Cấu trúc dữ liệu

Data Structure

Nguyễn Đình Hoàng Sơn
BM Hệ thống Thông Tin
sonndh@ntu.edu.vn

Tài liệu tham khảo

[1] Hồ Thuần – Hồ Cẩm Hà – Trần Thiên Thành
- CẤU TRÚC DỮ LIỆU, PHÂN TÍCH THUẬT TOÁN VÀ PT PHẦN MỀM

[2] Đỗ Xuân Lôi
- CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

[3] Lê Minh Hoàng
- GIẢI THUẬT & LẬP TRÌNH

[4] Đinh Mạnh Tường
- CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

[5] Đặng Bình Phương
- Slide bài giảng KỸ THUẬT LẬP TRÌNH

[6] A.V. Aho, J.E. Hopcroft, J.D. Ullman
- THE DESIGN & ANALYSIS OF COMPUTER ALGORITHMS

[7] N.E. Wirth
- DATA STRUCTURE & ALGORITHM

Chương I

THIẾT KẾ &
PHÂN TÍCH GIẢI THUẬT

Don’t start coding

You must design a working algorithm first.

1. THUẬT TOÁN (Algorithm)

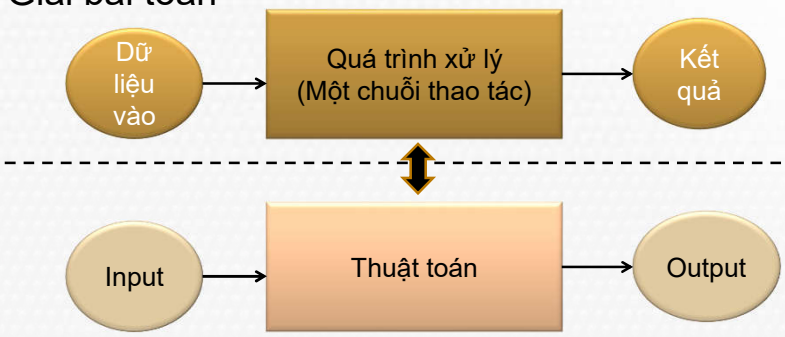
Al' Khwarizmi

D.E Knuth

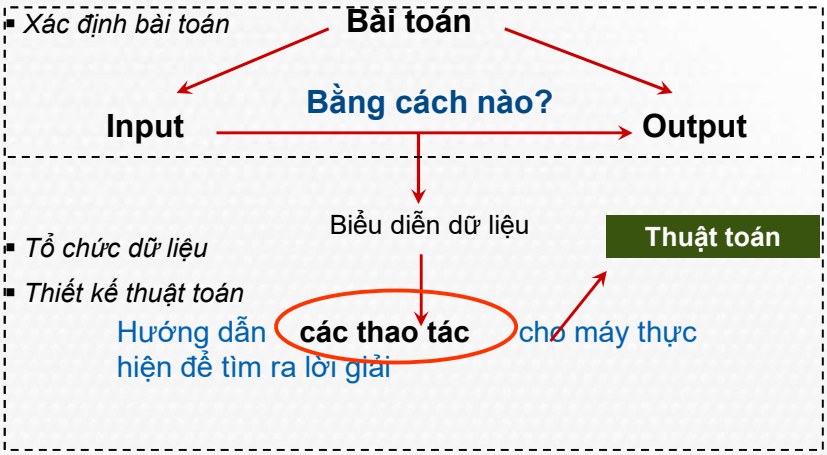
N.E Wirth

1. THUẬT TOÁN (Algorithm)

- Giải bài toán



1. THUẬT TOÁN (Algorithm)



1. THUẬT TOÁN (Algorithm)

- *Tổ chức dữ liệu:*
  - *Mô hình dữ liệu: (Data Model) trừu tượng hóa dữ liệu: Đồ thị, tập hợp, danh sách, cây, …*
    - *Phản ánh đúng thực tế + thao tác phù hợp*
    - *Tiết kiệm tài nguyên hệ thống*
  - *Cấu trúc dữ liệu (Data Structure): chọn các đơn vị cấu trúc (construct) của NNLT để biểu diễn mô hình dữ liệu*
    - *Kiểu dữ liệu: Data Type = (V, O)*
    - *Ví dụ: Mảng, bản ghi, chuỗi, file, ..*

1. THUẬT TOÁN (Algorithm)

- *Thuật toán để giải một bài toán là một dãy hữu hạn các thao tác được sắp xếp theo một trật tự xác định sao cho sau khi thực hiện dãy thao tác đó, từ Input của bài toán ta nhận được Output cần tìm.*

