Bài tập nhóm môn **Phân tích và thiết kế thuật toán**

GVLT: **Huỳnh Thị Thanh Thương**

Mã lớp: **CS112.J11S**

Thành viên nhóm:

**Hoàng Minh Anh – 16520035**

**Phan Hoàng Ân – 16520017**

**BÀI TẬP 1.B**

**Kỹ thuật sơ cấp 2 – Xét dấu hàm**

**Câu 1:**

1.1- Dùng kỹ thuật toán sơ cấp để Đếm số phép gán và số phép so sánh trong các đoạn chương trình trên. Suy ra độ phức tạp

**a)**

float Alpha (float x, long n)

{ long i= 1; float z = 0; (2 g)

while ( i ≤ n) (n+1 ss)

{ long j = 1; float t = 1; (2n g)

while (j ≤ i ) (αi + 1 ss)

{ t = t\*x; (αi g)  
 j = 2\*j; (αi g)

}

z = z+i\*t; (2n g)

i=i+1;

}

return z;

}

Số lần lặp của vòng while ngoài là số con i với i = 1 → n

🡺 while ngoài có n lần lặp 🡪 n + 1 phép so sánh

Gọi αi là số lần lặp của vòng while trong (độc lập với while ngoài)

αi bằng số con j với j = 1 → i; j = 2\*j

αi bằng số phần tử của tập {1 ; 2 ; 22 ; 23 ;… }

Gán(n) =

Sosánh(n) =

T(n) = Gán(n) + Sosánh(n) =

\* Tính với

Ta có (1 số)

(2 số)

(22 = 4 số)

(23 = 8 số)

…

(2k-1 số)

(n – 2k + 1 số)

với

(áp dụng công thức )

với

Sosánh(n) = với

Gán(n) = với

T(n) = với

🡺

**b)**

sum = 0; i = 1; (2 g)

while ( i ≤ n) (n+1 ss)

{ j = n – i ; (n g)

while (j ≤ 2\* i ) (αi + 1 ss)

{ sum = sum + i\*j; (αi + g)  
 j = j + 2; (αi + g)

} **Ai**  
 k = i ; (n g)

while ( k > 0) (βi + 1 ss)

{ sum = sum + 1; (βi g)  
 k = k / 2; (βi g)

} **Bi**

i = i + 1; (n g)

}

Số lần lặp của vòng while ngoài là số con i với i = 1 → n

🡺 while ngoài có n lần lặp 🡪 n + 1 phép so sánh

Gọi αi là số lần lặp của đoạn mã **Ai** (xét độc lập với while ngoài)

Gọi βi là số lần lặp của đoạn mã **Bi** (xét độc lập với while ngoài)

Sosánh(n) =

Gán(n) =

T(n) =

\* Tính

Điều kiện thực hiện vòng lặp Ai là:

Khi đó αi bằng số con j với j = n - i → 2\*i ; j = j + 2

\* Tính

βi bằng số con k với k = i → 1 ; k = k / 2

βi bằng số phần tử của tập

với

Sosánh(n) =

với

Gán(n) =

với

T(n) =

với

**c)**

i = 1; ret = 0; s = 0; (3 g)

while ( i ≤ n) (n+1 ss)

{ j = 1 ; (2n g)  
 s = s+1/i; // {số thực}

while (j ≤ s ) (αi+1 ss)

{ ret = ret + i\*j; (αi g)

j = j + 1; (αi g)

}

i = i + 1; (n g)

}

Số lần lặp của vòng while ngoài là số con i với i = 1 → n

🡺 while ngoài có n lần lặp 🡪 n + 1 phép so sánh

Gọi αi là số lần lặp của vòng while trong (độc lập với while ngoài)

Gán(n) =

Sosánh(n) =

T(n) = Gán(n) + Sosánh(n) =

\* Tính αi

Ta có s = s + 1/i với i = 1 → n

αi bằng số con j với j = 1 →

Ta có với (hằng số Euler)

Gán(n) =

Sosánh(n) =

T(n) = Gán(n) + Sosánh(n) =

**d)**

i = 1; count = 0; (2 g)

while (i ≤ 3\*n) (3n+1 ss)

{ x = 2\*n – i; y = i – n; j = 1; (9n g)

while (j ≤ x) (xi+1 ss)

