KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
227 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM
Điện Thoại: (08) 38.354.266 - Fax:(08) 38.350.096

CTT308 – NHẬP MÔN PHÂN TÍCH ĐỘ PHỨC TẠP THUẬT TOÁN

BÀI TẬP LÝ THUYẾT VỀ NHÀ

BTVN#A1: PHÂN LOẠI HÀM THEO O LỚN và Θ LỚN

I. Thông tin chung

Mã số bài tập:	BTVN#A1
Thời lượng dự kiến:	3 giờ
Deadline nộp bài:	
Hình thức:	Bài tập cá nhân
Hình thức nộp bài:	Nộp qua Moodle môn học
GV phụ trách:	
Thông tin liên lạc với GV:	

II. Chuẩn đầu ra cần đạt

Bài tập này nhằm mục tiêu đạt được các chuẩn đầu ra sau:

- G2
- G3.1, G3.2, G3.3

III. Mô tả bài tập

Mỗi sinh viên làm 2 bài tập trong số những bài tập sau đây.

1. Cho hai hàm $f(n) = 2000n^2 + 9000$ và $g(n) = n^3 - 70n^2$. Chứng minh $f = O(g)$ và $g \neq O(f)$ dựa trên định nghĩa của ký hiệu O lớn.

2. Cho hai hàm $f(n) = 20000n^3 + 90$ và $g(n) = n^4 - 700n^2$. Chứng minh $f = O(g)$ và $g \neq O(f)$ dựa trên định nghĩa của ký hiệu O lớn.

3. Chứng minh $n^3 - 70n^2 = \Theta(n^3)$.

4. Chứng minh $n^4 - 700n^2 = \Theta(n^4)$.

5. Cho hai hàm $f(n) = 20000n^2\sqrt{n} + 3000$ và $g(n) = n^3 - 80000n^2 + 100$. Chứng minh $f = O(g)$ và $g \neq O(f)$.

6. Cho hàm $f(n) = \dfrac{1}{\sqrt{1}} + \dfrac{1}{\sqrt{2}} + \dfrac{1}{\sqrt{3}} + ... + \dfrac{1}{\sqrt{n}}$.

   a) Chứng minh $f = \Omega(\sqrt{n})$.

   b) Có chắc là $f = \Theta(\sqrt{n})$ hay không? Bạn hãy chứng minh hoặc phủ nhận (một cách chặt chẽ) điều này.

IV. Các yêu cầu & quy định chi tiết cho bài nộp

Trình bày bài nộp tự do, không theo mẫu, ghi họ tên và mã số sinh viên.

V. Cách đánh giá

2% điểm môn học

VI. Tài liệu tham khảo

Xem các slides bài giảng về O lớn và Θ lớn.

VII. Các quy định khác

Không có

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
227 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM
Điện Thoại: (08) 38.354.266 - Fax:(08) 38.350.096

CTT308 – NHẬP MÔN PHÂN TÍCH ĐỘ PHỨC TẠP THUẬT TOÁN

BÀI TẬP LÝ THUYẾT VỀ NHÀ

BTVN#A2: BÀI TẬP NGHIÊN CỨU VỀ SO SÁNH ĐỘ LỚN HÀM

I. Thông tin chung

Mã số bài tập:	BTVN#A2
Thời lượng dự kiến:	8 giờ
Deadline nộp bài:	
Hình thức:	Bài tập cá nhân
Hình thức nộp bài:	Nộp qua Moodle môn học
GV phụ trách:	
Thông tin liên lạc với GV:	

II. Chuẩn đầu ra cần đạt

Bài tập này nhằm mục tiêu đạt được các chuẩn đầu ra sau:

- G3.1, G3.2, G3.3

III. Mô tả bài tập

Mỗi sinh viên chọn một trong số những bài tập sau đây, làm và nộp đúng theo thời gian qui định. Sinh viên có thể làm hết các câu của mỗi bài tập hoặc có thể bỏ bớt những câu cảm thấy quá sức mình.

1. Cho hàm $f(n) = 1^k + 2^k + 3^k + ... + n^k$.

   a) Hãy chứng minh $f = O(n^{k+1})$ khi k là một số không âm.

   b) Chứng minh (hoặc phản chứng minh): $f = \Theta(n^{k+1})$ khi k là một số không âm.

   c) Khảo sát trường hợp $k = -\frac{1}{3}$, nghĩa là $f(n) = \frac{1}{\sqrt[3]{1}} + \frac{1}{\sqrt[3]{2}} + \frac{1}{\sqrt[3]{3}} + ... + \frac{1}{\sqrt[3]{n}}$.

   d) Nghiên cứu trường hợp tổng quát của câu trên, nghĩa là $-1 < k < 0$.

   e) Trường hợp $k = -1$ thì sao?

   f) Trường hợp $k < -1$ thì sao?

