TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM

ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ 1 – Năm học 2024-2025

MÃ LƯU TRỮ (do phòng KT-ĐBCL ghi)
CK2425.1
CSC14007

Tên học phần: Nhập môn phân tích độ phức tạp thuật toán **Mã HP:** CSC14007

Thời gian làm bài: **105 phút** Ngày thi: **09/01/2025**

Ghi chú: *Sinh viên được phép sử dụng tài liệu giấy khi làm bài.*

Họ tên sinh viên: ... . **MSSV:** ..... **STT:** ....

Đề thi có 04 câu với 02 trang, tổng là 10 điểm.

Câu 1. (*2.5 điểm*).

1) Với số n nguyên dương, hãy ước lượng độ lớn của biểu thức T_n sau đây theo ký hiệu Θ bằng cách thích hợp:

$$T_n = (\log 1)^2 + (\log 2)^2 + (\log 3)^2 + \cdots + (\log n)^2.$$

Từ đó chứng minh rằng $T_n \in O(n\sqrt{n})$ và $T_n \in \Omega(n \log n)$.

2) Cho n nguyên dương, chứng minh rằng hai thuật toán sau đây thì có cùng độ phức tạp (SV có thể đếm số phép gán/so sánh hoặc dùng các lập luận thích hợp khác cũng đều được):

```
//algorithm 1
for(int i = 1; i*i <= n; i++){
    for(int j = 1; j*j*j <= n - i*i; j++){
        res += j;
    }
}
//algorithm 2
for(int i = 1; i*i*i <= n; i++){
    for(int j = 1; j*j <= n - i*i*i; j++){
        res += j;
    }
}
```

Câu 2. (*2.5 điểm*). Trong dãy các số nguyên phân biệt $a_1, a_2, \ldots, a_n$, ta gọi a_i là "phần tử cực tiểu bên trái" nếu như nó là số nhỏ nhất so với các số bên trái của nó, cụ thể là

$$a_i = \min\{a_1, a_2, a_3, \ldots, a_i\}.$$

Ví dụ: trong dãy 3, 2, 5, 1, 4, 6 thì các phần tử cực tiểu bên trái sẽ là 3, 2 và 1 vì

$$3 = \min\{3\},\ 2 = \min\{3,2\} \text{ và } 1 = \min\{3,2,5,1\}.$$

1) Hãy đề xuất HAI thuật toán khác nhau để đếm số phần tử cực tiểu bên trái của một dãy số nguyên gồm n phần tử phân biệt cho trước, có cài đặt cụ thể bằng C++ hoặc Python. Từ đó, đánh giá sơ lược độ phức tạp của từng thuật toán. *Chú ý:* nếu số phép gán/so sánh là không cố định thì SV nên đánh giá dựa trên trường hợp xấu nhất của dữ liệu đầu vào (worst case).

2) Hỏi có thể dùng đại lượng "số phần tử cực tiểu bên trái" (hoặc nói chung là các phần tử cực đại, cực tiểu bên trái/phải) để đánh giá cho độ hiệu quả của thuật toán sắp xếp nào trong các thuật toán sắp xếp đã biết? Cho một ví dụ cụ thể và phân tích.

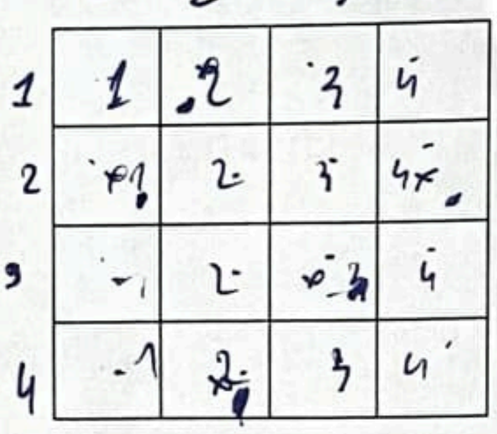
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM

ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ 1 – Năm học 2024-2025

MÃ LƯU TRỮ
(do phòng KT-ĐBCL ghi)

Câu 3. (*2.5 điểm*). Xét một thuật toán chia để trị **process**, thực hiện trên ma trận A kích thước $n\times n$ gồm các phần tử nguyên, có $T(n)$ là chi phí tính toán, và được mô tả như sau:

(i) Đầu tiên chia ma trận A thành 16 ma trận con đều nhau như hình bên (nếu số dòng/cột không chia hết cho 4 thì bổ sung một số hàng/cột gồm toàn 0 vào).



(ii) Coi mỗi ma trận con là một ô để đánh số 1, 2, 3, 4 cho các ô từ trái sang phải, từ trên xuống dưới và với các ô ở vị trí $(i;j)$ mà $i+j$ chia hết cho 3, thực hiện đệ quy **process** ở ngay ma trận con đó.

(iii) Thực hiện một thủ tục với chi phí $f(n)$ để gộp các kết quả tính toán được tại các ma trận con lại với nhau cho ra kết quả của hàm **process** ban đầu.

1) Viết công thức đánh giá độ lớn cho $T(n)$ ở dạng $T(n)=aT\left(\frac{n}{b}\right)+f(n)$ với a,b phù hợp.

2) Biết rằng chi phí $f(n)$ có thể là $f_1(n)=n\log n$ hoặc $f_2(n)=n$. Dùng định lý Master, chứng minh rằng trong cả hai trường hợp thì $\Theta(T(n))$ đều bằng nhau.

Hỏi kết quả sẽ thay đổi thế nào nếu chi phí cho $f(n)$ là $f_3(n)=n\sqrt{n}$?

Câu 4. (*2.0 điểm*). Xét thuật toán kiểm tra tính nguyên tố của số n nguyên dương như sau:

```
bool primality(int n){
    if(n < 2) return false;
    for(int i = 2; i*i <= n; i++){
        if(n % i == 0) return false;
    }
    return true;
}
```

1) Thực hiện kiểm tra độ phức tạp trung bình của thuật toán này theo các bước sau:

(i) SV chọn tuỳ ý một số LẺ $m\in[50;100]$ và xét 4 số liền trước và 4 số liền sau nó (tổng cộng sẽ có 9 số: $m-4,m-3,m-2,m-1,m,m+1,m+2,m+3,m+4$).

(ii) Lập bảng thống kê lại số lần lặp của vòng for trong thuật toán primality đã cho ứng với từng số trong danh sách ở (ii), chỉ cần đưa ra kết quả, không cần giải thích.

(iii) Tính số lần lặp trung bình và so sánh với $\sqrt{m}$.

2) Gọi $E(X_n)$ là kỳ vọng của biến ngẫu nhiên X mô tả số phép gán mà thuật toán cần dùng để kiểm tra tính nguyên tố của một số tuỳ ý thuộc $\{2,3,4,\ldots,n\}$ với n lớn. Ký hiệu $\alpha(n)$ là ước nguyên tố nhỏ nhất của số nguyên $n>1$ và cho biết rằng xét về độ phân bố của số nguyên tố thì sẽ có khoảng $\frac{n}{\log n}$ số nguyên tố không vượt quá n. Giải thích ngắn gọn cho

$$E(X_n)=\Theta\left(\frac{\sqrt{n}}{\log n}\right)+\frac{\sum_{x\text{ is not prime}}(\alpha(x)-1)}{n-1},$$

từ đó hãy đánh giá độ phức tạp trung bình của thuật toán đã nêu.