**Bài tập 1:**

a,   
- Độ phức tạp của thuật toán là một khái niệm để đo lường lượng tài nguyên (thời gian và bộ nhớ) cần thiết để thực hiện thuật toán:  
 + Độ phức tạp thời gian: các thuật toán cần một lượng thời gian hữu hạn để thực thi, thời gian mà thuật toán cần để giải bài toán đã cho gọi là độ phức tạp thời gian của thuật toán.   
 + Độ phức tạp bộ nhớ: việc giải quyết các bài toán sử dụng máy tính yêu cầu bộ nhớ để lưu dữ liệu tạm thời hoặc kết quả cuối cùng trong khi thuật toán đang thực thi. Dung lượng bộ nhớ được yêu cầu bởi thuật toán để giải quyết bài toán được gọi là độ phức tạp bộ nhớ.

b,   
Em nghĩ ý kiến này là đúng, bởi vì:   
Trên thực tế, với các thuật toán có độ phức tạp cao, việc để tìm ra hàm T(n) chính xác là rất khó, và có thể gặp khó khăn khi so sánh 2 hàm số T(n) với nhau. Do đó, người ta lựa chọn phương pháp so sánh tương đối, chỉ quan têm đến những giá trị n lớn, n càng lớn (tiến tới vô cùng) 🡪 So sánh tốc độ tăng (tỷ lệ tăng trưởng) của 2 hàm vì hàm nào có tốc độ tăng nhanh hơn thì luôn lớn hơn ở vô cùng 🡪 Phân chia thành các bậc tăng trưởng “Order of growth”, hàm có bậc tăng trưởng lớn hơn thì tăng nhanh hơn (phức tạp hơn) 🡪 Dễ so sánh

c,  
Cách sử dụng các ký hiệu tiệm cận khi nói về độ phức tạp thuật toán:

* O (Big O): biểu diễn giới hạn trên của độ phức tạp thuật toán, nó cho biết thời gian chạy của thuật toán trong trường hợp tệ nhất đối với một input có kích thước n. Dễ dàng tìm thấy giới hạn trên của một thuật toán. Giới hạn trên.
* Omega (Big Omega): biểu diễn giới hạn dưới của thuật toán, nó cho biết thời gian chạy tốt nhất có thể của thuật toán đối với một input có kích thước n. Tuy nhiên nó lại không thực sự hữu ích và ít được sử dụng nhất trong cả 3. Giới hạn dưới.
* Theta (Big Theta): biểu diễn giới hạn một hàm từ trên xuống dưới, xác định hành vi tiệm cận chính xác. Nó cho biết thời gian chạy trung bình của thuật toán đối với một input có kích thước n.

**Bài tập 2:**

Yêu cầu của đề bài: Với mỗi hàm f(n) và thời gian t, chọn giá trị n lớn nhất của bài toán để có thể được giải quyết trong thời gian t, xem như thuật toán sẽ giải bài toán với thời gian f(n) µs.

Quy ước: 1 month = 30 days, 1 year = 365 days, lg(n) là log cơ số 2 của n

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 second | 1 minute | 1 hour | 1 day | 1 month | 1 year | 1 century |
| lg(n) | 2^(1e6) | 2^(6e7) | 2^(3.6e9) | 2^(8.64e10) | 2^(2.592e12) | 2^(3.1536e13) | 2^(3.1536e15) |
|  | 10^12 | 3.6e15 | 1.296e19 | 7.46496e21 | 6.718464e24 | 9.941409e26 | 9.941409e30 |
| n | 10^6 | 6e7 | 3.6e9 | 8.64e10 | 2.592e12 | 3.1536e13 | 3.1536e15 |
| nlg(n) | 62746 | 2801417 | 133378058 | 2755147513 | 71870856404 | 797633893349 | 68610956750570 |
|  | 1000 | 7745 | 60000 | 293938 | 1609968 | 5615692 | 56156922 |
|  | 100 | 391 | 1532 | 4420 | 13736 | 31593 | 146645 |
|  | 19 | 25 | 31 | 36 | 41 | 44 | 51 |
| n! | 9 | 11 | 12 | 13 | 15 | 16 | 17 |