  – *Có nhiều thuật toán để giải cho cùng một bài toán*

  – *Cần xác định thuật toán thích hợp*

1. THUẬT TOÁN (Algorithm)

- Đặc trưng của thuật toán
  – *Tính kết thúc*: Sau hữu hạn bước cho kết quả
  – *Tính xác định*: Các bước riêng lẻ, cùng dữ liệu vào cho cùng kết quả ra
  – *Dữ liệu vào/ra*
  – *Tính phổ dụng*: Giải quyết cho 1 lớp bài toán
  – *Tính hiệu quả*: Nhanh chóng, chính xác
    - Thời gian
    - Dung lượng lưu trữ

1. THUẬT TOÁN (Algorithm)

- Mô tả thuật toán
  - Liệt kê: Nêu tuần tự các bước cần tiến hành
  - Sơ đồ khối (flow chart): Biểu diễn các thao tác bằng các hình vẽ qui ước
  - Mã giả (Peusedo Code): dùng ngôn ngữ tự nhiên + từ khóa qui ước thể hiện các thao tác

1. THUẬT TOÁN (Algorithm)

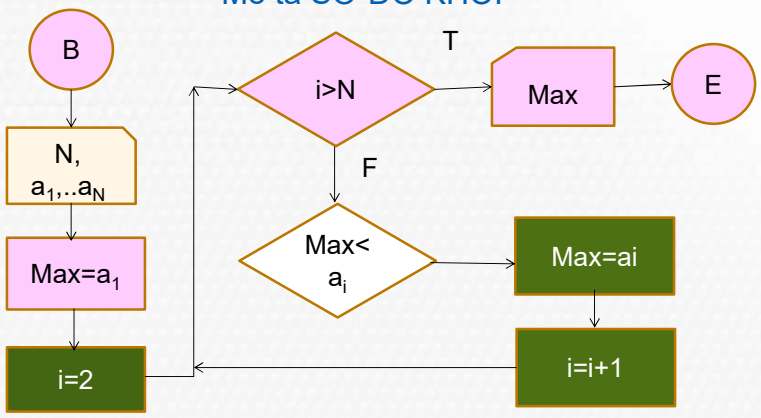
Ví dụ: Tìm giá trị lớn nhất của dãy số $a_1, a_2,\ldots,a_N$,
Input : Số nguyên dương N và dãy $a_1, a_2, ..., a_N$.
Output : Tìm Max là giá trị lớn nhất của dãy đã cho.

Mô tả LIỆT KÊ

Thuật toán:
B_1: Nhập N, dãy $a_1, a_2, ..., a_N$.
B_2: Đặt Max = a_1
B_3: i=2
B4: i>N $\Rightarrow B_7$
B5: Nếu Max<a_i thì Max=a_i
B6: i=i+1 $\Rightarrow B_4$
B7: Xuất Max $\Rightarrow$ Kết thúc

1. THUẬT TOÁN (Algorithm)

Mô tả SƠ ĐỒ KHỐI



1. THUẬT TOÁN (Algorithm)

Mô tả MÃ GIẢ

```
Read (N, a[1..N])
Set  Max = a[1];
i=2;
While  i <= N do {
      If Max<a[i] then Max=a[i] ;
     i=i+1;
}
Return Max.
```

1. THUẬT TOÁN (Algorithm)

- Tiêu chí đánh giá thuật toán
  - Tiêu chuẩn 1: Đơn giản, dễ hiểu, dễ cài đặt
  - Tiêu chuẩn 2:
    - Thời gian thực hiện nhanh
    - Dung lượng lưu trữ bé

  Nhận xét:

  - Tiêu chuẩn 1 mang tính chủ quan
  - Tiêu chuẩn 2 mang tính khách quan

2. ĐỘ PHỨC TẠP THUẬT TOÁN

- Để có thể so sánh thời gian chạy của các thuật toán:
  - Không phụ thuộc vào chủ quan của con người
  - Không phụ thuộc vào thiết bị

⇒Xây dựng một hàm đánh giá phụ thuộc vào kích thước của dữ liệu gọi là Độ phức tạp thuật toán.

