TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

BÀI TẬP MÔN PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN

**HOMEWORK #03: Độ phức tạp và các ký hiệu tiệm cận tới hạn**

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

                    Nhóm thực hiện:

                            1. Đoàn Thanh Tùng 21521646

                            2. Đàm thành Nam 21522354

3. Lê Phan Hiển 21520839

4. Lê Khai Trí 21521565

TP.HCM, ngày 16 tháng 11 năm 2023

# Bài 1:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 second | 1 minute | 1 hour | 1 day | 1 month | 1 year | 1 century |
| lg n | 10^10^6 | 10^  (6\*10^7) | 10^  (36\*10^8) | 10^  (25920\*10^8) | 10^  (25920\*10^8) | 10^  (315360\*10^8) | 10^  (31556736\*10^8) |
|  | 10^12 | 36\*10^14 | 1296\*10^16 | 746496\*10^16 | 6718464\*10^18 | 994519296\*10^18 | 995827586973696\*10^16 |
| n | 1 \* 10^6 | 6 \* 10^7 | 36 \* 10^8 | 864 \* 10^8 | 2592 \* 10^9 | 31104 \* 10^9 | 31104 \* 10^11 |
| n lg n | 62746 | 2801417 | 133378058 | 2755147513 | 71870856404 | 797633893349 | 68654697441062 |
| n2 | 1000 | 7745 | 60000 | 293938 | 1609968 | 5615692 | 56175382 |
| n3 | 100 | 391 | 1532 | 4420 | 13736 | 31593 | 146677 |
| 2n | 19 | 25 | 31 | 36 | 41 | 44 | 51 |
| n! | 9 | 11 | 12 | 13 | 15 | 16 | 17 |

A screenshot of a computer

Description automatically generated

Code python:

import math

# Hàm chuyển đổi các đơn vị thời gian về giây

def convert\_to\_seconds(value, unit):

    if unit == 'second' or unit == 'seconds':

        return value

    elif unit == 'minute' or unit == 'minutes':

        return value \* 60

    elif unit == 'hour' or unit == 'hours':

        return value \* 3600

    elif unit == 'day' or unit == 'days':

        return value \* 86400

    elif unit == 'month' or unit == 'months':

        return value \* 30 \* 86400

    elif unit == 'year' or unit == 'years':

        return value \* 365 \* 86400

    elif unit == 'century' or unit == 'centuries':

        return value \* 100 \* 365 \* 86400

    else:

        return None

# Hàm tìm giá trị nguyên dương lớn nhất của n mà lgn \* 10^-6 <= 1

def largest\_n(value, unit):

    seconds = convert\_to\_seconds(value, unit)

    if seconds is None:

        return "Đơn vị không hợp lệ!"

    n = 1

    while True:

        # thay n^2 thành các hàm khác tại đây

        if (n\*\*2) \* 10\*\*-6 <= seconds:

            n += 1

        else:

            return n - 1

# Hàm biểu diễn giá trị dưới dạng a\*10^b

def format\_result(number):

    if number == 0:

        return "0"

    # nếu number không chia hết cho 10

    if(number % 10 != 0):

        return f"{number:.0f}"

    exponent = int(math.log10(number))

    coefficient = number / (10 \*\* exponent)

    return f"{coefficient:.0f} \* 10^{exponent}"

# Nhập giá trị và đơn vị thời gian từ người dùng

value = 1

unit = input("Nhập đơn vị thời gian (second/minute/hour/day/month/year/century): ")

# Tìm giá trị nguyên dương lớn nhất của n thỏa mãn điều kiện

result = largest\_n(value, unit)

# Biểu diễn kết quả dưới dạng a\*10^b

formatted\_result = format\_result(result)

print(f"Giá trị nguyên dương lớn nhất của n là: {formatted\_result}")

# Bài 2

1. Trong kí hiệu: T(n) = O(f(n)), dấu = chỉ là kí hiệu hình thức.

Bản chất: T(n) ∈ O(f(n)).

