

(Sinh viên được sử dụng tài liệu giấy)

HỌ VÀ TÊN SV: MSSV: STT: PHÒNG THI:.....	<u>CÁN BỘ COI THI</u>
---	-----------------------

CÂU HỎI TỰ LUẬN

Câu 1 (3 điểm) (G1.) Ký hiệu tiệm cận

- a. Hãy cho biết ý kiến của bạn về nhận định sau: “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 $T(n)$ ”. (1 điểm)
- b. Chứng minh:
- Với mọi $k \geq 1, n^k \notin O(n^{k-1})$ (0.5 điểm)
 - $(n \log n - 3n) = \Omega(n \log n)$ (0.5 điểm)
- c. Hãy sắp xếp 07 hàm số bên dưới theo thứ tự tăng dần của “order of growth”, và có giải thích ngắn gọn cách thực hiện. Ký hiệu: log là log cơ số 2. (1 điểm)

$f_1(n) = n^2 \sqrt{n}$	$f_2(n) = n^{\log n}$
$f_3(n) = n^2 * (\log n)^4$	$f_4(n) = 4^{n^4}$
$f_5(n) = \sum_{i=1}^n (i^2)$	$f_6(n) = 5^n$
$f_7(n) = n^{n^{1/5}}$	

Câu 2 (2 điểm) (G2.) Phân tích thuật toán

Xét một hàm đệ quy như sau:

```
int findMax(int a[], int l, int r)
{
    if(l==r) return a[l];
    int m=(l+r)/2;
    if(a[m]<a[m+1])
        return findMax(a,m+1,r);
    else if(a[m]>a[m+1])
        return findMax(a,l,m);
    else return max(findMax(a,m+1,r), findMax(a,l,m));
}
```

Yêu cầu:

- Thành lập phương trình đệ quy (kèm giải thích ngắn gọn cách thành lập
- Giải phương trình đệ quy để xác định độ phức tạp của hàm trên dùng **phương pháp đoán nghiệm**
Gợi ý: Có thể dùng định lý Master để đoán một nghiệm $f(n)$ và dùng chứng minh quy nạp để chứng tỏ rằng $T(n) \leq f(n), \forall n$.

Câu 3 (3 điểm) (G3.) Thiết kế thuật toán

a. Bạn đang lên kế hoạch sắp xếp chỗ ngồi cho một đám cưới với danh sách khách mời V . Giả sử bạn được cung cấp một bảng tra cứu T , trong đó $T[u]$ với $u \in V$ là danh sách khách mời mà u biết. Nếu u biết v thì v cũng biết u . Bạn được yêu cầu sắp xếp chỗ ngồi sao cho bất kỳ vị khách nào ngồi tại một bàn đều "biết" những vị khách khác đang ngồi cùng bàn đó, hoặc là hai người đã biết nhau trước đó rồi hoặc là biết nhau thông qua một số khách khác ngồi cùng bàn. Ví dụ, nếu x biết y và y biết z thì x, y, z có thể ngồi cùng một bàn. Cho trước V và T , hãy thiết kế một thuật toán/thuật giải để trả về số lượng bàn tối thiểu cần thiết để đạt được yêu cầu này. Cho biết giải thuật đó thiết kế theo phương pháp nào? Tại sao?

b. Bây giờ giả sử **chỉ có hai bàn** và bạn được cung cấp một bảng tra cứu khác là S , trong đó $S[u]$ với $u \in V$ là danh sách những vị khách có quan hệ không tốt với u . Nếu v có quan hệ không tốt với u thì u có quan hệ không tốt với v . Mục tiêu của bạn là sắp xếp chỗ ngồi sao cho không có cặp khách ngồi cùng bàn nào có quan hệ không tốt với nhau. Ví dụ, có 5 vị khách $\{A, B, C, D, E\}$, trong đó những cặp khách có quan hệ không tốt với nhau là $\{AB, BC, CE, AE, AD\}$. Chúng ta có thể sắp xếp chỗ ngồi bằng cách để A, C ngồi ở một bàn và B, E, D ngồi ở bàn khác. Một ví dụ khác, cũng có 5 vị khách $\{A, B, C, D, E\}$, những cặp không thể ngồi chung với nhau là $\{AB, BC, CE, AD, DE\}$. Trường hợp này thì không thể sắp được bàn. Cho trước V và S , hãy thiết kế một thuật toán/thuật giải, trả về YES nếu bạn có thể sắp xếp được chỗ ngồi ở 2 bàn hoặc NO nếu không sắp xếp được.