⇒Hàm có dạng T(<kích thước bài toán>)

2. ĐỘ PHỨC TẠP THUẬT TOÁN

- Xây dựng hàm đánh giá độ phức tạp:
  - ❖1. Đếm số lượng các phép toán cơ bản
    - ➢ Phép toán cơ bản: +, -, *, /, phép gán, phép so sánh, thao tác đọc/ghi file …

Ký hiệu: D_n: tập các dữ liệu vào kích thước n

$cost_A(d)$: số phép toán cơ bản thuật toán A xử lý dữ liệu d

Độ phức tạp trong trường hợp xấu nhất: $T(n) = \underset{d \in D_n}{Max} \{cost(d)\}$

Độ phức tạp trong trường hợp tốt nhất: $T(n) = \underset{d \in D_n}{Min} \{cost(d)\}$

2. ĐỘ PHỨC TẠP THUẬT TOÁN

Ví dụ: Tính tổng S(n) = 1!+2!+..+n!

```
int P1(int n)
{
  int t, gt, i;
  t = 0;
  gt = 1;
  for(i= 1; i<= n; i++) {
      gt *= i;
      t  += gt;
  }
  return t;
}
```

Cách 1

Số lần	Chi phí
1	c_1
1	c_2
n	c_3
n	c_4
n	c_5

Độ phức tạp: $(c_3+c_4+c_5)n + c_1+c_2 \rightarrow T(n)$

Page ▪ 19

```
int P2(int n)
{
  int t, gt, i, j;
  t = 0;
  for(i= 1; i<= n; i++) {
      gt = 1;
      for(j= 1; j<= i; j++)
            gt *= j;
      t += gt;
  }
  return t;
}
```

Cách thứ 2:

Số lần	Chi phí	
1	c_1	
n	c_2	
n	c_3	
$\sum_1^n i$	c_4	$n(n+1)/2$
$\sum_1^n i$	c_5	$n(n+1)/2$
n	c_6	

Độ phức tạp:

$(c_4+c_5)n^2/2 + (c_2+c_3+c_6+(c_4+c_5)/2)n + c_1 \rightarrow T(n)$

Page ▪ 20

2. ĐỘ PHỨC TẠP THUẬT TOÁN

Ví dụ trường hợp tốt nhất, xấu nhất

```
int TimKiem(int a[100], int n, int x)
{
  int i;
  for(i= 0; i< n; i++)
      if (a[i] == x)
            return i;
  return -1;
}
```

Số lần

t = ?

1

- Nếu x ở đầu mảng: t = 1 (best case)
- Nếu x ở cuối mảng: t = n (worst case)

2. ĐỘ PHỨC TẠP THUẬT TOÁN

- Xây dựng hàm đánh giá độ phức tạp:
  - 2. Dùng ký hiệu O() (big oh)

Việc đánh giá thời gian thực hiện thông qua số lượng phép toán cơ bản quá chi tiết ⇒ khó khăn trong việc so sánh và phân lớp các thuật toán

Để thể hiện bản chất hơn độ phức tạp của thuật toán phụ thuộc vào kích thước ⇒ Khái niệm O-lớn

2. ĐỘ PHỨC TẠP THUẬT TOÁN

- **Khái niệm O()**

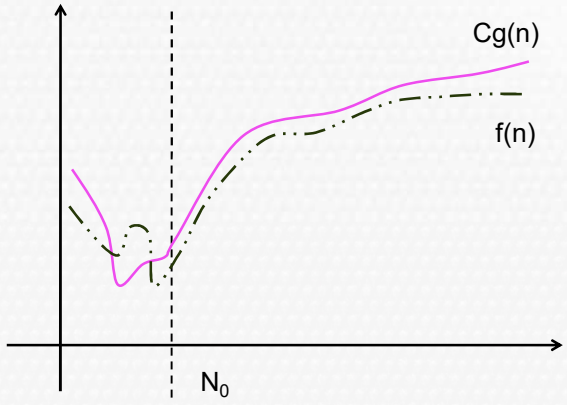
Cho f(n), g(n) là hai hàm xác định trên tập số nguyên không âm (tập số tự nhiên $\mathbb{N}$) có miền giá trị là R^+ tập số thực không âm:

$$f, g: \mathbb{N} \rightarrow R^+$$

Ta nói f(n) là O-lớn g(n), ký hiệu f(n)=O(g(n)) nếu và chỉ nếu:

$$\exists C, \exists N_0, \forall n \geq N_0 : f(n) \leq C.g(n)$$

2. ĐỘ PHỨC TẠP THUẬT TOÁN



Ký hiệu O-lớn cho ta phân lớp các hàm

2. ĐỘ PHỨC TẠP THUẬT TOÁN

Ký hiệu O-lớn	Tên gọi thường dùng
O(1)	Hằng
O(logn)	Logarit
O(n)	Tuyến tính
O(nlogn)	nlogn
$O(n^2)$	Bình phương
$O(2^n)$	Mũ

