉ xảy ra khi a, b có 19 chữ số và c có 20 chữ số
- 1 Tách $a = a1 \times 10 + a0$
- 2 Tách $b = b1 \times 10 + b0$
- 3 Tách $a0 + b0 = c1 \times 10 + c0$
- 4 In ra liên tiếp $a1 + b1 + c1$ và $c0$

Code (chỉ cần đoạn code chính thể hiện thuật toán)

```
int main() {  
    unsigned long long a,b,c;  
    cin >> a>>b;  
  
    unsigned long long a0 = a % 10;  
    unsigned long long a1 = (a-a0) / 10;  
    unsigned long long b0 = b % 10;  
    unsigned long long b1 = (b-b0) /10;  
    unsigned long long c0 = (a0+b0) % 10;  
    unsigned long long c1 = (a0+b0-c0) / 10;  
    c1 = a1 + b1 + c1;  
    if (c1>0) cout << c1;  
    cout << c0;  
    return 0;  
}
```

Mục lục

1 ADD

2 SUBSEQMAX

01. SUBSEQMAX

- Cho dãy số $s = \langle a_1, \dots, a_n \rangle$
- một dãy con từ i đến j là $s(i, j) = \langle a_i, \dots, a_j \rangle$, $1 \leq i \leq j \leq n$
- với trọng số $w(s(i, j)) = \sum_{k=i}^j a_k$
- Yêu cầu: tìm dãy con có trọng số lớn nhất

Ví dụ

- dãy số: -2, 11, -4, 13, -5, 2
- Dãy con có trọng số cực đại là 11, -4, 13 có trọng số 20

Có bao nhiêu dãy con?

- Số lượng cặp (i, j) với $1 \leq i \leq j \leq n$
- $\binom{n}{2} + n$
- Thuật toán trực tiếp!

Thuật toán trực tiếp — $\mathcal{O}(n^3)$

- Duyệt qua tất cả $\binom{n}{2} + n = \frac{n^2+n}{2}$ dãy con

```
public long algo1(int[] a){
    int n = a.length;
    long max = a[0];
    for(int i = 0; i < n; i++){
        for(int j = i; j < n; j++){
            int s = 0;
            for(int k = i; k <= j; k++){
                s = s + a[k];
            }
            max = max < s ? s : max;
        }
    }
    return max;
}
```

Thuật toán tốt hơn — $\mathcal{O}(n^2)$

- Quan sát: $\sum_{k=i}^j a[k] = a[j] + \sum_{k=i}^{j-1} a[k]$

```
public long algo2(int[] a){
    int n = a.length;
    long max = a[0];
    for(int i = 0; i < n; i++){
        int s = 0;
        for(int j = i; j < n; j++){
            s = s + a[j];
            max = max < s ? s : max;
        }
    }
    return max;
}
```

Thuật toán Chia để trị

- Chia dãy thành 2 dãy con tại điểm giữa $s = s_1 :: s_2$
- Dãy con có trọng số cực đại có thể
 - nằm trong s_1 hoặc
 - nằm trong s_2 hoặc
 - bắt đầu tại một vị trí trong s_1 và kết thúc trong s_2
- Code Java:

```
private long maxSeq(int i, int j){
    if(i == j) return a[i];
    int m = (i+j)/2;
    long ml = maxSeq(i,m);
    long mr = maxSeq(m+1,j);
    long maxL = maxLeft(i,m);
    long maxR = maxRight(m+1,j);
    long maxLR = maxL + maxR;
    long max = ml > mr ? ml : mr;
    max = max > maxLR ? max : maxLR;
    return max;
}

public long algo3(int[] a){
    int n = a.length;
    return maxSeq(0,n-1);
}
```

Chia để trị — $\mathcal{O}(n \log n)$

```
private long maxLeft(int i, int j){
    long maxL = a[j];
    int s = 0;
    for(int k = j; k >= i; k--){
        s += a[k];
        maxL = maxL > s ? maxL : s;
    }
    return maxL;
}

private long maxRight(int i, int j){
    long maxR = a[i];
    int s = 0;
    for(int k = i; k <= j; k++){
        s += a[k];
        maxR = maxR > s ? maxR : s;
    }
    return maxR;
}
```

Thuật toán Quy hoạch động

- Thiết kế hàm tối ưu:
 - Đặt s_i là trọng số của dãy con có trọng số cực đại của dãy a_1, \dots, a_i mà kết thúc tại a_i
- Công thức Quy hoạch động:
 - $s_1 = a_1$
 - $s_i = \max\{s_{i-1} + a_i, a_i\}, \forall i = 2, \dots, n$
 - Đáp án là $\max\{s_1, \dots, s_n\}$
- Độ phức tạp thuật toán là n (thuật toán tốt nhất!)

Quy hoạch động — $\mathcal{O}(n)$

```
public long algo4(int[] a){
    int n = a.length;
    long max = a[0];
    int[] s = new int[n];
    s[0] = a[0];
    max = s[0];
    for(int i = 1; i < n; i++){
        if(s[i-1] > 0) s[i] = s[i-1] + a[i];
        else s[i] = a[i];
        max = max > s[i] ? max : s[i];
    }
    return max;
}
```

Thuật toán ứng dụng

Bài thực hành số 1.1: Nhập môn

TS. Đinh Viết Sang, TA. Đặng Xuân Vương



Trường Đại học Bách khoa Hà Nội
Viện Công nghệ thông tin và Truyền thông

Ngày 20 tháng 12 năm 2020