Lưu ý, các giải thuật phải được trình bày dưới dạng mã giả và có chú thích cho người đọc dễ hiểu. Mô tả chi tiết cấu trúc dữ liệu được sử dụng.

SINH VIÊN CHỈ CHỌN LÀM 1 TRONG 2 CÂU SAU: 4.1 HOẶC 4.2

Câu 4.1. (2 điểm) (G 3.) Thiết kế thuật toán

Cho một chuỗi s không quá 100 ký tự, hãy tìm tất cả các cách chia chuỗi s thành các chuỗi con sao cho tất cả chuỗi con đều là chuỗi đối xứng. Các chuỗi con phải được xuất theo đúng thứ tự chúng xuất hiện trong chuỗi s . Thứ tự xuất các cách chia là không quan trọng.

Input	Output
aab	(aa b), (a a b)
bbaccb	(b b a c c b), (b b a c c b), (b b a c c b), (b b a c c b), (bb a c c b), (bb a c c b), (bb a c c b), (bb a c c b)

Yêu cầu : Trình bày ý tưởng và mã giả để giải bài toán trên bằng phương pháp “Quay lui (Backtracking)”. Mã giả có chú thích đầy đủ cho người đọc dễ hiểu.

Gợi ý: Có thể trình bày ý tưởng bằng cách lần lượt trả lời các câu hỏi sau: Phân tích cấu trúc lời giải $S = \langle S_1, S_2, \dots \rangle$, nghĩa là có bao nhiêu bước giải thành phần, mỗi thành phần S_i là gì? Để đạt được lời giải S , ta phải xây dựng dần các bước giải S_i . Để xây dựng S_i , có những khả năng nào mà S_i có thể chọn? Một khả năng j như thế nào là chấp nhận được cho S_i ? Cách thức lưu lại sự lựa chọn j cho S_i ? Cách ghi nhận trạng thái mới/trả lại trạng thái cũ cho bài toán nếu có. Khi nào lời giải đã hoàn thành? Cách xuất các lời giải cho bài toán?

Câu 4.2. (2 điểm) (G 3.) Thiết kế thuật toán

Một dãy số (y_1, y_2, \dots, y_n) được gọi là một dãy “gợn sóng” nếu mọi bộ ba liên tiếp y_i, y_{i+1}, y_{i+2} có trong dãy thỏa $y_i < y_{i+1} > y_{i+2}$ hoặc $y_i > y_{i+1} < y_{i+2}$.

Cho một dãy số nguyên dương có n phần tử $x = (x_1, x_2, \dots, x_n)$, nhiệm vụ của bạn là tìm một dãy con “gợn sóng” dài nhất của x . Dãy con của một dãy số là dãy có được sau khi loại bớt một hay một số phần tử, các phần tử khác giữ nguyên vị trí.

Yêu cầu:

- a. Trình bày ý tưởng và mã giả để giải bài toán trên bằng phương pháp “**Quy hoạch động (Dynamic Programming)**”.

Gợi ý: Có thể trình bày ý tưởng gồm các bước: 1) Phân tích đặc trưng “Optimal substructure”; 2) Xác định phương trình quy hoạch động; 3) Cách xây dựng bảng lưu trữ kết quả của các bài toán con; 4) Xây dựng lời giải của bài toán ban đầu;

- b. Minh họa cách giải cho trường hợp cụ thể sau: cho dãy x gồm 7 phần tử: $x_1 = 10, x_2 = 90, x_3 = 86, x_4 = 50, x_5 = 63, x_6 = 4, x_7 = 100$

-----HẾT-----

Duyệt đề Khoa/Bộ Môn

Giáo viên ra đề

Huỳnh Thị Thanh Thương

Bảng chuẩn đầu ra môn học CS112 – Phân tích và thiết kế thuật toán:

CĐRMH	Mô tả CĐRMH
G1.	Giải thích được các khái niệm cơ bản trong việc ước lượng, phân tích, đánh giá độ phức tạp thuật toán
G2.	Trình bày và giải thích được các kỹ thuật cơ bản để thiết kế thuật toán; Trình bày và giải thích được các kỹ thuật phân tích ước lượng độ phức tạp thuật toán
G3.	Áp dụng được các kỹ thuật trừu tượng hóa để phát biểu được các bài toán cơ bản trong khoa học máy tính; Áp dụng được các chiến lược thiết kế để xây dựng giải thuật cho các bài toán cơ bản trong khoa học máy tính; Áp dụng được các kỹ thuật ước lượng độ phức tạp để phân tích và đánh giá các thuật toán đã thiết kế