

Phân tích & Thiết kế thuật toán (Algorithms Design & Analysis)

L/O/G/O

GV: HUỖNH THỊ THANH THƯỜNG

Email: thuonghtt@uit.edu.vn

TỔNG QUAN

CHƯƠNG 1



L/O/G/O

www.themegallery.com

Bài tập về nhà HW#01

MỘT SỐ LƯU Ý

- ❖ Bài tập đầu tiên này nhằm giúp SV “làm quen lại” với cách thức làm toán, trình bày và lập luận logic trong toán học.
- ❖ Trình bày bài làm như cách cô đã hướng dẫn trên lớp, phải có lập luận toán học chứ không phải chỉ đưa ra đáp án. Phải giải thích tại sao đếm ra được con số này...

HÌNH THỨC BÀI LÀM

- ❖ Cách 1: làm trực tiếp trên file word rồi chuyển sang pdf để công thức ko chạy lung tung
- ❖ Cách 2: làm trực tiếp trên powerpoint rồi chuyển sang pdf
- ❖ Cách 3: Làm bằng Latex (khuyến khích, sẽ có điểm bonus nếu dùng Latex, sau này ai làm Nghiên cứu khoa học thì phải trình bày report bằng Latex nên giờ làm quen luôn cũng có nhiều lợi ích)
- ❖ **NỘP ĐẦY ĐỦ file gốc và file pdf (VD: word và pdf, hay ppt và pdf, nếu dùng Latex thì cũng phải nộp file source của Latex và file pdf). Nếu thiếu thì GV sẽ không chấm bài**

HÌNH THỨC BÀI LÀM

- Trong bài làm có 1 trang bìa ghi các thông tin sau:

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH
BÀI TẬP MÔN PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN
HOMEWORK #01: ĐÁNH GIÁ THUẬT TOÁN DÙNG KỸ THUẬT TOÁN SƠ
CẤP

GV hướng dẫn: Huỳnh Thị Thanh Thương (viết sai tên trừ điểm)

Nhóm thực hiện:

1. Nguyễn Văn A 18001874

2. ...

TP.HCM, ngày ...tháng ...năm

CÁCH THỨC NỘP BÀI

- Nộp trên course
- Cú pháp đặt tên file nén: [HW01] Kỹ thuật sơ cấp (Ho ten-MSSV trưởng nhóm)
- GV không nhận bài nộp trễ qua mail với bất kỳ lí do gì. Hãy làm trước đó, không để phút cuối.
- Làm CÁ NHÂN hoặc theo nhóm đã đăng ký (nhóm có thể tan rã sau bài tập này do hợp tác không hiệu quả, khi có sự điều chỉnh/ cập nhật gì về nhóm thì chỉ cần chỉnh sửa lại file excel/google sheet và ghi chú bên cạnh cho GV nắm thông tin).
- Trường hợp chưa có nhóm thì có thể làm cá nhân. Vì số câu hỏi nhiều nên làm cá nhân e rằng không xuể, khuyến khích lập nhóm để share task và kiểm tra chéo lẫn nhau, sửa lỗi cho nhau
- Nếu làm theo nhóm thì chỉ 1 người đại diện nhóm nộp bài.
- Nếu không có gì đặc biệt thì GV đánh giá điểm số cho các thành viên trong nhóm là như nhau. Nếu đóng góp của mỗi bạn khác nhau, có nhiều chênh lệch và nhóm muốn đánh giá riêng theo năng lực từng người, thì trong bài nộp bổ sung “Bảng phân công công việc và tự đánh giá % đóng góp của từng thành viên, ai làm nhiều nhất là đóng góp 100%, làm ít hơn thì 90%, 80%, 50%, 20% ...)
- Thời gian làm bài là 2 tuần.
- Trong các buổi học tại lớp, SV có thể hỏi GV về bài tập về nhà: nhờ gợi ý cách giải, hỏi về cách trình bày, xin gia hạn nếu làm không kịp, trình bày những khó khăn của mình để được tư vấn, hướng dẫn, hỗ trợ,
- Nếu làm CÁ NHÂN và làm không nổi vì số lượng bài nhiều quá thì có thể Thương lượng với GV để tìm giải pháp phù hợp cho mình, ví dụ như giảm số lượng, hoặc tăng thời gian,