Giải thích:

* Xét f(n) = n!, cần tìm giá trị nguyên của n lớn nhất sao cho n! <= t (thời gian)

Với t = 1 second = 10^6 microseconds, ta xét lần lượt các giá trị của n từ 1, bước tăng là 1:

Ta thấy tại n = 9: 9! = 362880 mà (10^6)/9! = 2.75 < 10 (số tiếp theo sau 9) nên 9 là giá trị lớn nhất có thể của n

Tương tự t = 1 minute = 6e7 microseconds, xét từ n = 9, bước tăng, tại n = 11: (6e7)/11! = 1.5 < 12 (số tiếp theo sau 11) nên 11 là giá trị lớn nhất có thể của n khi t = 1 minute

* Xét f(n) = 2^n, cần tìm giá trị nguyên của n lớn nhất sao cho 2^n <= t (thời gian)
* n\_max = []

Với t = 1 second = 10^6 microseconds, vậy n\_max = [] = 19

Với t = 1 minute = 6e7 microseconds, vậy n\_max = [] = 25

Với t = 1 hour = 3.6e9 microseconds, vậy n\_max = [] = 31

... tương tự với các giá trị khác của t

* Xét f(n) = n^3, cần tìm giá trị nguyên của n lớn nhất sao cho n^3 <= t (thời gian)
* n\_max = []

Với t = 1 second = 10^6 microseconds, vậy n\_max = [] = 100

Với t = 1 minute = 6e7 microseconds, vậy n\_max = [] = 391

Với t = 1 hour = 3.6e9 microseconds, vậy n\_max = [] = 1532

... tương tự với các giá trị khác của t

**Bài tập 3:**

a/ Phép suy ra này là sai, vì:  
Hai biểu thức này đều là hàm bậc 2, do đó chúng đều có bậc tăng trưởng cùng là O(n^2). Không thể kết luận là 2 biểu thức bằng nhau khi chúng có cùng bậc tăng trưởng.

b/

* n^3 O(n^2)

Giả sử n^3 O(n^2), nghĩa là tồn tại CR+ và n0 sao cho: n^3 <= C.n^2

Khi xét với n = 1, thì n^3 <= n^2 đúng  
Khi xét với n tiến đến vô cùng: n^3 <= C.n^2

* n <= C (mọi n >= n0) (không thể tìm được giá trị . nào của C thỏa mãn điều kiện, nếu n0>C thì bất đẳng thức sai )

Vậy giả sử là sai 🡪 n^3 O(n^2)

* n^4 + n + 1 O(n^2)

Giả sử n^4 + n + 1 O(n^2),

nghĩa là tồn tại CR+, n0 sao cho: n^4 + n + 1 <= C.n^2 ( >= n0)

* n^2 <= C ( >= n0) (không thể tìm được giá trị nào của C . thỏa mãn điều kiện, nếu n0 > C thì bất đẳng thức sai)

Vậy giả sử là sai 🡪 n^4 + n + 1 O(n^2)

* O(n^2) O(n)

Xét n^2 O(n^2), giả sử n^2 O(n), nghĩa là tồn tại C và n0 sao cho n^2 <= C.n ( >= n0)

⬄ n <= C ( >= n0) (không có giá trị C nào thỏa mãn điều kiện nếu, vì nếu . . n0 > C thì bất đẳng thức sai)

Vậy n^2 O(n) với >= n0 🡪 O(n^2) O(n)

c/ Giả sử tồn tại CR+, n0 sao cho:

T(n) = <= C.n^k ( >= n0, chọn n0 = 1)  
Mà <= =

🡪Chọn C = thì thỏa mãn T(n) = <= C.n^k ( >= n0)  
Vậy khi chọn C = và n0 = 1, thì ta có T(n) = O(n^k)

**Bài tập 4:**

Group 1:  
f1(n) =

f2(n) =

f3(n) = 1/n = O(1)

f4(n) =

f5(n) = nlog(n) = O

* f3 < f4 < f5 < f1 = f2 (O(1) luôn luôn là nhỏ nhất, > vì C+1 > 1, vì C rất nhỏ nên 1+C < 100)