{ if (j ≥ n) (xi ss)

count = count – 1; **(a)** (αi g)

j = j + 1; (xi g)

} **Pi**

if (y > 0) (3n ss)

if (x > 0) (2n ss)

count = count + 1; **(b)** (βi g)

i = i + 1; (3n g)

}

Số lần lặp của vòng while ngoài là số con i với i = 1 → 3n

🡺 while ngoài có n lần lặp 🡪 3n + 1 phép so sánh

Gọi αi là số lần phép gán dòng **(a)** được thực hiện

Gọi βi là số lần phép gán dòng **(b)** được thực hiện

Gọi xi là số lần lặp đoạn mã **Pi** (xét độc lập với while ngoài)

Gán(n) =

Sosánh(n) =

T(n) = Gán(n) + Sosánh(n) =

Điều kiện thực hiện đoạn mã **Pi** là: 2n – i ≥ 1 → i ≤ 2n – 1

\*Ta có xi = 2n – i

\*Tính αi

Điều kiện thực hiện phép gán **(a)** là j ≥ n với j ≤ 2n – i ↔ n ≤ 2n – i ↔ i ≤ n

khi đó αi bằng số con j với j chạy từ n → 2n – i , tức αi = 2n – i – n = n – i

\*Tính βi

Phép so sánh ***if (y > 0)*** được thực hiện ở mỗi vòng while ngoài nên có 3n phép ss

Điều kiện thực hiện phép ss *if* ***(x > 0)*** là i – n > 0 với i = 1→3n

↔ n < i ≤ 3n , do đó số phép ss (x > 0) là 2n phép

Điều kiện thực hiện phép gán **(b)** là 2n – i > 0 với n < i ≤ 3n

↔ n < i < 2n

🡪 βi = 2n – n – 1 = n – 1

Gán(n)

Sosánh(n) =

T(n) = Gán(n) + Sosánh(n) =

**e)**

i = 1; count =0; (2 g)

while ( i ≤ 5n) (5n+1 ss)

{ x=(n-i)(i-3n) ;

y=i-2n; (15n g)

j=1;

while (j ≤ x ) (xi+1 ss)

{ if ( i>=2y ) (xi ss)

count = count - 2; **(a)** (αi g)

j = j + 1; (xi g)

} **Pi**  
 if (y>0) (5n ss)

count = count +1; **(b)** (βi g)

i = i + 1; (5n g)

}

Số lần lặp của vòng while ngoài là số con i với i = 1 → 5n

🡺 while ngoài có n lần lặp 🡪 5n + 1 phép so sánh

Gọi αi là số lần phép gán dòng **(a)** được thực hiện

Gọi βi là số lần phép gán dòng **(b)** được thực hiện

Gọi xi là số lần lặp đoạn mã **Pi** (xét độc lập với while ngoài)

xi = (n – i)(i – 3n)

Gán(n) =

Sosánh(n) =

T(n) = Gán(n) + Sosánh(n) =

Đoạn mã **Pi** chỉ được thực hiện khi x > 0

\*Tính αi

αi bằng số lần lặp của đoạn mã Pi mà thỏa điều kiện i ≥ 2y ↔ i ≥ 2(i – 2n) ↔ i ≤ 4n

Lập bảng xét dấu:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 1 |  | n |  | 3n |  | 4n |  | 5n |
| x=(n-i)(i-3n) |  | – | 0 | + | 0 |  | – |  |  |

Ta thấy điều kiện thực hiện Pi cũng thỏa điều kiện thực hiện phép gán **(a)**

Suy ra αi = xi

\*Tính βi

Phép so sánh ***if (y > 0)*** được thực hiện ở mỗi vòng while ngoài nên có 5n phép ss

Điều kiện thực hiện phép gán **(b)** là i – 2n > 0 ↔ i > 2n với i = 1 → 5n

Suy ra βi = 5n – 2n = 3n

Gán(n) =

Sosánh(n) =

T(n) = Gán(n) + Sosánh(n)

**f)**

s = 0; i = 1; (2 g)

while ( i ≤ n) (n+1 ss)

{ j=i; (n g)

while (j ≤ i2 ) (αi +1 ss)

{ if ( j ≥ ni/2 ) (αi ss)

s = s+j; (βi g)

j = j + 1; (αi g)

