**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**

---------- 🙢★🙠 ----------

****

**BÀI TẬP MÔN PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN**

**[HW02.a] Ký hiệu tiệm cận Big-O**

***Giảng viên hướng dẫn:*** ThS. Huỳnh Thị Thanh Thương

***Nhóm sinh viên:***

1. Phan Thanh Hải 18520705

**TP. HỒ CHÍ MINH, 24/09/2019**

**BÀI 1**

1. Hãy cho biết ý nghĩa của "độ phức tạp" khi đề cập đến thuật toán?
2. Hãy cho biết ý kiến của bạn về nhận định dưới đây và giải thích vì sao?

“Khi nghiên cứu về các thuật toán, người ta quan tâm đặc biệt đến tính hiệu quả về thời gian của chúng nhưng thường là quan tâm đến bậc tăng trưởng (order of growth) của hàm thời gian thực hiện của thuật toán, chứ không phải là bản thân thời gian thực hiện T(n)”.

**Bài làm**

1. Ý nghĩa của "độ phức tạp" khi đề cập đến thuật toán:

"Độ phức tạp" của thuật toán là thời gian thực hiện thuật toán đó, cụ thể đó là một hàm số với *n* là kích thước dữ liệu đầu vào. "Độ phức tạp" của thuật toán thể hiện sự hiệu quả của một thuật toán (cùng 2 thuật toán khác nhau thuật toán nào mang lại hiệu quả tốt hơn với cùng một kích thước dữ liệu đầu vào *n* hoặc theo sự phân bố ngẫu nhiên của dữ liệu nhập vào).

1. Hãy cho biết ý kiến của bạn về nhận định dưới đây và giải thích vì sao?

“Khi nghiên cứu về các thuật toán, người ta quan tâm đặc biệt đến tính hiệu quả về thời gian của chúng nhưng thường là quan tâm đến bậc tăng trưởng (order of growth) của hàm thời gian thực hiện của thuật toán, chứ không phải là bản thân thời gian thực hiện T(n)”.

Khi quan tâm đến tính hiệu quả về thời gian của một thuật toán, nếu chỉ xét đến bản thân thời gian thực hiện T(n) tuy đơn giản nhưng sẽ nảy sinh một số vấn đề sau:

Với kích thước dữ liệu đầu vào *n* khá nhỏ thì một trong 2 thuật toán có thời gian thực hiện T(n) ít hơn so với của thuật toán kia nhưng không chênh lệch bao nhiêu. Thực tế ta phải xử lý với kích thước dữ liệu đầu vào *n* khá lớn (trong thực tế thường gặp) thì việc xét bản thân thời gian thực hiện T(n) không đem lại hiệu quả cao, đặc biệt là nếu hàm T(n) là một hàm phức tạp, biến thiên liên tục. Do đó cần phải xét tới bậc tăng trưởng (order of growth) của hàm thời gian thực hiện của thuật toán.

**BÀI 2**

Tìm sao cho

**Bài làm**

Chọn *c* = 7,

Ta có:

Do đó

Vậy

Chọn *c* = 5,

Ta có:

Thật vậy, thì

Do đó

Vậy

Chọn *c* = 4,

Ta có:

Thật vậy, thì

Do đó

Vậy

Chọn *c* = ,

Ta có:

Do đó

Vậy

Chọn *c* = ,

Ta có:

Do đó

Vậy

Chọn

Ta có:

Do đó

Vậy

Chọn

Ta có:

Do đó

Vậy

Chọn

Ta có:

Thật vậy, thì

Do đó

Vậy

Chọn *c* = 25,

Ta có:

Thật vậy, thì

Do đó

Vậy

Chọn *c* = 4,

Ta có:

Thật vậy, thì

Do đó

Vậy

Chọn *c* = ,

Ta có:

Thật vậy, thì

Do đó

Vậy

Chọn *c* = ,

Ta có:

Thật vậy, thì

Do đó

Vậy

Chọn *c* = ,

Ta có:

Thật vậy, thì:

Mà:

(1)

Suy ra:

Do đó:

Vậy

Chọn *c* = ,

Ta có:

Thật vậy, thì:

