Nhóm 7
Phi Quang Đạt
Huỳnh Nhật Hoà
Phạm Quốc Anh Khoa

BÀI 1

Xét thuật toán sau:

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

a. Output của thuật toán này là gì?

Output là tổng S = $\sum_{i=1}^{n} i^2$

b. Basic operation của thuật toán này là gì?

Basic operator: S ← S + i*i

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

$C(n) = \sum_{i=1}^{n} 1 = n$

d. Lớp hiệu năng của thuật toán?

$C(n) \in \theta(n)$

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

Công thức cải thiện: $\sum_{i=1}^{n} i^2 = \frac{n.(n+1).(2n+1)}{6}$

$\Rightarrow C(n) \in \theta(1)$

BÀI 2

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

//Input: An array $A[0..n-1]$ of n real numbers

$minval \leftarrow A[0]$; $maxval \leftarrow A[0]$

for $i \leftarrow 1$ **to** $n-1$ **do**

  if $A[i] < minval$

    $minval \leftarrow A[i]$

  if $A[i] > maxval$

    $maxval \leftarrow A[i]$

return $maxval - minval$

a. **Output của thuật toán là gì ?**
   Out put của thuật toán là hiệu giữa 2 phần tử lớn nhất và nhỏ nhất của mảng n số thực
b. **Basic operation của thuật toán là gì ?**
   Basic operation của thuật toán là phép so sánh
c. **Số lần thực thi của basic operation là gì ?**
   Số lần thực thi của basic operation là 2(n-1) lần
d. **Lớp hiệu năng của thuật toán là gì ?**
   Lớp hiệu năng của thuật toán O(n)
e. **Cải thiện hoặc đề xuất một thuật toán tốt hơn:**
   không có

BÀI 3

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
```

1) **Output của thuật toán này là gì?**

Output của thuật toán trên sẽ là true hoặc false, tương ứng với việc ma trận input có đối xứng hay không. Nếu ma trận có đối xứng, output sẽ là true, ngược lại sẽ là false.

2) Basic operation của thuật toán này là gì?

Basic operation của thuật toán này là so sánh phần tử của ma trận, cụ thể là so sánh A[i,j] và A[j,i]

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

Ta sẽ lấy tổng số phần tử của ma trận trừ đi số phần tử nằm trên đường chéo chính

Số phần tử của ma trận là n^2, số phần tử nằm trên đường chéo chính là n

$C(n) = n^2 - n + n-1 + n-2 + ... + 2 + 1$

$= n^2 - n(n+1)/2$

$= \frac{n(n-1)}{2}$

4) Lớp hiệu năng của thuật toán?

$C(n) \in \Theta(n^2)$

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

Thuật toán trên kiểm tra tính đối xứng của ma trận

bằng cách so sánh các phần tử trên đường chéo chính với các phần tử đối xứng qua đường chéo chính. Tuy nhiên, ta có thể cải thiện thuật toán này bằng cách sử dụng thuật toán trực quan hơn là kiểm tra từng phần tử, và do đó giảm đáng kể thời gian thực thi của thuật toán.

Thuật toán mới sẽ so sánh từng phần tử của ma trận hiện tại với ma trận chuyển vị của nó. Việc chuyển vị của ma trận có thể thực hiện trong $O(n^2)$. Vì vậy lớp hiệu năng của thuật toán mới cũng là $O(n^2)$

Bài 4

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

```
//Input: An n × (n + 1) matrix A[0..n − 1, 0..n] of real numbers
for i ← 0 to n − 2 do
    for j ← i + 1 to n − 1 do
        for k ← i to n do
            A[j, k] ← A[j, k] − A[i, k] ∗ A[j, i] / A[i, i]
```

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

- Vòng lặp 1: (n - 1) lần -> n
- Vòng lặp 2: $\frac{n\,(n-1)}{2}$ lần -> n

- Vòng lặp 3: ($\frac{n\,(n-1)}{2}$ + n) lần -> n

Vì vậy hiệu năng của thuật toán này là $O(n^3)$

2) **Xác định sự kém hiệu quả trong thuật toán này và làm sao để tăng tốc độ của thuật toán**
   - Trường hợp A[i,i] = 0, thuật toán sẽ bị lỗi và phải thêm cách kiểm tra và xử lý trường hợp này
   - Kiểm tra coi tất cả các phần tử trong ma trận có bằng 0 hay không, nếu có thì thoát khỏi vòng lặp ngay lập tức
   - Thuật toán này sử dụng để giải hệ phương trình tuyến tính Ax = b bằng phương pháp Guass. Thay vào đó ta có thể sử dụng phương pháp Guass-Jordan, điều này sẽ giúp độ phức tạp của thuật toán từ $O(n^3)$ xuống $O(n^2)$