2. ĐỘ PHỨC TẠP THUẬT TOÁN

- **Các tính chất của O()**

1. Tính phản xạ:

   $f(n) = O(f(n))$

2. Tính bắt cầu:

   $f(n)=O(g(n)), g(n)=O(t(n)) \Rightarrow f(n)=O(t(n))$

3. Qui tắc hằng số:

   $f(n) = O(C.g(n)) \Rightarrow f(n) = O(g(n))$

4. Qui tắc cộng:

   $f_i(n)=O(g_i(n)), i=1,2 \Rightarrow f_1(n)+f_2(n) = O(\max(g_1(n),g_2(n)))$

2. ĐỘ PHỨC TẠP THUẬT TOÁN

5. Qui tắc nhân:

$f_i(n)=O(g_i(n))$, i=1,2

$\Rightarrow f_1(n) \times f_2(n) = O((g_1(n) \times g_2(n))$

▪Ứng dụng

- Câu lệnh đơn thực hiện một thao tác $\Rightarrow$ QT hằng số
- Câu lệnh hợp thành là dãy các câu lệnh $\Rightarrow$ QT tổng
- Câu lệnh rẽ nhánh dạng If ..then..else.. $\Rightarrow$ QT Max
- Các câu lệnh lặp $\Rightarrow$ QT Nhân

2. ĐỘ PHỨC TẠP THUẬT TOÁN

- Dãy tuần tự các lệnh đơn giản

  S_1; S_2; ..S_k, k: hằng số Độ phức tạp O(1)

- Vòng lặp đơn giản

```
for(i=1; i<= n;i++) {s}
```

s : câu lệnh đơn giản, độ phức tạp O(1)

Độ phức tạp n.O(1) hay O(n)

- Vòng lặp lồng nhau

```
for(i=1;i<n; i++)
    for(j=0;j<n;j++) {s}
```

Độ phức tạp n.O(n) hay $O(n^2)$

2. ĐỘ PHỨC TẠP THUẬT TOÁN

- Chỉ mục lặp không tuyến tính

```
h=1;
while (h<=n) {
    s;
    h=2*h;
}
```

h nhận các giá trị 1, 2, 4, ,, cho đến khi vượt n

Có $1+\log_2 n$ lần lặp: Độ phức tạp $O(\log_2 n)$

2. ĐỘ PHỨC TẠP THUẬT TOÁN

- Chỉ mục lặp phụ thuộc vào chỉ mục lặp khác

```
for(i=0; i<n; i++)
    for(j=0 ; j<i; j++) {s}
```

$$\sum_{i=0}^{n}\sum_{k=0}^{i} k$$

Độ phức tạp $\cong O(n^2)$

BÀI TẬP

1. XÁC ĐỊNH ĐỘ PHỨC TẠP CÁC ĐOẠN CHƯƠNG TRÌNH

a)
```
tong = 0;
for(i=0; i<n; i++)
  for(j=i; j<=i+3; j++)
      tong += j;
```

b)
```
tich = 0;
for(i=n; i>0; i--)
  for(j= 0; j<2*i; j++)
      tich *= j;
```

c)
```
tong= 0;
for(i=n; i>=1; i--)
  for(j=i; j>=1; j--)
      for(k=i; k<=i+3; k++)
            tong+= i+ j+ k;
```

d)
```
tong= 0;
for(i=1; i<=n; i++)
  for(j=1; j<=i; j++)
      for(k=1; k<=2*i; k++)
            tong+= i+ j+ k;
```

e)

```
tong= 0;
for(i=1; i<=n; i++)
  for(j=1; j<=i; j++)
     for(k=1; k<=j; k++)
           tong+= i+ j+ k;
```

f)

```
for(i=0; i<n-1; i++)
  for(j=i+1; j<n; j++)
    if (a[j]< a[i])
    {
        int t = a[i];
        a[i]= a[j];
        a[j]= t;
    }
```


Gợi ý:

a) $\sum_{i=0}^{n-1} 4$

b) $\sum_{i=0}^{n-1} (2*i)$

c) $\sum_{i=1}^{n} \sum_{j=1}^{i} 4$

d) $\sum_{i=1}^{n} \sum_{j=1}^{i} (2*i)$

e) $\sum_{i=1}^{n} \sum_{j=1}^{i} j$

f) $\sum_{i=0}^{n-1} n-i-1$

2. XÁC ĐỊNH ĐỘ PHỨC TẠP ĐOẠN CHƯƠNG TRÌNH

```
void NhiPhan(int n)
{
  int c, a[100];
  c = 0;
  while (n>0)
  {
      c++;
      a[c]= n%2;
      n= n/2;
  }
  printf("Ket qua la: ");
  for(i=c; i>=1; i--)
      printf("%d", a[i]);
}
```