}

i = i + 1; (n g)

}

Số lần lặp của vòng while ngoài là số con i với i = 1 → n

🡺 while ngoài có n lần lặp 🡪 n + 1 phép so sánh

Gọi αi là số lần lặp của vòng while trong (độc lập với while ngoài)

Điều kiện thực hiện vòng while trong là: j ≤ i2 ↔ i(i – 1) ≥ 0 luôn đúng do i = 1 → n

Khi đó αi bằng số con j với j = i → i2 🡪 αi = i2 – i + 1

Gọi βi là số vòng while trong mà thỏa mãn điều kiện ***(j ≥ ni/2)*** với j ≤ i2

⬄ i2 ≥ ni/2 ⬄ 2i2 – ni ≥ 0 ⬄ i(2i – n) ≥ 0 ⬄ 2i – n ≥ 0 (do i > 0) ⬄ i ≥ n/2

Gán(n)

Sosánh(n) =

T(n) = Gán(n) + Sosánh(n)

**1.2- Viết chương trình thực nghiệm để đếm số phép gán và số phép so sánh ở các câu trên, qua đó kiểm tra lại phần đếm lý thuyết.** (các file .cpp đính kèm)

**Câu 2:**

|  |  |
| --- | --- |
| **Thuật toán 1** | **Thuật toán 2** |
| i = 1; m = n – n\*(n mod 2);  s = 3; (3 g)  while (i ≤ m) (m+1 ss)  { j = 1;  k = i\*i\*(i mod 2); (3m g)  s = s +3;  while ( j ≤ k) (αi+1 ss)  { j = j +1;  s = s+2; (αi g)  }  i = i +1; s = s+2; (2m g)  } | i = 1; m = n\*(n mod 2);  s = 3; (3 g)  while (i ≤ m) (m+1 ss)  { j = 1;  k = i\*i - i\*i\*(i mod 2); (3m g)  s = s +3;  while ( j ≤ k) (αi+1 ss)  { j = j +1;  s = s+2; (αi g)  }  i = i +1; s = s+2; (2m g)  } |
| **a) Tính theo n số phép gán v số phép so sánh trong mỗi thuật toán** | |
| Nhận xét:  - Với n lẻ ⬄ n%2=1 thì m = n – n = 0  Khi đó G1(n) = 3; SS1(n) = 1 | Nhận xét:  - Với n chẵn ⬄ n%2=0 thì m = 0  Khi đó G2(n) = 3; SS2(n) = 1 |
| - Với n chẵn ⬄ n%2=0 thì m = n – 0 = n  Số lần lặp của vòng while ngoài là số con i với i = 1 → m (m=n)  🡺 while ngoài có n lần lặp 🡪 n + 1 phép so sánh  Gọi αi là số lần lặp của vòng while trong (độc lập với while ngoài)  Điều kiện thực hiện của while trong là:  k > 0 ⬄ i2(i%2) > 0 ⬄ ⬄ i lẻ  αi bằng số con j với j = 1 → k (với k = i2)  🡪 αi = i2 khi i lẻ và αi = 0 khi i chẵn  G1(n)  SS1(n) | - Với n lẻ ⬄ n%2=1 thì m = n  Số lần lặp của vòng while ngoài là số con i với i = 1 → m (m=n)  🡺 while ngoài có n lần lặp 🡪 n + 1 phép so sánh  Gọi αi là số lần lặp của vòng while trong (độc lập với while ngoài)  Điều kiện thực hiện của while trong là:  k > 0⬄i2 – i2(i%2) > 0⬄⬄i chẵn  αi bằng số con j với j = 1 → k (với k = i2)  🡪 αi = i2 khi i chẵn và αi = 0 khi i lẻ  G2(n)  SS2(n) |
| **b) Suy ra độ phức tạp của mỗi thuật toán** | |
| T1(n) = G1(n) + SS1(n) | T2(n) = G2(n) + SS2(n) |

**c) Có thể so sánh độ phức tạp của 2 thuật toán trên hay không**

Độ phức tạp của 2 thuật toán trên là như nhau vì đều thuộc

**d) Cho biết ý nghĩa của biến s trong mỗi thuật toán**

Trong cả 2 thuật toán trên, biến s đều có vai trò như 1 biến đếm số phép gán

Giá trị ban đầu của s bằng 3 ứng với thao tác khởi tạo các biến i, m, s

Trong các vòng lặp, cứ mỗi phép gán xuất hiện thì giá trị của s đều tăng số đơn vị tương ứng (tính cả phép gán vào s)

