



Cau truc du lieu va giai thuat_ Nhóm 08

[Nhà của tôi](#) / [Các khoá học của tôi](#) / [DASA230179_21_1_08](#) / [Kiểm tra quá trình](#) / [Nhóm câu hỏi 1](#)

Bắt đầu vào lúc Saturday, 11 December 2021, 9:06 AM

Trạng thái Đã xong

Kết thúc lúc Saturday, 11 December 2021, 11:21 AM

Thời gian thực hiện 2 giờ 15 phút

Câu hỏi 1

Hoàn thành

Đạt điểm 1,00

Thời gian thực thi nhanh nhất của thuật toán Bubble Sort là

- ☐ a. $O(N*N)$
- ☒ b. $O(N)$
- ☐ c. Không có đáp án nào
- ☐ d. $O(N\log N)$

Câu hỏi 2

Hoàn thành

Đạt điểm 1,00

Thuật toán sắp xếp được gọi là stable nếu sau khi sắp xếp, thứ tự các giá trị bằng nhau không bị thay đổi.

Hãy chọn thuật toán sắp xếp KHÔNG stable

- ☐ a. Bubble Sort
- ☒ b. Quick Sort
- ☐ c. Merge Sort
- ☐ d. Insert Sort

Câu hỏi 3

Hoàn thành

Đạt điểm 1,00

Đánh giá thời gian thực thi của đoạn chương trình sau

```
int n, k, j;  
for (int i = 0; i < n; i = i + j) {  
    for (j = 0; j < k; j++) {  
        // some work  
    }  
}
```

- ☐ a. $O(N*N)$
- ☐ b. $O(N/K)$
- ☐ c. Không đánh giá được
- ☒ d. $O(N*K)$
- ☐ e. $O(N)$

Câu hỏi 4

Không trả lời

Đạt điểm 1,00

Bạn có rất nhiều công việc cần phải thực hiện trong ngày. Khi chọn 1 việc để làm thì bạn phải hoàn thành việc này trước khi thực hiện việc tiếp theo. Mỗi công việc có thời gian bắt đầu, thời gian kết thúc trong ngày (giá trị từ 0-24). Hãy viết hàm để tính số công việc nhiều nhất bạn có thể thực hiện trong ngày.

Input:

N: tổng công việc trong ngày

[start,end]: bắt đầu, kết thúc mỗi công việc

Ví dụ:

6

7 9

0 10

4 5

8 9

4 10

5 7

Output: 3 (số lượng công việc tối đa có thể làm)

Note: starts, ends là mảng số nguyên không âm có N phần tử tương ứng thời gian bắt đầu và kết thúc

starts[i], ends[i] thời gian bắt đầu và kết thúc công việc i

Expected Complexity $O(N\log N)$

Câu hỏi 5

Hoàn thành

Đạt điểm 1,00

Cho mảng các số nguyên có N phần tử chưa được sắp xếp, đếm số mảng con có tổng các phần tử bằng K

Sample Input

```
1 | arr = {10, 2, -2, -20, 10}
2 | K = -10
```

Sample Output

3

Explanation

 $10 + 2 - 2 + -20 = 10$ $-20 + 10 = -10$ $2 + -2 + -20 + 10 = -10$

Expected Complexity

Linear

```
#include<stdio.h>
int cntSubarrays(int arr[], int N, int K){
    //your code here

    int res = 0;

    for (int i = 0; i < N; i++)
    {
        int sum = 0;
        for (int j = i; j < N; j++)
        {
            sum += arr[j];
            if (sum == K)
                res++;
        }
    }

    return res;

    // end your code
}
```

Câu hỏi 6

Hoàn thành

Đạt điểm 1,00

Lớp cấu trúc dữ liệu và giải thuật có 1 triệu sinh viên. Điểm môn học được đánh giá theo thang điểm từ 0-100. Giảng viên muốn sắp xếp danh sách sinh viên theo điểm để chọn ra các bạn sinh viên có điểm cao nhất. Thuật toán sắp xếp nào được sử dụng là phù hợp

- ☒ a. Quick Sort
- ☐ b. Selection Sort
- ☐ c. Merge Sort
- ☐ d. Counting Sort

Câu hỏi 7

Không trả lời

Đạt điểm 1,00

Cho mảng có N phần tử. Hãy viết Hàm xuất vị trí bắt đầu và vị trí kết thúc của mảng con thỏa mãn điều kiện: nếu sắp xếp mảng con này theo thứ tự không giảm thì mảng lớn cũng theo thứ tự không giảm. Nếu mảng lớn đã có thứ tự thì xuất ra giá trị -1,-1

Sample Input

1 | [0, 2, 4, 7, 10, 11, 7, 12, 13, 14, 16, 19, 29]

Sample Output

1 | [4,6]

Câu hỏi 8

Hoàn thành

Đạt điểm 1,00

Thuật toán Quick Sort chỉ sắp xếp ngay trên mảng dữ liệu input, không dùng thêm bộ nhớ tạm để lưu trữ mảng. Vì vậy độ phức tạp về lưu trữ thuật toán Quick Sort là $O(1)$ là đúng hay sai?