Do đó

Vậy

Chọn *c* = ,

Ta có:

Thật vậy, thì

Do đó

Vậy

Chọn *c* = ,

Ta có:

Thật vậy, thì

Do đó

Vậy

**BÀI 5**

Với mỗi nhóm hàm bên dưới, hãy sắp xếp tăng dần theo bậc tăng trưởng Big-O:

**Group 1:** **Group 2:**

**Group 3:**

**Group 4:**

**Bài làm**

**Group 2:**

Chọn *c* = ,

Ta có:

Do đó hay (1)

Chọn *c* = 1,

Ta có:

Thật vậy, thì

Do đó hay (2)

Chọn *c* = 2,

Ta có:

Thật vậy, thì

Do đó hay (3)

Từ (1), (2) và (3) ta có thứ tự sắp xếp tăng dần theo bậc tăng trưởng Big-O là:

**Group 3:**

Chọn *c* = ,

Ta có:

Do đó hay (1)

Chọn *c* = 1,

Ta có:

Thật vậy, thì

Do đó hay (2)

Chọn *c* = 1,

Ta có:

Thật vậy, thì

Do đó hay (3)

Từ (1), (2) và (3) ta có thứ tự sắp xếp tăng dần theo bậc tăng trưởng Big-O là:

**Group 4:**

Chọn *c* = 100,

Ta có:

Thật vậy, thì

Do đó hay (1)

Chọn *c* = 2,

Ta có:

Thật vậy, thì

Do đó hay (2)

Chọn *c* =

Ta có:

Thật vậy, thì

Do đó hay (3)

Chọn *c* = 1000,

Ta có:

Do đó hay (4)

Chọn *c* = 1,

Ta có:

Thật vậy, thì

Do đó hay (5)

Chọn *c* = 1,

Ta có:

Do đó hay (6)

Từ (1), (2), (3), (4), (5) và (6) ta có thứ tự sắp xếp tăng dần theo bậc tăng trưởng Big-O là:

**BÀI 4**

Chứng minh:

**Bài làm**

Giả sử là đúng.

Như vậy thì:

(1)

Xét trường hợp thì:

thì ta thầy tồn tại trường hợp *n* > *c*.

Xét trường hợp thì:

thì ta thầy luôn luôn *n* > *c*.

Như vậy (1) vô lí, mâu thuẫn với giả thuyết.

Vậy

Ta chứng minh tồn tại ít nhất một hàm số bất kỳ sao cho nhưng .

Xét

Chọn *c* = 2,

Ta có:

Do đó

Vậy

Giả sử là đúng.

Như vậy thì:

(2)

Xét trường hợp thì:

thì ta thầy tồn tại trường hợp *n* > *c*.

Xét trường hợp thì:

thì ta thầy luôn luôn *n* > *c*.

Như vậy (2) vô lí, mâu thuẫn với giả thuyết.

Vậy

Do đó

**BÀI 5**

Chứng minh:

Nếu và thì

Nếu và thì

**Bài làm**

Ta chứng minh và

Để chứng minh thì ta chứng minh mọi hàm số đều

Ta có:

(1)

Đặt thì (1)

Do đó

Vậy

Ta có:

(2)

Đặt thì (2)

Do đó

Vậy

Vậy

Nếu và thì

Ta có:

Do đó:

Đặt thì (2)

Vậy

Nếu và thì

Ta có:

Do đó:

Đặt thì (2)

Do đó

**BÀI 6**

Cho và Chứng minh hoặc bác bỏ .

Chứng minh:

**Bài làm**

Cho và Chứng minh hoặc bác bỏ .

Chọn *c* = 1,

Ta có:

Thật vậy, thì

Do đó

Chứng minh:

Cho hàm số bất kỳ sao cho

Khi đó thì:

⇔

Mà ta có:

Do đó:

Đặt thì

Do đó

Vậy

**BÀI 7**

Chứng minh các tính chất sau:

và

và

**Bài làm**

Cho hàm số bất kỳ. Ta chứng minh với mọi thì .

Với thì:

Mà ta có:

Hay

Do đó:

Đặt thì

Do đó

Vậy