

Nhóm 15:

- Bùi Văn Thuận - 20521990

-Nguyễn Thanh Phúc - 20521769

-Hoàng Gia Huy - 19521607

Phần I (5 điểm) - Tự luận

(Phần này gồm có 02 câu)

Câu 1 (2,5 điểm): Cho thuật toán tìm kiếm tuần tự như bên dưới. Xác định độ phức tạp của thuật toán này.

```
int search(int x,int[a],int n){
    int i ;
    int found ;
    i = 0;
    found = 0 ;
    while (i<=n-1 && !found)
        if (x==a[i])
            found =1 ;
        else i++;
    return found ;
}
```

Ta thấy các lệnh {1}, {2}, {3}, {7} nối tiếp nhau, do đó độ phức tạp của hàm Search chính là độ phức tạp lớn nhất trong 4 lệnh này.

Dễ dàng thấy rằng ba lệnh {1}, {2} và {7} đều có độ phức tạp $O(1) \Rightarrow$ độ phức tạp của hàm Search chính là độ phức tạp của lệnh {3}.

Lòng trong lệnh {3} là lệnh {4}, {5}, {6}.

Lệnh {4}, {5}, {6} có độ phức tạp $O(1)$.

Trong trường hợp xấu nhất(tất cả các phần tử của mảng a đều khác x) thì vòng lặp {3} thực hiện n lần,

$\Rightarrow T(n) = O(n)$

Câu 2 (2,5 điểm):

Mô tả thuật toán: tính hiệu các phần tử trong mảng thỏa $x > y$, lưu trữ kết quả trong 1 mảng. Trả về phần tử có giá trị lớn nhất trong mảng vừa tìm được.

Code

```
int maxDiff(int A[], int n) {  
    vector<int> vect;  
    for (int i = n - 1; i-- >= 0) {  
        for (int j = i - 1; j >= 0; j--) {  
            vect.push_back(A[i] - A[j]);  
        }  
    }  
    sort(vect.begin(), vect.end());  
    return vect.back();  
}
```

Bài toán gồm 2 phần lớn: tính hiệu và tìm hiệu lớn nhất

Với bước tính hiệu, với i nằm trong khoảng từ khoảng $[0; n-1]$, tại mỗi lượt thứ i thực hiện i phép tính nên:

$$\sum_{i=0}^{n-1} i = \frac{(n-1)(n-2)}{2}$$

=> độ phức tạp của bước tính hiệu là: $O(n^2)$

Bước tìm hiệu lớn nhất thì phải sắp xếp lại mảng hiệu và xuất ra phần tử ở cuối mảng. Sử dụng hàm được cung cấp sẵn của C++ có độ phức tạp là $O(n \log n)$.

⇒ Độ phức tạp của toàn bộ thuật toán là: $O(n^2)$

1

Phần II (5 điểm) - Trắc nghiệm nhanh (chọn và giải thích):

(Phần này gồm có 05 câu)

1C. Đặc trưng của thuật toán:

- Tính đúng đắn
- Tính dừng
- Tính xác định

2B. Do không có vòng lặp

3A

4D. Ta có $i: 0 \rightarrow n-1$ và $j: i \rightarrow n$

$\Rightarrow (n-1) * n = n^2 - n = O(n^2)$ (sử dụng quy tắc lấy max)

5D. Câu lệnh so sánh có độ phức tạp $O(1)\{3\}$

Do $j: 0 \rightarrow n-i$ nên có độ phức tạp là $O(n-i)$