Nhắc lại bài học tuần 1&2

- ❖ Mathematical analysis = Count the number of times “each of operations” or “basic operation” is executed.
- ❖ (Dịch: Đánh giá tính hiệu quả về mặt thời gian của thuật toán bằng phương pháp toán học = Đếm tất cả số phép toán hoặc số phép toán cơ bản được thực hiện trong thuật toán đó)
- ❖ Basic operation = the most important operation, contribute the most to total running time
- ❖ (Dịch: Phép toán cơ bản là phép toán quan trọng nhất, “tích cực nhất”, tốn nhiều chi phí thực hiện nhất hoặc có thể hiểu là phép toán mà số lần thực hiện không ít hơn các phép toán khác)

Cấp số cộng

❖ **Cấp số cộng**: dãy số thoả mãn điều kiện: 2 phần tử liên tiếp nhau sai khác nhau một hằng số (công sai)

- **Số hạng thứ n :**

$$a_n = a_1 + (n - 1)d$$

- **Tổng của n số hạng đầu:**

$$S_n = a_1 + a_2 + \cdots + a_n = \frac{n(a_1 + a_n)}{2}$$

(a_1 : phần tử đầu, d : công sai)

$$= \frac{n[2a_1 + (n - 1)d]}{2}$$

Cấp số nhân

❖ **Cấp số nhân**: dãy số thoả mãn điều kiện: tỷ số của hai phần tử liên tiếp là hằng số (công bội)

- **Số hạng thứ n:**

$$a_n = a r^{n-1}$$

- **Tổng các phần tử của cấp số nhân:**

$$S_{n+1} = \sum_{k=0}^n ar^k = \frac{a(1 - r^{n+1})}{1 - r}$$

(*a*: phần tử đầu, *r*: công bội)

$$\sum_{k=m}^n ar^k = \frac{a(r^m - r^{n+1})}{1 - r}$$

Tổng hữu hạn

❖ Tính chất:

Sum Manipulation Rules

1.
$$\sum_{i=l}^u ca_i = c \sum_{i=l}^u a_i$$
2.
$$\sum_{i=l}^u (a_i \pm b_i) = \sum_{i=l}^u a_i \pm \sum_{i=l}^u b_i$$
3.
$$\sum_{i=l}^u a_i = \sum_{i=l}^m a_i + \sum_{i=m+1}^u a_i, \text{ where } l \leq m < u$$
4.
$$\sum_{i=l}^u (a_i - a_{i-1}) = a_u - a_{l-1}$$

Tổng hữu hạn

❖ Một số công thức cần nhớ:

Important Summation Formulas

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

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

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

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

Tổng hữu hạn

❖ Một số công thức cần nhớ:

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

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

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

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

[1]Anany Levitin, Introduction to the Design and Analysis of Algorithms, 3rd Edition, 2014 (trang 476)

Cận trên, cận dưới

Floor and Ceiling Formulas

floor of a real number x , denoted $\lfloor x \rfloor$, $\lfloor 3.8 \rfloor = 3$, $\lfloor -3.8 \rfloor = -4$, $\lfloor 3 \rfloor = 3$

ceiling of a real number x , denoted $\lceil x \rceil$, $\lceil 3.8 \rceil = 4$, $\lceil -3.8 \rceil = -3$, $\lceil 3 \rceil = 3$

1. $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$
2. $\lfloor x + n \rfloor = \lfloor x \rfloor + n$ and $\lceil x + n \rceil = \lceil x \rceil + n$ for real x and integer n
3. $\lfloor n/2 \rfloor + \lceil n/2 \rceil = n$
4. $\lceil \lg(n + 1) \rceil = \lfloor \lg n \rfloor + 1$

Logarithms

Properties of Logarithms

All logarithm bases are assumed to be greater than 1 in the formulas below; $\lg x$ denotes the logarithm base 2, $\ln x$ denotes the logarithm base $e = 2.71828 \dots$; x, y are arbitrary positive numbers.

1. $\log_a 1 = 0$
2. $\log_a a = 1$
3. $\log_a x^y = y \log_a x$
4. $\log_a xy = \log_a x + \log_a y$
5. $\log_a \frac{x}{y} = \log_a x - \log_a y$
6. $a^{\log_b x} = x^{\log_b a}$
7. $\log_a x = \frac{\log_b x}{\log_b a} = \log_a b \log_b x$

Bài 1: Tính tổng hữu hạn

1.1 Yêu cầu: Tính chính xác, không cho phép sai số hay xấp xỉ

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

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

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

i. $\sum_{j \in \{2,3,5\}} (j^2 + j)$

j. $\sum_{i=1}^m \sum_{j=0}^n \sum_{k=0}^{100} (i+j)$

Exercises 2.3, trang 67

[1]Anany Levitin, Introduction to the Design and Analysis of Algorithms, 3rd Edition, 2014