A blue rectangular sign with white numbers and symbols

Description automatically generated

Vì vậy, phép suy ra trên là sai

A white background with blue text

Description automatically generated

* f(n) = O(g(n))
* Ta cần chứng minh ∃c ∈ **R**+ và ∃n0 ∈ **N** sao cho:

7n2 ≤ c(n2 – 80n) ∀n ≥ n0

* Chọn c = 8 => - n2 + 640n ≤ 0
* Bảng xét dấu:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n |  | 0 |  | 640 |  |
| - n2 + 640n | - | 0 | + | 0 | - |

Ta thấy với c = 8, n0 = 640 => - n2 + 640n ≤ 0, ∀n ≥ 640

* Điều cần chứng minh
* g(n) = O(f(n))
* Ta cần chứng minh ∃c ∈ **R** + và ∃n0 ∈ **N** sao cho:

n2 – 80n ≤ c(7n2) ∀n ≥ n0

* Chọn c = 3 => 20n2 + 80n ≥ 0

Bảng xét dấu:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n |  | -4 |  | 0 |  |
| 6n2 + 80n | + | 0 | - | 0 | + |

Ta thấy với c = 3, n0 = 0 => 20n2 + 80n ≥ 0, ∀n ≥ 0

* Điều cần chứng minh
* f(n) = O(h(n))
* Ta cần chứng minh ∃c ∈ **R**+ và ∃n0 ∈ **N** sao cho:

7n2 ≤ c(n3) ∀n ≥ n0

* Chọn c = 1 => n3 - 7n2 ≥ 0
* Xét z(n) = n3 - 7n2 = n2(n - 7)

Bảng xét dấu:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n |  |  | 0 |  | 7 |  |
| n2 | + | | 0 | + | | |
| n - 7 | - | | | | 0 | + |
| z(n) | - | | 0 | - | 0 | + |

Ta thấy với c = 1, n0 = 7 => n3 - 7n2 ≥ 0, ∀n ≥ 7

* Điều cần chứng minh
* h(n) ≠ O(f(n))
* Ta thử chứng minh: h(n) = O(f(n)), nghĩa là chứng minh: ∃c ∈ **R**+ và ∃n0 ∈ **N** sao cho:

n3 ≤ c(7n2) ⬄ n3 - c(7n2) ≤ 0 ∀n ≥ n0

* Xét z(n) = n3 - 7cn2 = n2(n – 7c)
* Bảng xét dấu:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n |  |  | 0 |  | 7c |  |
| n2 | + | | 0 | + | | |
| n – 7c | - | | | | 0 | + |
| z(n) | - | | 0 | - | 0 | + |

Ta thấy z(n) ≤ 0 khi n ≤ 7c (vì c  **R**+ nên 7c > 0)

* Không tồn tại n0 ∈ **N** sao cho z(n) ≤ 0 ∀n ≥ n0
* h(n) ≠ O(f(n))

c.

c.1. Chứng minh

Ta chứng minh

Ta có:

sao cho

Biểu thức trên luôn sai bởi vì c là một hằng số nên khi giá trị n đủ lớn thì n > c

Vậy ta có (điều phải chứng minh)

c.2. Chứng minh

Ta chứng minh

Ta có:

sao cho

Biểu thức trên luôn sai bởi bì c là một hằng số nên khi giá trị n đủ lớn thì

Vậy ta có (điều phải chứng minh)

c.3. Chứng minh

Ta chứng minh

khi và chỉ khi và

Giả sử

Xét 1 hàm bất kì

sao cho

Dựa trên các điều kiện trên ta chọn 1 hàm

Vì mà nên

sao cho

Biểu thức trên luôn sai bởi vì d là một hằng số nên khi giá trị n đủ lớn thì

* (điều phải chứng minh)

d.

sao cho

Ở đây ta chọn c là dãy số ta có

Biểu thức trên luôn đúng ta có điều cần chứng minh

# Bài 3:

Group 1:

Group 2:

Group 3:

Group 4:

Group 5:

Group 6:

Group 7:

# Bài 4:

**a/ O(C.f(n)) = O(1.f(n)) với C là hằng số**

* Chứng minh :

sao cho

* Chứng minh :

sao cho

Suy ra: O(C) = O(1)

**b/ Nếu f(n) O(g(n)) và g(n) O(h(n)) thì f(n) O(h(n))**

* Xét f(n) O(g(n)): có nghĩa là tồn tại sao cho: f(n) <= (với mọi n >= )
* Xét g(n) O(h(n)): có nghĩa là tồn tại sao cho: g(n) <= (với mọi n >= )
* f(n) <= , chọn thì f(n) <=

