CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (Data Structures & Algorithms)

L/O/G/O
GV: HUỲNH THỊ THANH THƯƠNG
Email: thuonghtt@uit.edu.vn

TỔNG QUAN VỀ GIẢI THUẬT VÀ CTDL

CHƯƠNG 1

L/O/G/O

Tổng hữu hạn

❖ Một số công thức cần nhớ:

Important Summation Formulas

1. $\sum_{i=l}^{u} 1 = \underbrace{1+1+\cdots+1}_{u-l+1 \text{ times}} = u-l+1$ (l, u are integer limits, $l \le u$); $\sum_{i=1}^{n} 1 = n$

2. $\sum_{i=1}^{n} i = 1+2+\cdots+n = \frac{n(n+1)}{2} \approx \frac{1}{2}n^2$

3. $\sum_{i=1}^{n} i^2 = 1^2+2^2+\cdots+n^2 = \frac{n(n+1)(2n+1)}{6} \approx \frac{1}{3}n^3$

4. $\sum_{i=1}^{n} i^k = 1^k+2^k+\cdots+n^k \approx \frac{1}{k+1}n^{k+1}$

Tổng hữu hạn

❖ Một số công thức cần nhớ:

5. $\sum_{i=0}^{n} a^i = 1 + a + \cdots + a^n = \frac{a^{n+1} - 1}{a - 1} \quad (a \neq 1); \qquad \sum_{i=0}^{n} 2^i = 2^{n+1} - 1$

6. $\sum_{i=1}^{n} i2^i = 1 \cdot 2 + 2 \cdot 2^2 + \cdots + n2^n = (n-1)2^{n+1} + 2$

7. $\sum_{i=1}^{n} \frac{1}{i} = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n + \gamma$, where $\gamma \approx 0.5772\ldots$ (Euler's constant)

8. $\sum_{i=1}^{n} \lg i \approx n \lg n$

Bài 1: Tính tổng hữu hạn

a. $1+3+5+7+\cdots+999$

b. $2+4+8+16+\cdots+1024$

c. $\sum_{i=3}^{n+1} 1$ **d.** $\sum_{i=3}^{n+1} i$ **e.** $\sum_{i=0}^{n-1} i(i+1)$

f. $\sum_{j=1}^{n} 3^{j+1}$ **g.** $\sum_{i=1}^{n}\sum_{j=1}^{n} ij$ **h.** $\sum_{i=1}^{n} 1/i(i+1)$

i. $\sum_{j\in\{2,3,5\}} (j^2+j)$ *j.* $\sum_{i=1}^{m}\sum_{j=0}^{n}\sum_{k=0}^{100}(i+j)$

Đếm số phép gán và so sánh, suy ra Độ phức tạp

Bài 2

```
s = 0;
i = 1;
while (i ≤ n) do
        j  = 1;
        while (j ≤ i²) do
                s = s + 1;
                j = j + 1;
        end do;
        i = i +1;
end do;
```

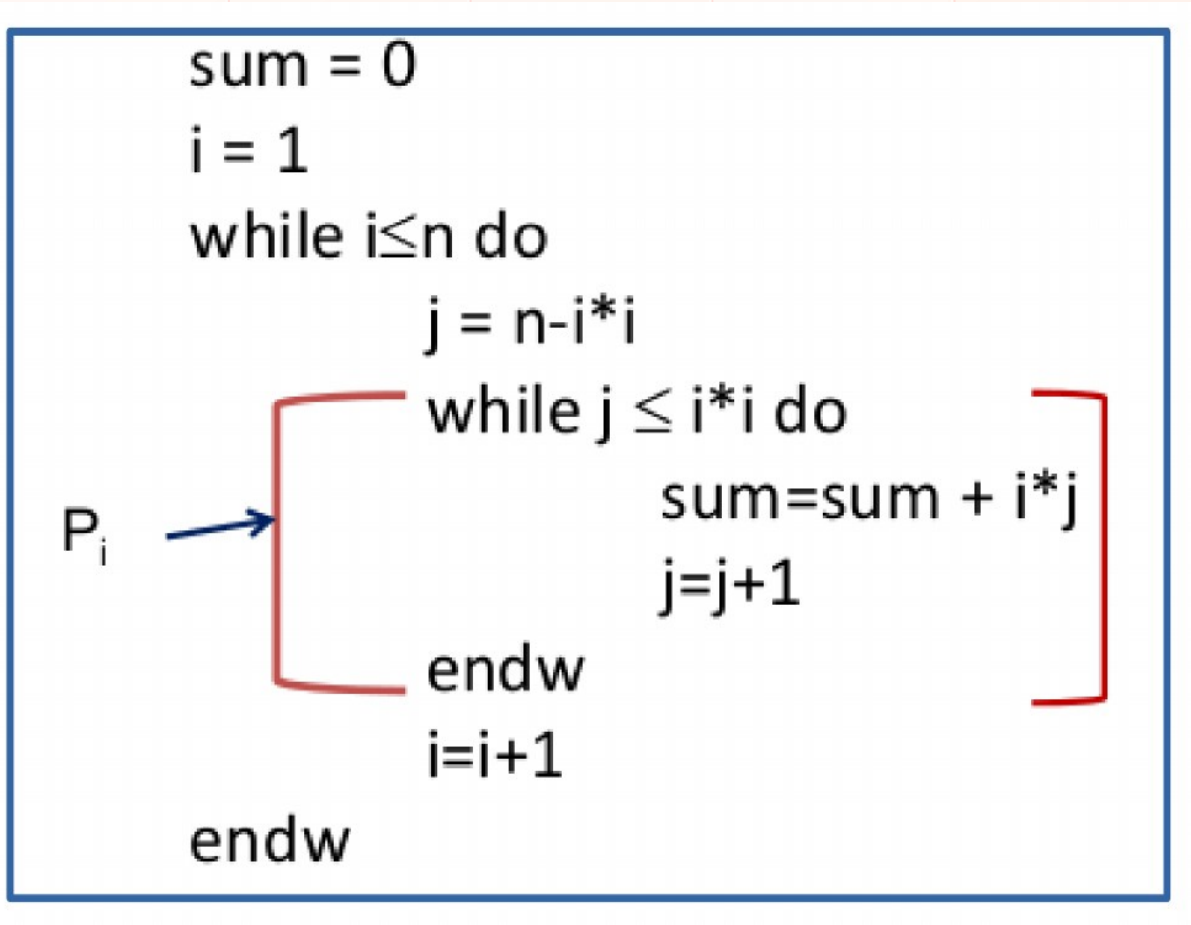
Đếm số phép gán và so sánh, suy ra Độ phức tạp