Bài 1: Tính tổng hữu hạn

1.2 Yêu cầu: Tính chính xác được thì càng tốt, không thì cho phép tính gần đúng/xấp xỉ

a. $\sum_{i=0}^{n-1} (i^2 + 1)^2$

b. $\sum_{i=2}^{n-1} \lg i^2$

c. $\sum_{i=1}^n (i + 1)2^{i-1}$

d. $\sum_{i=0}^{n-1} \sum_{j=0}^{i-1} (i + j)$

Exercises 2.3, trang 67

[1]Anany Levitin, Introduction to the Design and Analysis of Algorithms, 3rd Edition, 2014

Đếm số phép gán và so sánh

Bài 2

Yêu cầu:

- phải khử được tổng Σ và KHÔNG CẦN rút gọn
- Khử chính xác, không dùng xấp xỉ

```
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

Bài 3

Lưu ý:

- Cần có biện luận chặt chẽ để tìm số lần lặp của while trong
- Phải khử được tổng Σ và không cần rút gọn
- Khi khử tổng Σ , nếu có áp dụng công thức nào bên ngoài (khác các công thức GV cung cấp trong slide này) thì phải có tham chiếu đầy đủ, tức nói rõ trích từ tài liệu nào, chụp hình công thức gốc đưa vào bài làm

```
float Alpha (float x, long n)
{
    long i= 1; float z = 0;
    while ( i ≤ n)
    {
        long j = 1; float t = 1;
        while (j ≤ i)
        {
            t = t*x;
            j = 2*j;
        }
        z = z+i*t;
        i=i+1;
    }
    return z;
}
```

Đếm số phép gán và so sánh

Bài 4

Đề thi cuối kỳ

Lưu ý: Cần có biện luận chặt chẽ để tìm số lần lặp của while trong 1 và while trong 2

- phải khử tổng Σ và không cần rút gọn
- Khi khử tổng Σ , nếu có áp dụng công thức nào bên ngoài (khác các công thức GV cung cấp trong slide này) thì phải có tham chiếu đầy đủ, tức nói rõ trích từ tài liệu nào, chụp hình công thức gốc đưa vào bài làm

```
sum = 0; i = 1;
while ( i ≤ n )
{
    j = n - i ;
    while ( j ≤ 2* i )
    {
        sum = sum + i*j;
        j = j + 2;
    }
    k = i ;
    while ( k > 0 )
    {
        sum = sum + 1;
        k = k / 2;
    }
    i = i + 1;
}
```

Đếm số phép gán và so sánh

Bài 5(*)

Lưu ý:

- Cần có biện luận chặt chẽ để tìm số lần lặp của while trong
- Phải khử tổng Σ và rút gọn biểu thức gán, so sánh
- Nếu không khử Σ hay rút gọn sẽ bị trừ điểm

```
i=1;  
res=0;  
while i≤n do  
    j=1;  
    k=1;  
    while j ≤ i do  
        res=res+i*j;  
        k=k+2;  
        j=j+k;  
    endw  
    i=i+1;  
endw
```

P_i →

Đếm số phép gán và so sánh

Bài 6(*)

- KHÔNG CẦN KHỬ tổng Σ và không cần rút gọn \rightarrow dừng lại tại bước tổng hợp thành tổng Σ là được
- Cần có biện luận chặt chẽ để tìm số lần lặp của while trong và số lần thực hiện các câu lệnh if, câu lệnh gán trong thân if
- Phải lập bảng xét dấu, đưa bảng vào trong bài làm, để người đọc hiểu được tại sao i chạy từ a \rightarrow b

```
i=1;
count = 0;
while (i<=4n)
{
    x=(n-i) (i-3n)
    y=i-2n
    j=1
    while (j<=x)
    {
        if (i>=2y)
            count = count -2
        j=j+1
    }
    i=i+1
}
```

Đếm số phép gán và so sánh

Bài 7(*)

- KHÔNG CẦN KHỬ
tổng Σ và không cần rút
gọn \rightarrow dừng lại tại bước
tổng hợp thành tổng Σ
là được

- Cần có biện luận chặt chẽ để
tìm số lần lặp của while trong
và số lần thực hiện các câu
lệnh if, câu lệnh gán trong
thân if

- Phải lập bảng xét dấu, đưa
bảng vào trong bài làm, để
người đọc hiểu được tại sao i
chạy từ a \rightarrow b