**e) Không dùng các công thức đã tính ở câu a, viết chương trình để tính trực tiếp giá trị của các hàm G1(n), G2(n) với n = 1, 2, …, 20. Sau đó in ra dạng bảng để so sánh kết quả với công thức đã có nhờ câu a.**

(file source code đính kèm)

Theo phân tích ở câu d, giá trị cuối cùng của s ở 2 thuật là giá trị tương ứng của G(n)

**f) Với mỗi n, gọi p(n) là số nguyên tố thứ n (nghĩa là p(1) = 2, p(2) = 3, p(3) = 5… ). Gọi K(n) là thuật toán có được từ thuật toán 1 bằng cách thay giá trị của n bởi p(n). Hãy xác định số phép gán và số phép so sánh được thực hiện bới thuật toán K và suy ra độ phức tạp của thuật toán này.**

Thuật toán K:

- Xét n=1; p(1) = 2

K(1) được xác định bằng cách thay giá trị của n bởi 2:

i = 1; m = 2;

s = 3; (3 g)

while (i ≤ m) (3 ss) // i = 1, 2, 3

{ j = 1; (6 g) // vòng while ngoài lặp 2 lần khi i = 1, 2

k = i\*i\*(i mod 2); // k = 1 khi i = 1, k=0 khi i = 2

s = s +3;

while ( j ≤ k) (3 ss) // 2ss khi k = 1 và 1 ss khi k = 0

{ j = j +1;

s = s+2; (2 g) // vòng while chỉ lặp 1 lần khi i=1, k=1

}

i = i +1; s = s+2; (4 g)

}

Vậy khi n = 1 thì số phép gán: Gk(1)=15; số phép so sánh SSk(1)=6

- Xét n > 1 ⬄ p(n) lẻ (số nguyên tố lớn hơn 2) ⬄ p(n)%2 ⬄ p(n) – p(n)\*[p(n)%2] = 0

K(n) được xác định bằng cách thay giá trị của n bởi p(n):

i = 1; m = 0;

s = 3; (3 g)

while (i ≤ m) (1 ss) // i > m, vòng while không thực hiện

{ j = 1;

k = i\*i\*(i mod 2);

s = s +3;

while ( j ≤ k)

{ j = j +1;

s = s+2;

}

i = i +1; s = s+2;

}

Vậy khi n > 1 thì số phép gán: Gk(n)=3; số phép so sánh SSk(n)=1

- Vậy độ phức tạp của thuật toán K là

**g) Gọi H(n) là thuật toán có được từ thuật toán 1 bằng cách thay giá trị của n bởi n(n+1). Hãy xác định số phép gán và số phép so sánh được thực hiện bới thuật toán H và suy ra độ phức tạp của H.**

- Nhận xét: n(n+1) là số chẵn với mọi n nguyên, tức n(n+1) mod 2 = 0

H(n) được xác định từ thuật toán 1 bằng cách thay n bởi n(n+1):

i = 1; m = n(n+1);

s = 3; (3 g)

while (i ≤ m) (m+1 ss)

{ j = 1;

k = i\*i\*(i mod 2); (3m g)

s = s +3;

while ( j ≤ k) (αi+1 ss)

{ j = j +1;

s = s+2; (αi g)

}

i = i +1; s = s+2; (2m g)

}

Số lần lặp của vòng while ngoài là số con i với i = 1 → m với m = n(n+1)

🡺 while ngoài có n lần lặp 🡪 m + 1 phép so sánh

Gọi αi là số lần lặp của vòng while trong (độc lập với while ngoài)

Điều kiện thực hiện của while trong là:

k > 0 ⬄ i2(i%2) > 0 ⬄ ⬄ i lẻ

αi bằng số con j với j = 1 → k (với k = i2)

🡪 αi = i2 khi i lẻ và αi = 0 khi i chẵn

Gh(n)

với m = n(n+1)

SSh(n)

với m = n(n+1)

Suy ra độ phức tạp:

Th(n) = Gh(n) + SSh(n)