2. Xét hàm $f(n) = \log(n)$ (hàm logarit cơ số $\boldsymbol{e} \approx$ 2,71828...) và $g(n) = \sqrt[15]{n}$.

   a) Chứng minh $f = O(g)$.

   b) Chứng minh $g \neq O(f)$.

c) Tìm một giá trị nguyên dương của n mà ta có $f(n) < g(n)$. Bạn có thể viết chương trình để làm câu này nếu bị trở ngại khi tính toán lý thuyết.

d) Khảo sát đại lượng $\frac{f(n)}{g(n)} = \frac{\log(n)}{\sqrt[15]{n}}$ để so sánh độ lớn của hai hàm $f(n)$ và $g(n)$ tùy theo n. Bạn viết chương trình để làm câu này. Bạn cần suy nghĩ để chọn các khoảng giá trị thích hợp của n để trình bày kết quả so sánh một cách có ý nghĩa. Vẽ các đồ thị minh họa.

3. Hãy cho ví dụ về hai hàm $f(n)$ và $g(n)$ thỏa tất cả những điều kiện sau đây:

- Cả hai hàm thỏa mãn: $f(n) = \Omega(n)$ và $g(n) = \Omega(n)$;
- Hai hàm không so sánh nhau được theo O lớn, nghĩa là: $f \neq O(g)$ và $g \neq O(f)$;
- Hai hàm này lại trội bởi $n\log(n)$, nghĩa là $f = O(n\log(n))$ và $g = O(n\log(n))$.

a) Bạn cần chứng minh một cách chặt chẽ rằng hai hàm mà bạn đề xuất thỏa mãn tất cả những điều kiện nói trên.

b) Cho ví dụ về hai thuật toán có độ phức tạp lần lượt là $\Theta(f)$ và $\Theta(g)$.

IV. Các yêu cầu & quy định chi tiết cho bài nộp

Trình bày bài nộp tự do, không theo mẫu, ghi họ tên và mã số sinh viên.

V. Cách đánh giá

2% điểm môn học

VI. Tài liệu tham khảo

Xem các slides bài giảng về O lớn và Θ lớn.

VII. Các quy định khác

Không có

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
227 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM
Điện Thoại: (08) 38.354.266 - Fax:(08) 38.350.096

CTT308 – NHẬP MÔN PHÂN TÍCH ĐỘ PHỨC TẠP THUẬT TOÁN

BÀI TẬP LÝ THUYẾT VỀ NHÀ

BTVN#A3: PHÂN TÍCH THUẬT TOÁN KHÔNG CÓ YẾU TỐ NGẪU NHIÊN

I. Thông tin chung

Mã số bài tập:	BTVN#A3
Thời lượng dự kiến:	3 giờ
Deadline nộp bài:	
Hình thức:	Bài tập cá nhân
Hình thức nộp bài:	Nộp qua Moodle môn học
GV phụ trách:	
Thông tin liên lạc với GV:	

II. Chuẩn đầu ra cần đạt

Bài tập này nhằm mục tiêu đạt được các chuẩn đầu ra sau:

- G3.3, G4.1, G4.2, G4.3

III. **Mô tả bài tập**: Mỗi sinh viên chọn 2 trong số bài tập dưới đây để làm và nộp theo thời gian qui định.

Đếm số phép gán và số phép so sánh của các thuật toán mô tả bằng mã giả hay đã cài đặt mã nguồn, so sánh kết quả đếm lý thuyết và kết quả viết chương trình thử nghiệm.