```
i = 1; count=0;
while (i<=4*n)
{
    x=(n-i) * (i-3*n);
    y=i-2*n;
    j=1;
    while (j<=x)
    {
        if (i>=2*n)
            count=count-2;
        j++;
    }
    if (x>0)
        if (y>0)
            count=count+1;
    i++;
}
```

Đếm số phép gán và so sánh

Bài 8(*)

Đề thi cuối kỳ (CT Tài năng/CLC)

- KHÔNG CẦN KHỬ
tổng Σ và không cần rút
gọn \rightarrow dừng lại tại bước
tổng hợp thành tổng Σ
là được

- Cần có biện luận chặt chẽ để
tìm số lần lặp của while trong
và số lần thực hiện các câu
lệnh if, câu lệnh gán trong
thân if

- Phải lập bảng xét dấu, đưa
bảng vào trong bài làm, để
người đọc hiểu được tại sao i
chạy từ a \rightarrow b

```
i= 1;count = 0;
while (i ≤ 3*n)
{
    x = 2*n - i;
    y = i - n ;
    j = 1;
    while (j ≤ x)
    {
        if(j ≥ n)
            count = count - 1;
        j = j+1;
    }
    if (y > 0)
        if (x > 0)
            count = count + 1;
    i = i+1;
}
```


Đếm số phép gán và so sánh

Đề thi cuối kỳ (CT Tài năng/CLC) HỌC KỲ 2 – NĂM HỌC 2021-2022

Bài 9(*)

- Phải có lập luận chặt chẽ trong bài làm
- Phải khử tổng Σ và rút gọn biểu thức gán, so sánh. Nếu không khử Σ hay rút gọn sẽ bị trừ điểm
- Nếu có quy ước gì đặc biệt thì phải ghi rõ trong bài làm
Ví dụ: “Quy ước: chỉ xét những biểu thức so sánh tường minh”

```
sum = 0; i=1; idx=-1;
while (i<=n)
{
    j=1;
    while(j<=n)
    {
        if( (i==j) && (i+j==n+1) )
            idx=i;
        sum=sum+a[i][j];
        j++;
    }
    i++;
}
if(idx !=-1)
    sum=sum-a[idx][idx];
```


Đếm số phép gán và so sánh

Bài 10(**)

```
i = 1; ret = 0; s = 0;
while ( i ≤ n)
{
    j = 1;
    s = s + 1/i; // {số thực}
    while (j ≤ s)
    {
        ret = ret + i*j;
        j = j + 1;
    }
    i = i + 1;
}
```

Bài 11(**): Bonus, không bắt buộc

Cho n, m là các số nguyên dương. Xem xét thuật toán sau đây, giả sử phép gán và phép so sánh đóng vai trò chủ yếu trong thời gian chạy của thuật toán

```
sum=0;
i=1;
while (i<=m)
{
    j=1;
    while (j<=n)
    {
        if (i!=j)
            sum=sum+1;
        j++;
    }
    i++;
}
```

Bài 11(**): Bonus, không bắt buộc

a) Đếm thủ công (tức đếm về mặt lý thuyết) số phép gán và số phép so sánh trong đoạn chương trình trên, yêu cầu giống như những bài trước là có lập luận trong quá trình đếm, không cần khử tổng, **dừng lại tại bước tổng hợp thành tổng Σ là được**

b) **Kiểm tra kết quả đếm bằng máy tính:**

- Bước 1: SV tự tính toán thủ công theo công thức ở câu a (chỉ là thế giá trị của n, m vào các biểu thức rồi cho biết kết quả), tự chọn 5 bộ dữ liệu thử nghiệm. Ví dụ, $(n=10, m=20)$, $(n=35, m=24)$, +

- Bước 2: Viết chương trình C++ cài đặt thuật toán và dùng các biến đếm hỗ trợ để máy tính đếm tự động từng câu lệnh gán và so sánh khi thực thi

 - + Phải copy code, đưa code vào bài làm

 - + Cho biết kết quả chạy chương trình

- Bước 3: Thống kê theo dạng bảng (ở slide sau) để so sánh kết quả của Bước 1 và Bước 2,

BONUS Không bắt buộc

Ví dụ 5 bộ dữ liệu thử nghiệm (n,m)	(10, 20)	(35, 24)	...	(100, 200)
Gán(n)= ...(theo công thức đếm thủ công từ câu a)				
Gán (n) kết quả khi chạy chương trình ở Bước 2				
So sánh (n)= ...(theo công thức đếm thủ công từ câu a)				
So sánh (n) kết quả khi chạy chương trình				

Nhận xét kết quả so sánh: có bằng nhau không, nếu không bằng thì tại sao?