Bài 3

```
sum := 0;
i := 1;
while  (i ≤ n) do
        j := n-i ;
        while (j  ≤ i)  do
                sum := sum + j;
                j := j  + 1;
        endw;
        i = i + 1;
endw;
```

Đếm số phép gán và so sánh, suy ra Độ phức tạp

Bài 4



Đếm số phép gán và so sánh, suy ra Độ phức tạp

Bài 5

```
sum := 0;
i := 1;
while  (i ≤ n) do
        j := i ;
        while (j  > 0)  do
                sum := sum + 1;
                j := j  div 2;
        endw;
        i = i +1;
endw;
```

Đếm số phép gán và so sánh, suy ra Độ phức tạp

Bài 6

(Lưu ý: hiện tại cô chưa có danh sách nhóm nên "số thứ tự của nhóm" có thể chọn là 1 số bất kỳ trừ số 1)

```
i = 1; res = 0;
while (i ≤ n) do
        j = 1;
        while (j ≤ i) do
                res = res + i*j ;
                j = j +  số thứ tự của nhóm;
        end do;
        i = i + 1;
end do;
```

Bài 7: Ước lượng nhanh Θ (Lưu ý: có giải thích ngắn gọn)

- a)

```
int Fifth_Element(int A[],int n) {
    return A[5];
}
int Partial_Sum(int A[],int n) {
    int sum=0;
    for(int i=0;i<42;i++)
        sum=sum+A[i];
    return sum;
}
```

Bài 7: Ước lượng nhanh Θ (Lưu ý: có giải thích ngắn gọn)

- b)

```
void sum_first_n(int n) {
    int i,sum=0;
    for (i=1;i<=n;i++)
        sum = sum + i;
    }
void m_sum_first_n(int n) {
    int i,k,sum=0;
    for (i=1;i<=n;i++)
        for (k=1;k<7;k++)
            sum = sum + i;
    }
```

Bài 7: Ước lượng nhanh Θ (Lưu ý: có giải thích ngắn gọn)

- c) Ví dụ: chia dãy số làm 2 và chỉ xét nửa đầu hoặc nửa cuối

```
int binarysearch(int a[], int n, int val)
{
int l=1, r=n, m;
   while (r>=1) {
     m = (l+r)/2;
     if   (a[m]==val) return m;
     if   (a[m]>val) r=m-1;
     else l=m+1; }
return -1;
}
```

Bài 7: Ước lượng nhanh Θ (Lưu ý: có giải thích ngắn gọn)

- d)

```
int *compute_sums(int A[], int n) {
    int M[n][n];
    int i,j;
    for (i=0;i<n;i++)
        for (j=0;j<n;j++)
            M[i][j]=A[i]+A[j];
    return M;
    }
```

Bài 7: Ước lượng nhanh Θ (Lưu ý: có giải thích ngắn gọn)

- e)

```
/*1*/ s = 0;
/*2*/ i = 1;
/*3*/ while (i ≤ n) {
/*4*/       j  = n – i;
/*5*/       while (j ≥ 1) {
/*6*/           s = s + 1;
/*7*/           j = j – 1;
      }
/*8*/   i = i +1;
      }
```