1)

```
S:=0; count:=n*n-7*n; i:=1;
while i<=n do
   j:=1;
   while j<=i*i do
       if (j>=i) and (count>0) then
              S := S+j;
              count := count-2;
       endif;
       count := count+1;
       j := j+1;
   endw;
   i := i+1;
endw
```

2)

```
S:=0; count:=n; i:=1;
while i<=n do
  j:=1;
  while j<= i*i do
     if (j>=i) and (count>0) then
           S: = S+i*j;
           count := count-1;
     endif;
     j := j+1;
  endw;
  i := i+1
endw;
```

3)

```
sum :=0; i:=1;
while i<=n do
     j:=1;
     while j<=n do
        if (i=j) or (i+j=n+1) then
           sum := sum+a[i,j];
        endif;
        j := j+1;
     endw;
     i := i+1;
endw;
```

4)

```
sum :=0; i:=1;
while i<=n do
     j := i div 2;
     while j<=n-i+1 do
        if i<>j then
           sum := sum+a[i,j];
        endif;
        j := j+1;
     endif;
     i := i+1;
endw;
```

5)

```
sum :=0; i:=1; idx=-1;
while i<=n do
      j:=1;
      while j<=n do
         if (i=j) and (i+j=n+1) then
            idx=i;
         endif;
         sum := sum+a[i,j];
         j := j+1;
      endw;
      i := i+1;
endw;
if idx<>-1 then
  sum := sum - a[idx,idx];
endif;
```

6)

```
count :=0; i:=n;
while i>0 do
   if (i mod 4) = 1 then
      j:=i;
      while j<=i*i do
         if j mod 2=0 then
            count := count+1;
         endif;
         j:=j+1;
      endw;
   endif;
   i := i - 16;
endw;
```

a) Với n=101 thì giá trị của biến count là bao nhiêu khi thuật toán kết thúc?

b) Tính theo n số phép so sánh và số phép gán được thực hiện

c) Xác định độ phức tạp của thuật toán tùy theo n.

7)

```
i:=1; count:=0;
while i<=n*n do
      x := (n - i)*(i - 3*n); y := i - 2*n;
      j := 1;
      while j<=x do
         if i>=2*n then
            count := count-2;
         endif;
         j := j+1;
      endw;
      if x>0 then
         if y>0 then
            count := count+1;
         endif
      endif;
      i := i+1;
endw;
```

8)

```
i:=1; count:=0;
while i<=n*n - n do
      x := 2*n - i; y := i - n;
      j := 1;
      while j<=x do
         if j>=n then
            count := count-1;
         endif;
         j := j+1;
      endw;
      if y>0 then
         if x>0 then
            count := count+1;
         endif;
      endif;
      i := i+1;
endw;
```

9)

```
i:=1
count:=0;
while i<=n*n+n do
      x := (n - i)*(i - 3*n);  y := i - 2*n
      j := 1
      while j<=x do
         if i>=2*y then
            count := count-2
         endif
         j := j+1
      endw
      if y>0 then
         count := count+1
      endif
      i := i+1
endw
```

Xem các hàm viết bằng ngôn ngữ C sau đây. Hãy ước lượng số phép so sánh và số phép gán của thuật toán. Suy ra độ phức tạp.

10)

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

11)

```
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;
}
```

12)

```
float Epsi(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 = j + 1;
          }
          z = z + i*t;
          i = 2*i;
      }
      return z;
}
```

13)

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

14)

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

15)

```
float Phi(float x, long n){
    long i=1; float z=0;
    while(i<=n){
        long j=1; float t=1;
        while(j-i<=i*i){
            if(j<i)
                t = t/2;
            t = t*x;
            j = j + 1;
        }
        z = z + i*t; i = 2*i;
    }
    return z;
}
```

16)

```
float Si(float x, long n) {
    long i=1; float z=0;
    while(i<=n){
        long j=1; float t=1;
        while(j-i<=i*i*i){
            if(j>=i*i)
                t = t/2;
            t = t*x;
            j = j + 1;
        }
        z = z + i*t; i = 2*i;
    }
    return z;
}
```

17) Giả sử n là một số nguyên dương và p là số nguyên lớn hơn 1 cho trước.
Xem thuật toán viết bằng mã giả sau đây:

```
count :=0; i:=n; sum := 0;
while i>0 do
   if i mod p =0 then
        count := count + 1;
   else
        temp := i mod p;
        sum := sum + temp ;
   endif
   i := i div p;
endw;
```

a) Gọi $\propto$ là số lần thực hiện vòng lặp while. Xác định theo $\propto$ số phép so sánh và số phép gán được thực hiện
b) Tính $\propto$ theo n.
c) Xác định độ phức tạp của thuật toán.

IV. Các yêu cầu & quy định chi tiết cho bài nộp

Trình bày bài nộp tự do, không theo mẫu, ghi họ tên và mã số sinh viên.

V. Cách đánh giá

2% điểm môn học

VI. Tài liệu tham khảo

Xem các slides bài giảng về tiếp cận lý thuyết để đánh giá thuật toán (chủ đề 4).