Vậy f(n) O(h(n))

**c/ Nếu**

* Xét t1(n) O(f(n)) có nghĩa là tồn tại sao cho: t1(n) <= (với mọi n >= )
* Xét t2(n) O(g(n)) có nghĩa là tồn tại sao cho: t2(n) <= (với mọi n >= )
* Chọn ta có
* Chọn ta có

Từ đây ta suy ra

d/

Để chứng minh ta cần chứng minh hai điều kiện là và

Chứng minh

Nếu có nghĩa là

sao cho

Chọn C bằng 2 ta có

Biểu thức ở trên luôn đúng nên

Chứng minh

Nếu có nghĩa là

sao cho

Chọn C bằng 1 ta có

Biểu thức trên luôn đúng nên

Kết luận: vì và

nên

**e/ Nếu t(n) O(g(n)) thì g(n) (t(n))**

* Xét t(n) O(g(n)), tồn tại C > 0, n0 > 0 thỏa mãn: t(n) <= C.g(n) (với mọi n >= n0)

 g(n) >= (vì C > 0 nên > 0)

* Chọn C1 = thì: g(n) >= C1t(n) (với mọi n >= n0)

Suy ra: g(n) (t(n))

**f/**

Ta có:Θ  
O(g(n)) = {f(n): C1 > 0, n0 > 0 thì có 0 <= f(n) <= C1g(n) với mọi n >= n0}

Mà

Suy ra

# Bài 5:

**a/ Nếu f(n) = và g(n) = , thì h(n) =**

* Xét f(n) =

 với 0 < C1 <

* Xét g(n) =

 với 0 < C2 <

Suy ra: mà 0 < <

Vậy h(n) =

**b/ Nếu f(n) = O(g(n)) và g(n) = O(h(n)) thì h(n) =**

* Xét f(n) = O(g(n))



* Xét g(n) = O(h(n))



Suy ra: mà và nên không xác định được.

Vậy khẳng định trên là sai

**c/ Nếu f(n) = O(g(n)) và g(n) = O(f(n)) thì f(n) = g(n)**

* Xét f(n) = O(g(n))



* Xét g(n) = O(f(n))



Mà và , không thể xác định được C1 có bằng C2 hay không

Vậy khẳng định trên là sai

**d/**

Ta có:

Vậy , khẳng định là đúng

**e/ f(n) + O(f(n)) =**

* Xét 



Mà

Vậy khẳng định là đúng

**f/**

Ta có



Vậy khẳng định là sai

**g/**

Ta có

Mà 

Vậy khẳng định trên là đúng

**h/**

Ta có

Mà 

Vậy khẳng định trên là đúng

# Bài 6: Ước lượng nhanh độ phức tạp của giải thuật đệ quy dùng Định lý Master

## T(n) = 3 T() +

Xác định: a = 3, b = 2, d = 2

Ta có = = 4 > 3 = a

* **Dạng đơn giản (1) - Case 1** được áp dụng
* T(n) () = ()

## T(n) = 7 T() +

Xác định: a = 7, b = 3, d = 2

Ta có = = 9 > 7 = a

* **Dạng đơn giản (1) - Case 1** được áp dụng
* T(n) () = ()

## T(n) = 3 T() +

Xác định: a = 3, b = 3, d = 1

Ta có = = 3 = 3 = a

* **Dạng đơn giản (1) - Case 2** được áp dụng
* T(n) () = ()

## T(n) = 16 T() + n

Xác định: a = 16, b = 4, d = 1

Ta có = = 4 < 16 = a

* **Dạng đơn giản (1) - Case 3** được áp dụng
* T(n) () = () = ()

## T(n) = 2 T() +

Xác định: a = 2, b = 4, d = 0.51

Ta có = 2.028 > 2 = a

* **Dạng đơn giản (1) - Case 1** được áp dụng
* T(n) () = ()

## T(n) = 3 T() + n

Xác định: a = 3, b = 2, d = 1

Ta có = = 2 < 3 = a

* **Dạng đơn giản (1) - Case 3** được áp dụng
* T(n) () = ()

## T(n) = 3 T() +

Xác định: a = 3, b = 3, d =

Ta có = = < 3 = a

* **Dạng đơn giản (1) - Case 3** được áp dụng
* T(n) () = () =(n)

