

Bài 3



Kỹ thuật thiết kế thuật toán





- **Kỹ thuật tuần tự**
 - Khái niệm
 - Bài toán
- **Kỹ thuật tổ chức theo cấu trúc**
 - Khái niệm
 - Bài toán
- **Kỹ thuật đệ quy và lặp**
 - Khái niệm
 - Bài toán
- **Lựa chọn cấu trúc dữ liệu phù hợp**
 - Ý tưởng
 - Bài toán

Kỹ thuật tuần tự



- Giải bài toán bằng cách thực hiện tuần tự từng thao tác cho đến khi nhận được kết quả.



- Ví dụ 1: Tính $n!$
- *Input:* n ;
- *Output:* $T = n!$
- *Chi tiết:*
 1. $T = 1$;
 2. $i = 0$;
 3. $i = i + 1$;
 4. $T = T * i$;
 5. if $(i < n)$ goto 3;



- *Viết lại:*
 - $T = 1;$
 - $i = 0;$
 - while ($i < n$)
 - {
 - $i = i + 1;$
 - $T = T * i;$
 - }

Kỹ thuật tuần tự



Ví dụ 2: Tìm xâu con dài nhất có các kí tự liên tiếp khác nhau

Ví dụ: 1 5 5 2 3 7 8 5 5 6 7



- *Input* : Xâu S;
- *Output*: Xâu $X \subseteq S$, X là xâu con dài nhất gồm những kí tự liên tiếp khác nhau
- *Chi tiết*:
 1. $X = \emptyset$; $d = 0$;
 2. Bắt đầu từ kí tự $i = 1$;
 3. Tìm xâu con gồm các kí tự liên tiếp khác nhau
 1. $j = i+1$; while ($S[j] \neq S[j-1]$) $j = j + 1$;
 4. if ($j-i > d$) $X = S[i..j-1]$; $d = j-i$;
 5. if ($j < n$)
 6. $i = j$; goto 3;



- *Chi tiết viết lại:*

```
n = len(S);
```

```
i = 1; d = 0; X=[];
```

```
while (i<n)
```

```
{
```

```
    j = i + 1;
```

```
    while (j≤n) (S[j] ≠S[j-1]) j=j+1;
```

```
    if (d < j-i)
```

```
        {d = j -i+1; X=S[i..j-1];}
```

```
    i = j;
```

```
}
```


Kỹ thuật tổ chức theo cấu trúc



- Tổ chức giải quyết bài toán từ trên xuống: từ mức tổng quát là ý tưởng giải quyết bài toán cho đến mức chi tiết là các chỉ dẫn để giải quyết công việc đặt ra.

Kỹ thuật tổ chức theo cấu trúc



- Ví dụ 3: Cho a, b là các số nguyên có không quá 100 chữ số. Tính $c = a \times b$.
- Mức ý tưởng:
 - Kí hiệu các chữ số tính từ hàng đơn vị của a và b là $a_0 a_1 a_2 \dots a_n$ và $b_0 b_1 b_2 \dots b_m$.
 - Nhân lần lượt từng chữ số của b là b_0, b_1, \dots, b_m với a , chú ý tới vị trí của chữ số, cộng dồn kết quả vào c .

Kỹ thuật tổ chức theo cấu trúc



- Chi tiết thuật toán B1 mức 1:
- *Input:* 2 số nguyên lớn a, b;
- *Output:* $c = a \times b$;
- *Chi tiết:*

$c = \emptyset$;

for (i=0; i ≤ m; i++)

{

$tg = b_i \otimes 10^i \otimes a$;

$c = c \oplus tg$;

}

Kí hiệu \otimes , \oplus là các phép toán nhân và cộng các số nguyên lớn.

Kỹ thuật tổ chức theo cấu trúc



- Chi tiết thuật toán B1 mức 2:
- Tổ chức lưu trữ số nguyên lớn vào mảng kiểu `vlint`, xây dựng các thủ tục:
- **void nhân (`byte` x, `vlint` a, &`vlint` kq):**
- thực hiện nhân số nguyên a với chữ số x và lưu kết quả vào kq: $kq = x \otimes a$;
- **void cộng (`vlint` x, `vlint` y, &`vlint` kq):**
- thực hiện cộng 2 số nguyên lớn x và y, lưu kết quả vào kq: $kq = x \oplus y$;
- **void thêm_10(`vlint` x, `int` i):**
- thực hiện nhân số nguyên lớn x với 10^i , thực chất là thêm i chữ số 0 vào sau x.