VII. Các quy định khác

Không có

CTT308 – NHẬP MÔN PHÂN TÍCH ĐỘ PHỨC TẠP THUẬT TOÁN

BÀI TẬP LÝ THUYẾT VỀ NHÀ

BTVN#A4: BÀI TẬP CƠ BẢN VỀ ÁP DỤNG HÀM SINH

I. Thông tin chung

Mã số bài tập:	BTVN#A4
Thời lượng dự kiến:	4 giờ
Deadline nộp bài:	
Hình thức:	Bài tập cá nhân
Hình thức nộp bài:	Nộp qua Moodle môn học
GV phụ trách:	
Thông tin liên lạc với GV:	

II. Chuẩn đầu ra cần đạt

Bài tập này nhằm mục tiêu đạt được các chuẩn đầu ra sau:

- G7.1, G7.2, G7.3

III. Mô tả bài tập

Mỗi sinh viên chọn ba trong số những bài tập sau đây, phải chọn ít nhất một bài có phân tích thuật toán, làm và nộp đúng theo thời gian qui định.

1. Dùng phương pháp hàm sinh để tìm công thức tính cho dãy:

   $a_{n-2} = (a_n + a_{n-1})/6$
   $a_0 = 2$, $a_1 = -1$

2. Dùng phương pháp hàm sinh để tìm công thức tính cho dãy:

   $a_{n+3} = (a_{n+2} + 2a_{n+1} - a_n)/2$
   $a_0 = 0$, $a_1=1$, $a_2 = 2$

3. Dùng phương pháp hàm sinh để tìm công thức tính cho dãy:
   $a_{n+2} = 4(a_{n+1} - a_n)$
   $a_0 = 1$, $a_1=3$

4. Dùng phương pháp hàm sinh để tìm công thức tính cho dãy:
   $a_{n+2} + Ba_{n+1} + Ca_n = 0$
   $a_0 = 0$, $a_1 = 1$, $a_2 = 4$, $a_3 = 37$

5. Xem bài toán tháp Hà nội. Gọi a_n là tổng số lần chuyển đĩa đối với trường hợp n đĩa.

   - Viết chương trình đệ qui để giải bài toán và chạy thử.

   - Hãy xác định công thức truy hồi của dãy $\{a_n\}$

   - Dùng phương pháp hàm sinh để xác định a_n theo n.

6. Giả sử tập S gồm n phần tử. Thuật toán đệ qui sau đây nhằm mục đích tìm phần tử lớn nhất của S:

- Chia S thành 2 tập rời nhau S1 và S2, mỗi tập có phân nửa số phần tử;
- Tìm phần tử lớn nhất trong mỗi tập (đệ qui);
- So sánh 2 phần tử để tìm phần tử lớn nhất của S.

Hãy viết chương trình đệ qui cài đặt thuật toán trên và chạy thử. Gọi a_n là số phép so sánh cần thiết, dùng phương pháp hàm sinh để tính a_n theo n.

7. Xem dãy Fibonacci xác định theo công thức:
   $F_0 = F_1 = 1$,
   $F_n = F_{n-1} + F_{n-2}$ nếu n>1.

   a) Viết chương trình đệ qui, chạy thử để tính F_n. Gọi a_n là số phép cộng cần thiết trong thuật toán. Dùng phương pháp hàm sinh để tính a_n theo n.

   b) Viết chương trình không đệ qui (dùng 3 biến U, V, W để lưu 3 giá trị liên tiếp F_n, F_{n-1}, F_{n-2} để đỡ tính lặp đi lặp lại). Gọi b_n là số phép cộng cần thiết trong thuật toán. Hãy tính F_n theo n.

   c) So sánh độ phức tạp của hai thuật toán trên.

8. Xét hai dãy $\{X_n\}$, $\{Y_n\}$ được định nghĩa như sau:
   $X_0 = Y_0 = 1$
   $X_n = X_{n-1} + Y_{n-1}$
   $Y_n = n^2 X_{n-1} + Y_{n-1}$
   Dùng phương pháp hàm sinh để tìm công thức tính cho 2 dãy nói trên.

9. Xem hàm viết bằng ngôn ngữ C/C++ sau đây:

```
float Zeta(float x, long n) {
        if(n<=0)
            return 1;
        float z=0; long i=1;
        while(i<=8){
            long m=n-1;
            z += Zeta(x, m);
            x /= 2; i ++;
        }
        i=1;
        while(i<=n){
            z /= i; i ++;
        }
        return z;
}
```

   Gọi Z_n là số so sánh được thực hiện. Thiết lập công thức truy hồi tính Z_n và dùng phương pháp hàm sinh để tìm công thức tính Z_n theo n.