Group 2:  
f1(n) =

f2(n) =

f3(n) =

f4(n) =

* f1 < f4 < f3 < f2 ( O(1) luôn là nhỏ nhất, < vì 3/2 < 2, và O(2^n) lớn nhất theo bậc tăng trưởng)

Group 3:   
f1(n) =

f2(n) =

f3(n) =

f4(n) =

* f4 < f1 < f3 < f2 ( Vì n^2 là nhỏ nhất, C + ½ < 1 do C rất nhỏ 🡪 , n/2 < n nên )

Group 4:   
f1(n) =

f2(n) =

f3(n) =

f4(n) =

f5(n) =

* f4 < f2 < f5 < f1 < f3 ( O(log n) là nhỏ nhất, 6 > 2 > 4C + ½ vì C rất nhỏ, C < log n 🡪 )

Group 5:  
f6(n) =

f7(n) =

f8(n) =

f9(n) =

f10(n) =

* f7 < f10 < f9 f6 < f8

Group 6:  
f1(n) =

f2(n) =

f3(n) =

f4(n) =

* f3 < f1 f2 < f4

Group 7:  
f1(n) = (n – 2)! = O(n!)

f2(n) =

f3(n) =

f4(n) =

f5(n) =

f6(n) =

f7(n) =

* f2 < f5 < f6 < f4 < f3 < f7 < f1

**Bài tập 5:**

**a/ O(C) = O(1) với C là hằng số**

Chứng minh :  
Chứng minh :   
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/ max{f(n), g(n)} =**

**d/ 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))

**e/**

Xét hàm f(n) , f(n) có dạng f(n) = với c > 0  
Vậy f(n) ⬄ 🡪 (1)

Xét hàm h(n) , h(n) có dạng h(n) = cg(n) với c > 0  
Vậy h(n) ⬄ 🡪 (2)

Từ (1) và (2) suy ra:

**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 tập 6:**

**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 tập 7:**

**Text

Description automatically generated**

**Text

Description automatically generated with medium confidence**

**1/ T(n) = 3T(n/2) + n^2**

Có

a = 3, b = 2 và

Và af(n/b) <= cf(n) (vì )

Vậy T(n) = (case 3 dạng tổng quát)

**2/ T(n) = 7T(n/3) + n^2**

Có

a = 7, b = 3 và

Và af(n/b) <= cf(n) (vì )

Vậy T(n) = (case 3 dạng tổng quát)

**3/ T(n) = 3T(n/3) + n/2**

a = 3, b = 3 và >= 0

Vậy T(n) = (case 2 dạng tổng quát)

**4/ T(n) = 16T(n/4) + n**

a = 16, b = 4 và f(n) = n

mà a >

Vậy T(n) = (case 3 dạng đơn giản)

**5/ T(n) = 2T(n/4) + n^0.51**

Có f(n) = n^0.51  
a = 2, b = 4 và

f(n) = n^0.51 =

mà

Vậy T(n) = (case 1 dạng đơn giản)

**6/ T(n) = 3T(n/2) + n**

Có a =3, b = 2 và f(n) = n

Mà   
Vậy T(n) = (case 3 dạng đơn giản)

**7/ T(n) = 3T(n/3) +**

Có a = 3, b = 3 và f(n) = với d = 0.5 >= 0

Mà a >

Vậy (case 3 dạng đơn giản)

**8/ T(n) = 4T(n/2) + cn**

Có a = 4, b = 2 và f(n) = cn với d = 1 > 0

Mà a >

Vậy T(n) = (case 3 dạng đơn giản)

**9/ T(n) = 4T(n/4) + 5n**

Có a = 4, b = 4 và f(n) = 5n

Mà a =

Vậy T(n) = (case 2 dạng đơn giản)

**10/ T(n) = 5T(n/4) + 4n**

Có a = 5, b = 4 và f(n) = 4n

Mà a >

Vậy T(n) = (case 3 dạng đơn giản)

**11/ T(n) = 4T(n/5) + 5n**

Có a = 4, b = 5 và f(n) = 5n

Mà a <

Vậy T(n) = (case 1 dạng đơn giản)

