***Bài 1:***

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

**for (j = 0; j < m; j++)**

**if (a[ i ][ j ] = = x)**

**return 1;**

**return -1;**

***giải:***

* Tính số phép gán cho for m:
* m = 0, 1 phép gán.
* m = 1, 2 phép gán.
* m = 2, 3 phép gán.
* m = ?, m + 1 phép gán.
* Tính số phép gán cho for n:
* n = 0, m + 1 (for m) \* 0 + 1 phép gán.
* n = 1, m + 1 (for m) \* 1 + 1 phép gán.
* n = 2, m + 1 (for m) \* 2 + 1 phép gán.
* n = ?, m + 1 (for m) \* n + 1 phép gán.
* T(m, n) = (m + 1) \* n + 1
* Nếu m = n ⬄ T(n) = (n + 1) \* n + 1 ≈ O(n2).

***Bài 2:***

**sum = 0;**

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

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

**sum++;**

***giải:***

Số phép gán cho for j:

i = 0 có 1 phép gán cho j; 0 phép cho sum.

i = 1 có 2 phép gán cho j; 1 phép cho sum.

i = 2 có 3 phép gán cho j; 2 phép cho sum.

Suy ra có i + 1 phép gán cho j và i phép gán cho sum => có 2i + 1 phép gán cho for j

Số phép gán cho for i:

n = 0 có (2i + 1) phép gán for j \* 0 + 1;

n = 1 có (2i + 1) phép gán for j \* 1 + 2;

n = 2 có (2i + 1) phép gán for j \* 2 + 3;

Suy ra T(i,n) = (2i + 1) \* n + n + 1 phép gán.

sum = 0 lúc đầu nên có thêm 1 phép gán nữa

=> T(i,n) = (2i + 1) \* n + n + 1 + 1

= 2\*i\*n + 2n + 2.

Mà i tối đa là n – 1.

=>T(n) = 2\*(n – 1)\*n + 2n +2

= 2n2 + 2. => O(n) = n2

\*số phép gán tối đa: 2n2 + 2 phép so sánh

\*số phép so sánh tối đa:

For i :

n = 0 thì có 1 phép so sánh vs i;

n = 1 thì có 2 phép so sánh vs i;

n = 3 thì có

=>có n phép so;

For j:

i = 0 thì có 2 phép so sánh vs i;

i = 1 thì có 3 phép so sánh vs i;

=> có i phép so sánh;

có n + i phép so sánh tổng quát

mà (i = n) phép so sánh nên có 2n phép so sánh tối đa.

***Bài 3:***

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

**sum1 += i;**

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

**sum2 += i;**

***giải:***

+ Số phép gán cho for i thứ 1

• n = 0 thì có 1 phép cho i;

• n = 1 thì có 2 phép gán cho i, 1 phép gán cho sum1;

• n = 2 thì có 3 phép gán cho i, 2 phép gán cho sum1;

* (n + 1)phép gán cho i và n phép gán cho sum1;
* For i thứ 1 có 2n +1 phép gán

+ Số phép gán cho for i

• n = 0 thì có 1 phép gán cho i, 0 phép gán cho sum2;

• n = 1 thì có 2 phép gán cho i, 1 phép gán cho sum2;

• n = 2 thì có 5 phép gán cho i, 2 phép gán cho sum2;

* Có n2 +1 phép gán cho i và n2 – 1 phép gán cho sum2
* For i thứ 2 có 2n2 phép gán
* T(n) = 2n2 +2n +1
* O(n) = n2

***Bài 4:***

**int GT(int n)**

**{**

**if (n == 1)**

**return 1;**

**return n \* GT(n - 1);**

**}**

***giải:***

T(n) = 1 (khi n = 1)

T(n) = C2 \* T(n -1) (khi n >1)

T(n) = GT(n -2) \*C2 \*C2 = T(n -2) \* (C2)2

T(n -3) \*C2 \*C2\*C2 = T(n -3) \* (C2)3

T(n – k)\*C2\*C2\*C2....\*C2 = T(n – k)\*(C2)k

Nhân k lần C2

Chương trình dừng khi n – k = 1 hay k = n – 1

T(n – k)\*(C2)k = T(n – n + 1)\* (C2)n - 1

T(1)\*(C2)n - 1 = (C2)n - 1

T(n) = (C2)n - 1

O(n) = (C2)2

***Bài 5:***

**{**

**if (n <= 1)**

**return n;**

**return Fibo(n - 1) + Fibo(n - 2);**

**}**

***giải:***

T(0)=T(1) =1

T(n) = C+ (T(n-1) + T(n – 2))

+ T(n -1) ≈ T(n – 2)

T(n) = 2T(n -2) + C

= 2\*(2T(n -4) + C) +C

= 4T(n -4) +3C k = 2

= 8T(n – 6) +7C k = 3

= 16T(n – 8) +15C k = 4

= 2kT(n – 2k) + (2k - 1)\*C

n – 2k = 0 → k = n/2

T(n) = 2n/2 T(0) + (2n/2 – 1)\*C.

= 2n/2+ (2n/2)\*C – C

= 2n/2 \*(C + 1) – C

T(n) ≈ 2n/2→ O(n) = 2n/2