

# Thuật toán ứng dụng

## Bài thực hành số 1.1: Nhập môn

TS. Bùi Quốc Trung, TA. Đặng Xuân Vương



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

Ngày 8 tháng 3 năm 2021

# Mục lục

1 ALICEADD

2 SUBSEQMAX

# Mục lục

1 ALICEADD

2 SUBSEQMAX

# 01. ALICEADD

- 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  $a, b$  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  $c0$

```
int main() {
    unsigned long long a, b;
    cin >> a >> b;

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

# Mục lục

1 ALICEADD

2 SUBSEQMAX

# 01. SUBSEQMAX

- Cho dãy số  $s = \langle a_1, \dots, a_n \rangle$
- Dãy con liên tiếp  $s(i, j) = \langle a_i, \dots, a_j \rangle$ ,  $1 \leq i \leq j \leq n$
- Trọng số  $w(s(i, j)) = \sum_{k=i}^j a_k$
- Yêu cầu: tìm dãy con liên tiếp 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

```
long long algo1(int *a, int n){
    long 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:  $\sum_{k=i}^j a[k] = a[j] + \sum_{k=i}^{j-1} a[k]$

```
long long algo2(int *a, int n) {  
    long 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  $s_2$
- Các hàm cần xây dựng:
  - `maxSeq(int *a, int i, int j)` trả về dãy con lớn nhất của dãy  $s(i, j)$ .
  - `maxLeft(int *a, int i, int j)` trả về dãy con lớn nhất của dãy  $s(i, j)$  kết thúc tại  $j$ .
  - `maxRight(int *a, int i, int j)` trả về dãy con lớn nhất của dãy  $s(i, j)$  bắt đầu tại  $i$ .

```
long long algo3(int *a, int n) {  
    return maxSeq(a, 0, n-1);  
}
```

## Chia để trị — $\mathcal{O}(n \log n)$

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

# Chia để trị — $\mathcal{O}(n \log n)$

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

long long maxRight(int *a, int i, int j) {
    long long maxR = a[i];
    long long 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 — $\mathcal{O}(n)$

```
long long algo4(int *a, int n) {  
    long long *s = new long long[n];  
    s[0] = a[0];  
    long long 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];  
    }  
    delete s;  
    return max;  
}
```

# Thuật toán ứng dụng

## Bài thực hành số 1.1: Nhập môn

TS. Bùi Quốc Trung, TA. Đặng Xuân Vương



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

Ngày 8 tháng 3 năm 2021