## T(n) = 4 T() + cn

Xác định: a = 4, b = 2, d = 1

Ta có = = 2 < 4 = a

* **Dạng đơn giản (1) - Case 3** được áp dụng
* T(n) () = () = ()

## T(n) = 4 T() + 5n

Xác định: a = 4, b = 4, d = 1

Ta có = = 4 = a

* **Dạng đơn giản (1) - Case 2** được áp dụng

T(n) () = ()

## T(n) = 5 T() + 4n

Xác định: a = 5, b = 4, d = 1

Ta có = = 4 < 5 = a

* **Dạng đơn giản (1) - Case 3** được áp dụng
* T(n) () = ()

## T(n) = 4 T() + 5n

Xác định: a = 4, b = 5, d = 1

Ta có = = 5 > 4 = a

* **Dạng đơn giản (1) - Case 1** được áp dụng
* T(n) () = (n)

## T(n) = 25 T() +

Xác định: a = 25, b = 5, d = 2

Ta có = = 25 = a

* **Dạng đơn giản (1) - Case 2** được áp dụng
* T(n) () = ()

## T(n) = 10 T() +

Xác định: a = 10, b = 3, d = 1.2

Ta có = 3.73 < 10 = a

* **Dạng đơn giản (1) - Case 3** được áp dụng
* T(n) () = ()

## T(n) = 7 T() +

Xác định: a = 7, b = 2, d = 3

Ta có = = 8 > 7 = a

* **Dạng đơn giản (1) - Case 1** được áp dụng
* T(n) () = ()

## T(n) = 4 T() +

Xác định: a = 4, b = 2, f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Tuy nhiên:

Ta có

Ta có log n tăng chậm hơn so với bất cứ số dương n nào (CM ở trên mạng)

* log(n) = O(n) = O() với k > 0
* f(n) = O() với = 1
* f(n) = O() , với = 1
* **Dạng tổng quát (2) - Case 1** được áp dụng
* T(n) () = ()

## T(n) = 4 T() +

Xác định: a = 4, b = 5, f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Tuy nhiên:

Ta có

Ta có log n tăng chậm hơn so với bất cứ số dương n nào (CM ở trên mạng)

* log(n) = O(n) = O() với k > 0
* f(n) = O() với = 0.2
* f(n) = O() , với = 0.2
* **Dạng tổng quát (2) - Case 1** được áp dụng
* T(n) () = ()

## T(n) = T() +

Xác định: a = , b = 2, f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Tuy nhiên:

Ta có

=

Ta có log n tăng chậm hơn so với bất cứ số dương n nào (CM ở trên mạng)

* log(n) = O(n) = O() với k > 0
* f(n) = O() với = 0.2
* f(n) = O() , với = 0.2
* **Dạng tổng quát (2) - Case 1** được áp dụng
* T(n) () = ()

## T(n) = 2 T() +

Xác định: a =2, b = 3, f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Tuy nhiên:

Ta có

Ta có f(n) = () với = 0.4 vì =

Ta kiểm tra Regularity Condition:

a f() c f(n) với c < 1, n đủ lớn

 2 f() c f(n)

 2 () cn

 cn

Chọn c = < 1 ta có:

n (thỏa)

Vậy nếu ta chọn c = < 1 thì regularity condition sẽ được thỏa mãn

* **Dạng tổng quát (2) - Case 3** được áp dụng
* T(n) (f(n)) = (n )

## T(n) = 3 T() +

Xác định: a = 3, b = 4, f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Tuy nhiên:

Ta có

Ta có f(n) = () với = 0.2 vì =

Ta kiểm tra Regularity Condition:

a f() c f(n) với c < 1, n đủ lớn

 3 f() c f(n)

 3 () cn

 cn

Chọn c = < 1 ta có:

n (thỏa)

Vậy nếu ta chọn c = < 1 thì regularity condition sẽ được thỏa mãn

* **Dạng tổng quát (2) - Case 3** được áp dụng
* T(n) (f(n)) = (n )

## T(n) = 6 T() +

Xác định: a = 6, b = 3, f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Tuy nhiên:

Ta có

Ta có f(n) = () với = 0.4 vì =

Ta kiểm tra Regularity Condition:

a f() c f(n) với c < 1, n đủ lớn

 6 f() c f(n)