10. Xem hàm viết bằng ngôn ngữ C/C++ sau đây:

```
float Beta(float x, long n){
        if(n<=1)
            return 1;
        float z=0; long i=1;
        while(i<=4){
            long m=n-1;
            z += Beta(x, m);
            x /= 2; i ++;
        }
        i=1;
        while(i<=4){
            long m=n-2;
            z += Beta(x, m);
            x /= 2; i ++;
        }
        i=1;
        while(i<=n){
            z /= i; i ++;
        }
        return z;
}
```

Gọi B_n là số so sánh được thực hiện. Thiết lập công thức truy hồi tính B_n và dùng phương pháp hàm sinh để tìm công thức tính B_n theo n.

11. Xem hàm viết bằng ngôn ngữ C sau đây:

```
float Gamma(float x, long n) {
        if(n<=0)
            return 1;
        float z=0; long i=1;
        while(i<=8){
            long m=n/2;
            z += Gamma(x, m);
            x /= 2; i ++;
        }
        return z;
}
```

Gọi G_n là số so sánh được thực hiện. Thiết lập công thức truy hồi tính G_n và dùng phương pháp hàm sinh để tìm công thức tính G_n theo n.

12. Dùng phương pháp hàm sinh để tìm công thức tính cho dãy:

$X_o=1$, $X_n=n^2X_o + (n-1)^2X_1 + \ldots + 1^2X_{n-1}$, nếu n >= 1.

13. Số lượng các tổ hợp chập k của n phần tử ($0 \le k \le n$) được xác định theo công thức:

$$C_n^k = \frac{n!}{k!(n-k)!}$$

Mặt khác ta cũng có công thức truy hồi như sau ($1 \le k \le n$):

$$C_n^0 = C_n^n = 1$$

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

a) Hãy viết 3 hàm tính C_n^k theo ba cách sau đây (và chạy thử):
   - Cách 1: viết hàm tính m! và dựa vào hàm nầy để viết hàm tính giá trị của C_n^k.
   - Cách 2: dùng hàm đệ qui.
   - Cách 3: dựa vào liên hệ giữa C_n^k và C_n^{k-1}.

b) Đánh giá độ phức tạp mỗi thuật toán bên trên (nhờ ước lượng số phép gán và số phép so sánh) và so sánh độ phức tạp của 3 thuật toán.

IV. Các yêu cầu & quy định chi tiết cho bài nộp

Trình bày bài nộp tự do, không theo mẫu, ghi họ tên và mã số sinh viên.

V. Cách đánh giá

2% điểm môn học

VI. Tài liệu tham khảo

Xem các slides bài giảng về các công cụ toán học để phân tích thuật toán (chủ đề 5).

VII. Các quy định khác

Không có

CTT308 – NHẬP MÔN PHÂN TÍCH ĐỘ PHỨC TẠP THUẬT TOÁN

BÀI TẬP LÝ THUYẾT VỀ NHÀ

BTVN#A5: PHÂN TÍCH THUẬT TOÁN CÓ YẾU TỐ NGẪU NHIÊN

I. Thông tin chung

Mã số bài tập:	BTVN#A5
Thời lượng dự kiến:	8 giờ
Deadline nộp bài:	
Hình thức:	Bài tập cá nhân
Hình thức nộp bài:	Nộp qua Moodle môn học
GV phụ trách:	
Thông tin liên lạc với GV:	

II. Chuẩn đầu ra cần đạt

Bài tập này nhằm mục tiêu đạt được các chuẩn đầu ra sau:

- G6.2, G6.3, G7.6, G8.5

III. Mô tả bài tập

Mỗi sinh viên chọn một trong số những bài tập sau đây, làm và nộp đúng theo thời gian qui định.

1. Cho n là một số nguyên dương, mảng a là dữ liệu nhập (các phần tử đôi một phân biệt) và mảng b là dữ liệu xuất. Xem thuật toán sau đây, giả sử phép gán và phép so sánh đóng vai trò chủ yếu trong thời gian chạy của thuật toán.

```
i:=0;
while i<n do
      Index[i] := 0;  i := i+1;
endw;
i:=0;
while i<n-1 do
      j:=i+1; t:=n-1-i;
      while j<n do
            if a[i]<a[j] then
                  Index[j] := Index[j]+1;
                  t := t-1;
            endif;
            j:=j+1;
      endw;
      Index[i] := Index[i]+t;
      i:=i+1;
endw;
i:=0;
while i<n do
      b[Index[i]] := a[i]; i:=i+1;
endw;
```

a) Khảo sát trường hợp tốt nhất và xấu nhất của thuật toán, các trường hợp nầy xảy ra với dạng dữ liệu nhập như thế nào?

b) Ước lượng về mặt lý thuyết số phép gán, số phép so sánh của thuật toán. Suy ra độ phức tạp của thuật toán.