Hãy chọn một:

☐ Đúng☒ Sai

Câu hỏi 9

Hoàn thành

Đạt điểm 1,00

Cho một M x N ma trận có row và column đã được sắp xếp theo thứ tự tăng dần, hãy viết hàm tìm vị trí phần tử K trong ma trận này. Output là 1 cặp giá trị {đòng,cột} tương ứng với vị trí của K. Nếu K không có trong ma trận thì xuất {-1,-1}

Input:

3 3

1 4 9

2 5 10

6 7 11

K = 10

Output 1,2

```
#include<stdio.h>
struct Output {
    int row, column;
}
Output search(int ** arr, int N, int M, int K){
    //your code here
Output temp;
    temp.column = -1;
    temp.row = -1;
    int i, j;
    for (i = 0; i < M; i++)
    {
        for (j = 0; j < N; j++)
        {
            if (arr[i][j] == K)
            {
                temp.column = j;
                temp.row = i;
                break;
            }
        }
    }
    if (temp.column != -1 && temp.row != -1)
        break;
}
return temp;
//end your code
}
```

Câu hỏi 10

Hoàn thành

Đạt điểm 1,00

Kết quả đoạn chương trình sau

```
float a = 3.14;
double b = 3.14;
if (a == b) {
    printf("Thuat toan");
}
else {
    printf("Du lieu");
}
```

- ☒ a. Du lieu
- ☐ b. Thuat toan

Câu hỏi 11

Hoàn thành

Đạt điểm 1,00

Thời gian thực thi chậm nhất của thuật toán tìm kiếm là

- ☒ a. $O(N)$
- ☐ b. $O(1)$
- ☐ c. $O(N*N)$
- ☐ d. $O(\log N)$

Câu hỏi 12

Hoàn thành

Đạt điểm 1,00

Hoàn thành hàm xuất tổng lớn nhất của mảng khi bắt đầu tính tổng từ phần tử đầu tiên. Tới vị trí index i , nếu tổng < 0 thì sẽ không cộng dồn phần trước đó.

Input

1 | {-1, 2, 3, 4, -2, 6, -8, 3}

Output

13

Expected Time Complexity

$O(N)$

Space Complexity

$O(1)$

Giải thích: đoạn {-1, 2, 3, 4, -2, 6, -8, 3} có tổng bằng 13:

sum = 0

Vị trí 0, sum = -1 < 0 \rightarrow sum = 0

Vị trí 1, 2, 3 sum = 2 + 3 + 4 = 9

Vị trí 4 sum = 9 + (-2) = 7 > 0 tổng lớn nhất là 9

Vị trí 5 sum = 7 + 6 = 13 tổng lớn nhất là 13

Vị trí 6 sum = 13 + (-8) = 5 tổng lớn nhất 13

Vị trí 7 sum = 5 + 3 = 8 tổng lớn nhất 13

```
#include<stdio.h>
int maxSubarraySum(int arr[], int N){
    //your code here
    int best = 0, sum = 0, i;
    for (i = 0; i < N; i++) {
        sum = sum + arr[i];
        if (sum > best)
        {
            best = sum;
        }
        if (sum < 0)
            sum = 0;
    }
    return best;
    //end your code
}
```

Câu hỏi 13

Hoàn thành

Đạt điểm 1,00

Bạn cần thực hiện chương trình tính giá trị nhiệt độ trung bình trong 7 ngày gần nhất. Cấu trúc dữ liệu nào có thể sử dụng

- ☐ a. Sử dụng Deque để lưu giá trị 7 ngày bất kỳ
- ☒ b. Sử dụng Queue để lưu 7 giá trị gần nhất
- ☒ c. Sử dụng Stack để lưu giá trị

◀ Video báo cáo tiến độ tuần 5[Chuyển tới...](#)**Nhóm câu hỏi 2 ▶**