Kỹ thuật tổ chức theo cấu trúc



- Kí hiệu $Ln = \max\{n, m\}$.
- *Input*: Các mảng $a[0..n]$, $b[0..m]$.
- *Output*: $c[0..Ln]$;
- *Chi tiết*:
- for ($i=0$; $i \leq Ln$; $i++$) $c[i] = 0$;
- for ($i=0$; $i \leq m$; $i++$)
- {
 - nhân (b_i , a , tg);
 - thêm_10 (tg , i);
 - cộng (tg , c , c);
- }



- Một cấu trúc có tính chất đệ quy (hoặc được gọi là đệ quy) nếu trong mô tả của nó có một (hoặc vài) cấu trúc con được xác định bằng một số các trường hợp đơn giản và quy tắc đưa các trường hợp phức tạp về các trường hợp đơn giản đã biết.
- Một thủ tục hoặc hàm có thể có lời gọi đến chính nó. Thủ tục hoặc hàm viết có yếu tố đó được gọi là đệ quy.
- Nếu lời giải bài toán P được thực hiện bằng cách giải bài toán P' có cấu trúc giống P nhưng kích thước nhỏ hơn được gọi là lời giải đệ quy. Giải thuật tương ứng với lời giải như vậy gọi là giải thuật đệ quy.



- Ví dụ 4: Tích $n!$
 - $n!=1$; $n=0$
 - $n!=n*(n-1)!$; $n>0$
- Theo định nghĩa này ta có thể mô tả hàm gt tính $n!$ như sau

```
int gt(int n)
{
    if (n == 0) return 1;
    else      return n* gt(n - 1);
}
```



- Ví dụ 5: Xét bài toán tìm ước số chung lớn nhất (USCLN) của hai số nguyên dương m và n . Nếu sử dụng công thức $USCLN(m, n) = USCLN(n, m \bmod n)$ và $USCLN(n, 0) = n$



```
int USCLN(int m, int n)
{
    if (n == 0) return m;
    else return USCLN(n, m%n);
}
```



- Ví dụ 6: Bài toán fibonacci
 - $F(n) = f(n-1) + f(n-2)$
 - $F(0) = f(1) = 1;$

Lựa chọn cấu trúc dữ liệu thích hợp



- Sử dụng cấu trúc dữ liệu phù hợp có thể làm cho thuật toán gọn gàng và dễ xem hơn.

Lựa chọn cấu trúc dữ liệu thích hợp



- Tên năm tính theo âm lịch gồm có 2 thành phần: can và chi. Chẳng hạn, năm 2013 là “Quý Tị” được cấu tạo từ can “Quý” và chi “Tị”. Các can bao gồm Canh, Tân, Nhâm, Quý, Giáp, Ất, Bính, Đinh, Mậu, Kỷ, và 12 chi, bao gồm Thân, Dậu, Tuất, Hợi, Tí, Sửu, Dần, Mão, Thìn, Tị, Ngọ, Mùi. Các can và chi kết hợp với nhau một cách tuần tự và lặp lại. Như vậy có thể thấy ngay là năm 2012 là Nhâm Thìn vì can Nhâm trước can Quý, chi Thìn trước chi Tị, còn 2014 sẽ là Giáp Ngọ. Xây dựng giải thuật đổi tên năm dương lịch sang âm lịch.
- Phân tích: Có thể thấy chu kỳ lặp lại của các tên năm theo âm lịch là bội số chung nhỏ nhất của 10 (số can) và 12 (số chi) là 60.

Lựa chọn cấu trúc dữ liệu thích hợp



- Phương án 1: thành lập một mảng `TEN_AMLICH[0..59]` gồm 60 cái tên [..., “Nhâm Thìn”, “Quý Tị”, “Giáp Sửu”,...] với “Canh Thân” đứng đầu danh sách vì có thể thấy ngay là 1980, 1920, ..., 60,0 đều là Canh Thân. Nếu kí hiệu n là năm dương lịch thì thuật toán tính tên âm lịch tương ứng chỉ là
- `TEN_AMLICH[$n \bmod 60$]`.

Lựa chọn cấu trúc dữ liệu thích hợp



- Phương án 2: thành lập 2 mảng CAN[0..9] gồm 10 cái tên can ["Canh", ..., "Nhâm", "Quý", "Giáp",...] và CHI[0..11] với 12 tên chi ["Thân", ... "Thìn", "Tị", "Sửu",...]. Nếu kí hiệu n là năm dương lịch thì thuật toán tính tên âm lịch tương ứng là
- $\text{Can}[n \bmod 10] + \text{Chi}[n \bmod 12]$.

Thảo luận về INPUT, OUTPUT



- Tầm quan trọng của xác định INPUT
- Tầm quan trọng của OUTPUT
- Ví dụ 1: Cho một dãy, và một giá trị x xác định sự xuất hiện của x trong dãy.
- Ví dụ 2: Giải phương trình bậc 2 có dạng $a*x^2+b*x+c=0$
 - Xác định INPUT
 - Xác định OUTPUT
- Ví dụ 3: Cho một dãy các phần tử, mỗi phần tử là một ký tự tiếng anh, hãy sắp xếp dãy tăng dần.

Các vấn đề

- Kỹ thuật tuần tự
 - Khái niệm
 - Bài toán
- Kỹ thuật tổ chức theo cấu trúc
 - Khái niệm
 - Bài toán
- Kỹ thuật đệ quy và lặp
 - Khái niệm
 - Bài toán
- Lựa chọn cấu trúc dữ liệu phù hợp
 - Ý tưởng
 - Bài toán