 6 () c

 c

Chọn c = < 1 ta có:

n (thỏa)

Vậy nếu ta chọn c = < 1 thì regularity condition sẽ được thỏa mãn

* **Dạng tổng quát (2) - Case 3** được áp dụng
* T(n) (f(n)) = ( )

## T(n) = 3 T() +

Xác định: a = 3, b = 5, f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Tuy nhiên:

Ta có

Ta có

Theo định lý L’Hospital, ta có:

=

* f(n) = = với > 0
* **Dạng tổng quát (2) - Case 1** được áp dụng
* T(n) () = ()

## T(n) = 2 T() +

Xác định: a = 2, b = 2, f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Tuy nhiên:

Ta có

Ta có f(n) = () với k = 1

* **Dạng tổng quát (2) - Case 2** được áp dụng
* T(n) () = () = ()

## T(n) = T() +

Vì a không phải hằng số => Không áp dụng được Master Theorem

## T(n) = 0.5 T() + n

Vì a < 1 => Không áp dụng được Master Theorem

## T(n) = T() +

Xác định: a = 1; b = 2; f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Ta có

=

Ta có f(n) = () với = 1 vì = nhỏ hơn

Ta kiểm tra Regularity Condition:

a f() c f(n) với c < 1, n đủ lớn

 1 f() c f(n)

 c ()

Ta xem n = 2 với k lẻ và đủ lớn

Với bất cứ sự lựa chọn nào của n thì ta cũng có c >=

* **Regularity Condition không thỏa**
* **Không áp dụng được Master Theorem**

## T(n) = 64 T() -

Vì f(n) = < 0 => Không áp dụng được Master Theorem

## T(n) = T() +

Xác định: a = 1, b = 2, f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Tuy nhiên:

Ta có

= 1

Ta có f(n) = () với = 1 vì = nhỏ hơn rất nhiều lần so với

Ta kiểm tra Regularity Condition:

a f() c f(n) với c < 1, n đủ lớn

 1 f() c f(n)

 c

Chọn c = < 1 ta có:

(thỏa)

Vậy nếu ta chọn c = < 1 thì regularity condition sẽ được thỏa mãn

* **Dạng tổng quát (2) - Case 3** được áp dụng
* T(n) (f(n)) = ()

## T(n) = 16 T() +

Xác định: a = 16, b = 4, f(n) = không phải là đa thức

* Không áp dụng Dạng đơn giản (1) được

Tuy nhiên:

Ta có

=

Ta có f(n) = () với = 1 vì = nhỏ hơn rất nhiều lần so với

Ta kiểm tra Regularity Condition:

a f() c f(n) với c < 1, n đủ lớn

 16 f() c f(n)

 16 c

Chọn c = < 1 ta có:

16 (thỏa)

Vậy nếu ta chọn c = < 1 thì regularity condition sẽ được thỏa mãn

* **Dạng tổng quát (2) - Case 3** được áp dụng
* T(n) (f(n)) = ()

# Bài 7: Chứng mình dùng giới hạn lim

|  |  |  |  |
| --- | --- | --- | --- |
| **f(n)** | **Các ký hiệu tiệm cận** | **G(n)** | **Giải thích** |
|  | f(n) = |  | Ta có f(n) = =  Áp dụng định lý L’Hospital, ta có: =   * TH3: 0 < * f(n) = |
|  | f(n) = |  | Ta có  = 0   * TH1: * f(n) = |
|  | f(n) = |  | Ta có  Áp dụng định lý L’Hospital, ta có: = = =   * TH5: * f(n) = |
|  | f(n) = |  | Ta có   * TH3: 0 <   f(n) = |
|  | f(n) = |  | Ta có   * TH3: 0 <   f(n) = |
|  | f(n) = |  | Ta có  Áp dụng định lý L’Hospital, ta có:  =   * TH3: 0 <   f(n) = |
|  | f(n) = |  | Ta có  = (Dùng tính chất của log)  =   * TH5:   f(n) = |
|  | f(n) = |  | Ta có  Áp dụng định lý L’Hospital, ta có: = = = =   * TH1:   f(n) = |
|  | f(n) = |  | Ta có   * TH3: 0 <   f(n) = |