2. Cho *n* là một số nguyên dương, mảng *a* là dữ liệu nhập (các phần tử đôi một phân biệt). Xem thuật toán sau đây, giả sử phép gán và phép so sánh đóng vai trò chủ yếu trong thời gian chạy của thuật toán.

```
i := 0
while i < n-1 do
      j := i+1
      while j < n do
            if a[i] > a[j] then
                 t := a[i]
                 a[i] := a[j]
                 a[j] := t
            endif
            j := j+1;
      endw
      i := i+1;
endw
```

a) Khảo sát trường hợp tốt nhất và xấu nhất của thuật toán, các trường hợp nầy xảy ra với dạng dữ liệu nhập như thế nào?

b) Ước lượng về mặt lý thuyết số phép gán, số phép so sánh của thuật toán. Suy ra độ phức tạp của thuật toán.

3. Xem thuật toán sau đây, giả sử phép gán và phép so sánh đóng vai trò chủ yếu trong thời gian chạy của thuật toán.

```
a[1] := n;  j := n-1;
while j >= 1 do
      i := n - j ;
      while i>b[j] do
            a[i+1] := a[i];
            i := i-1;
      endw
      a[i+1] := j;
      j := j-1;
endw
```

Trong đó mảng *b*, xem như dữ liệu nhập, các phần tử của mảng thỏa mãn điều kiện $0 \le b[i] \le n - i$ với mỗi $i \in \{1, 2, 3, \ldots, n\}$. Mảng *a* xem như dữ liệu xuất.

a) Khảo sát trường hợp tốt nhất và xấu nhất của thuật toán, các trường hợp nầy xảy ra với dạng dữ liệu nhập như thế nào?

b) Ước lượng về mặt lý thuyết số phép gán, số phép so sánh của thuật toán. Suy ra độ phức tạp của thuật toán.

c) Đề xuất cách cải tiến thuật toán và cài đặt minh họa.

4. Cho *n* là một số nguyên dương, *a* là một mảng gồm các số đôi một khác nhau. Xem thuật toán sau đây, giả sử phép gán và phép so sánh đóng vai trò chủ yếu trong thời gian chạy của thuật toán.

```
i := n-1; test := TRUE;
while i>0 do
      j := i-1; dem :=0;
      while j>=0 do
            if a[i]>a[j] then
                 dem := dem+1;
            endif
            j := j-1;
      endw;
      if dem<i then
            test := FALSE;
            i := -1;
      else
            i := i-1;
      endif
endw;
```

a) Khảo sát trường hợp tốt nhất và xấu nhất của thuật toán, các trường hợp nầy xảy ra khi nào?

b) Ước lượng về mặt lý thuyết số phép gán, số phép so sánh trung bình của thuật toán. Suy ra độ phức tạp của thuật toán.

5. Cho n là một số nguyên dương, a là một mảng n mẩu tin. Xem thuật toán sau đây, giả sử phép gán và phép so sánh đóng vai trò chủ yếu trong thời gian chạy của thuật toán. Hàm random(a, b) trả về số nguyên ngẫu nhiên x sao cho $a \le x \le b$.

```
i := 0; s := 0;
while i<n do
     a[i].key := s;
     i := i+1;
     s := s+i;
endw;
K := random(1, s);
i := 0;
while (i<n) and (K>a[i].key) do
       i := i+1;
endw;
```

a) Khảo sát trường hợp tốt nhất và xấu nhất của thuật toán, các trường hợp nầy xảy ra khi nào?

b) Ước lượng về mặt lý thuyết số phép gán, số phép so sánh trung bình của thuật toán. Suy ra độ phức tạp của thuật toán.

IV. Các yêu cầu & quy định chi tiết cho bài nộp

Trình bày bài nộp tự do, không theo mẫu, ghi họ tên và mã số sinh viên.

V. Cách đánh giá

2% điểm môn học

VI. Tài liệu tham khảo

Xem các slides bài giảng về các công cụ toán học để phân tích thuật toán (chủ đề 5).

VII. Các quy định khác

Không có