**Tên SV:** Thái Tấn Phát

**MSSV:** 2051050343

**BÀI TẬP CHƯƠNG I**

**Bài 1:** Đếm số phép toán gán, phép so sánh được thực thi và xác định độ phức tạp trong đoạn code sau:

for (i = 0; i < n; i++) 🡺 O(n)

for (j = 0; j < m; j++) 🡺O(n)

  if (a[ i ][ j ] = = x) return 1; 🡺O(1)

return -1;

🡺 T(n)= O(n2)

|  |  |
| --- | --- |
| i = 0=>1lần  i = 1=>4 lần  i = 2=>9 lần  i = 3=>16 lần  ………  i = n => n2 + 2n+1 lần  **Vậy tổng số phép gán** = n2 + 2n+1 | i = 0=>1 lần  i = 1=>5 lần  i = 2=>13 lần  ….    i = n =>2n2+ 2n + 1 lần  **Vậy tổng số phép so sánh** = 2n2+ 2n + 1 |

**Bài 2:** Đếm số phép toán gán, phép so sánh được thực thi và xác định độ phức tạp trong đoạn code sau:

sum = 0;  
for( i = 0; i < n ; i++)

2n+2 = O(n)

for( j = 0; j < i ; j++)

n\*(n-1)/2=O(n2)

sum++;

O(1)

🡺 T(n) = O(n2)

|  |  |
| --- | --- |
| i=0=>2 lần  i=1=>4 lần  i=2=>8 lần  i=3=> 14 lần  ………  i = n => n2+n+2 lần  **Số phép gán**=n2+n+2 | i=0 => 1 lần  i=1 => 3 lần  i=2 => 6 lần  i=3 => 10 lần  ………….  i = n => (n2+n)/2 + n+1 lần  **Số phép so sánh** = (n2+n)/2 + n+1 |

**Bài 3:** Đánh giá độ phức tạp của đoạn code sau:

2n+2 =O(n)

for (i = 0; i < n; i++)

sum1+=i;

O(1)

for (i = 0; i < n\*n; i++)

1+n2+1+n2=O(n2)

sum2+=i;

O(1)

🡺T(n) = O(n2)

**Bài 4:** Đánh giá độ phức tạp của hàm tính giai thừa sau:

int GT(int n)

{

if (n == 1)

return 1;

return n\*GT(n-1);

}

**Gọi** T(n) là thời gian tính n!

Thì T(n-1) là thời gian thực hiện (n-1)!

Khi n = 1 thì chương trình return 1 tốn O(1) => T(1) = 1

Khi n > 1 thì chương trình phải tính:

+ (n-1)!, tốn thời gian T(n-1)

+Tính n\*(n-1)! Và return về kết quả tốn hằng thời gian => 1.

1 nếu n = 1

=>T(n) =

T(n-1) +1 nếu n > 0

T(n) = T(n-1)+1

=[T(n-2) + 1]+1 = T(n-2)+2

=[T(n-3) + 1]+1 = T(n-3)+3

…

= T(n-k)+k

Qúa trình kết thức khi n – k = 1 => k = n

**Khi đó ta có T(n)** =T(1) + n = 1 + n = O(n)

**Bài 5:** Đánh giá độ phức tạp của hàm tính dãy FIBONACCI sau:

int GT(int n)

{

if (n == 1)

return 1;

return n\*GT(n-1);

}

**🡺 Khi đó ta có T(n)** **=** O(2^n)