**12/ T(n) = 25T(n/5) + n^2**

Có a = 25, b = 5 và f(n) =

Mà

Vậy T(n) = (case 2 dạng đơn giản)

**13/ T(n) = 10T(n/3) + 17n^1.2**

Có a = 10, b = 3 và f(n) =

Mà

Vậy T(n) = (case 3 dạng đơn giản)

**14/ T(n) = 7T(n/2) + n^3**

Có a = 7, b = 2 và f(n) =

Mà

Vậy T(n) = (case 1 dạng đơn giản)

**15/ T(n) = 4T(n/2) + logn**

Ta có f(n) = logn, không phải là một đa thức nên không thể áp dụng dạng 1

Có a = 4, b = 2 và f(n) = logn =

Vậy T(n) = (case 1 dạng tổng quát)

**16/ T(n) = 4T(n/5) + logn**

Ta có f(n) = logn, không phải là một đa thức nên không thể áp dụng dạng 1

Có a = 4, b = 5 và f(n) = logn

Vậy T(n) = (case 1 dạng tổng quát)

**17/ T(n) =**

Ta có f(n) = logn, không phải là một đa thức nên không thể áp dụng dạng 1

Có a = , b = 2 và f(n) = logn

Vậy T(n) = (case 1 dạng tổng quát)

**18/** **T(n) =**

Ta có f(n) = nlogn, không phải là một đa thức nên không thể áp dụng dạng 1

Có a = 2, b = 3 và f(n) = nlogn với k = 1 >= 0

Vậy T(n) = (case 2 dạng tổng quát)

**19/** **T(n) =**

Ta có f(n) = nlogn, không phải là một đa thức nên không thể áp dụng dạng 1

Có a = 3, b = 4 và f(n) = nlogn với k = 1 >= 0

Vậy T(n) = (case 2 dạng tổng quát)

**20/** **T(n) =**

Ta có f(n) = , không phải là một đa thức nên không thể áp dụng dạng 1

Có a = 6, b = 3 và f(n) = với k = 1 >= 0

Vậy T(n) = (case 2 dạng tổng quát)

**18 19 20 lú quá, sai sai sao á**

**21/ T(n) =**

Có a = 3, b = 5 và không phải là một đa thức

f(n) =

Vậy T(N) = (case 1 dạng tổng quát)

**22/ T(n) =**

Có a = 2, b = 2 và không phải là một đa thức

f(n) =

Vậy T(N) = (case 1 dạng tổng quát)

**23/ T(n) =**

Vì a = không phải là hằng số nên không áp dụng được định lý Master

**24/ T(n) = 0.5T(n/2) + n**

Vì a = 0.5 < 1 nên không thể áp dụng định lý Master

**25/ T(n) = T(n/2) + n(2 - cos(n))**

Ta có f(n) = n(2-cos(n)) không đơn điệu nên không thể áp dụng dạng đơn giản

Có a = 1, b = 2 và f(n) = n(2 – cos(n))

Xét điều kiện af(n/b) <= cf(n) ta thấy không có giá trị c < 1 nào thỏa mãn

Vậy không thể giải được theo định lý Master

**26/ T(n) = 64T(n/8) -**

Ta có f(n) = - không phải là một đa thức nên không thể áp dụng được dạng đơn giản

Có a = 64, b = 8 và f(n) = - < 0 nên không thể áp dụng định lý Master

**27/ T(n) = T(n/2) +**

Ta có f(n) = không phải là một đa thức nên không thể áp dụng dạng đơn giản

Có a = 1, b = 2 và f(n) = =

Và af(n/b) <= cf(n) () không có giá trị c < 1 nào thỏa mãn

Không thể áp dụng định lý Master

**28/ T(n) = 16T(n/4) + n!**

Do f(n) = n! không phải là một đa thức nên không thể áp dụng dạng đơn giản

Có a = 16, b = 4 và f(n) = n! =

Mà xét af(n/b) <= cf(n) () có tồn tại c < 1 thỏa mãn khi n đủ lớn

Vậy T(n) = (case 3 dạng tổng quát)