CS112 – Phân tích độ phức tạp thuật toán không đệ quy

Nhóm 3:

- Ngô Đức Học
- Võ Duy Khang
- Nguyễn Tấn Huy

1. Cho thuật toán sau:

**ALGORITHM** *Mystery*$(n)$
    //Input: A nonnegative integer $n$
    $S \leftarrow 0$
    **for** $i \leftarrow 1$ **to** $n$ **do**
        $S \leftarrow S + i * i$
    **return** $S$

a) Output của thuật toán này là gì?

   Output: $\sum_1^n i^2$ (Tổng bình phương của các số từ 1 đến n)

b) Basic operation của thuật toán này là gì?

   Basic operation: Phép tính bình phương của biến i

c) Tính số lần thực thi basic operation? (Tính C(n))

   C(n) = n

d) Lớp hiệu năng của thuật toán?

   O(n)

e) Cải thiện hoặc đề xuất một thuật toán tốt hơn và xác định lớp hiệu năng? (Nếu có)

- Ta có thể cải thiện thuật toán này để giảm lớp hiệu năng xuống còn O(1)
- Ta có thể sử dụng công thức tính tổng bình phương các số từ 1 đến n đã được biết trước đó:

  $\sum_1^n i^2 = \frac{n(n+1)(2n+1)}{6}$
- Ta có thuật toán:

  *Mystery(n)*

  *S* ← 0

  For *i* ← *1* ***to*** *n* ***do***

  *S ← n * (n + 1) * (2 *n + 1) / 6*

  Return *S*

2. Cho thuật toán sau:

```
ALGORITHM   Secret(A[0..n − 1])
    //Input: An array A[0..n − 1] of n real numbers
    minval ← A[0]; maxval ← A[0]
    for i ← 1 to n − 1 do
        if A[i] < minval
            minval ← A[i]
        if A[i] > maxval
            maxval ← A[i]
    return maxval − minval
```

a) Giá trị chênh lệch giữa Min và Max trong mảng

b) Phép so sánh

c) C(n)=2(n-1)

d) 2(n-1)∈0(n)

e) Thuật toán đã tối ưu

3. **Cho thuật toán sau:**

```
ALGORITHM Enigma(A[0..n − 1, 0..n − 1])
    //Input: A matrix A[0..n − 1, 0..n − 1] of real numbers
    for i ← 0 to n − 2 do
        for j ← i + 1 to n − 1 do
            if A[i, j] ≠ A[j, i]
                return false
    return true
```

a) Output của bài toán là gì ?

True : nếu matrix A là matrix đối xứng

False: nếu matrix A khổng phải matrix đối xứng

b) Basic operation của thuật toán này là gì?

If A[i,j] # A[j,i]

c) Tính số lần thực thi basic operation? (Tính C(n))

C(n)= $\sum_{0}^{n-2}\sum_{i+1}^{n-1} 1$ =$\sum_{0}^{n-2}[(n-1)-(i+1)+1]$

= $\sum_{i=0}^{n-2}(n-1-i)$ =(n-1)+(n-2)+(n-3)+....1=(n-1)n/2 € o($n^2$)

d) Lớp hiệu năng của thuật toán: O($n^2$)

e) Không

4. **Cho thuật toán sau:**

**ALGORITHM** $GE(A[0..n-1, 0..n])$

//Input: An $n \times (n+1)$ matrix $A[0..n-1, 0..n]$ of real numbers
**for** $i \leftarrow 0$ **to** $n-2$ **do**
  **for** $j \leftarrow i+1$ **to** $n-1$ **do**
    **for** $k \leftarrow i$ **to** $n$ **do**
      $A[j,k] \leftarrow A[j,k] - A[i,k] * A[j,i] \,/\, A[i,i]$

**a)** Xác định lớp hiệu năng thời gian của thuật toán này?

C(n) $= \sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1}\sum_{k=i}^{n} 1 = \sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} n-i+1 = \sum_{i=0}^{n-2}(n-i+1)(n-i-1)$

$= \sum_{j=1}^{n-1} j(j+2) = \sum_{j=1}^{n-1} j^2 + \sum_{j=1}^{n-1} 2j = \frac{n(n-1)(2n-1)}{2} + n(n-1)$

$= \frac{n(n-1)(2n+5)}{6} \in \Theta(n^3)$.

**b)** Xác định sự kém hiệu quả trong thuật toán này và làm sao để tăng tốc độ của thuật toán?

Phép chia A[*j, i*] / A[*i, i*] nằm trong vòng lặp của biến *k* nhưng không phụ thuộc vào biến *k*. Để tối ưu thuật toán ta gán giá trị A[*j, i*] / A[*i, i*] bằng một biến temp ở ngoài vòng lặp k, đồng thời kiếm tra giá trị A[i,j] có khác 0 không rồi mới thực hiện tính và gán giá trị temp.