Câu 1: 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;
}
```

Giải

- Xét bên trong vòng lặp ***while***, ta thấy các phép gán và so sánh có độ phức tạp là $O(1)$.
- Có thể dễ dàng nhìn thấy vòng lặp ***while*** lặp n lần, nên sẽ có độ phức tạp là $O(n * 1) = O(n)$.
- Ở bên ngoài vòng lặp ***while*** còn có các phép gán ở phía trên với độ phức tạp là $O(1)$.
- Do vòng lặp ***while*** và 2 phép gán cho biến ***i*** và ***found*** là nối tiếp nhau nên độ phức tạp của thuật toán tìm kiếm tuần tự trên là $max(O(n), O(1)) = O(n)$.

Câu 2: Cho một mảng A có n phần tử số nguyên (với n > 1).
Bài toán yêu cầu tìm hiệu lớn nhất khi lấy bất kỳ A[x] – A[y] với x > y.
Ví dụ:
Mảng A = [9, 1, 2, 8,]; n = 4 maxDiff(A,n)
= 7 // bởi vì A[3] – A[1] = 8 -1 = 7
Hãy tìm một thuật toán có độ phức tạp thấp nhất có thể để giải quyết bài toán trên.

Phân tích độ phức tạp thuật toán không đệ quy

Giải

```
int maxDiff(int A[], int n) {
    int maxdiff = A[1] - A[0];
    int minele = A[0];
    for (int i = 0; i < n; i++) {
        if (A[i] - minele > maxdiff)
            maxdiff = A[i] - minele;
        if (A[i] < minele)
            minele = A[i];
    }
    return maxdiff;
}
```

- Thuật toán trên kết hợp việc xét từng phần tử trong mảng và tính khoảng cách của nó đến với phần tử nhỏ nhất của mảng tại từng thời điểm qua n lần lặp để tìm ra được hiệu lớn nhất trong mảng.
- Độ phức tạp của thuật toán trên bị chi phối chủ yếu bởi vòng lặp for với n lần lặp nên sẽ có độ phức tạp là $O(n)$.

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

(5 câu)

Câu 1: Chọn **C. Tính đa hình**

Vì: Thuật toán có 5 tính chất bao gồm: tính chính xác, tính khách quan, tính phổ dụng, tính rõ ràng, tính kết thúc.

Câu 2: Chọn **B.** $\boldsymbol{O(1)}$

Vì: Thuật toán chỉ trả về một kết quả là đáp án của phép tính $n(n+1)$
2
.

Câu 3: Chọn **A. input – input**

Vì: Đây là đúc kết trong khái niệm về big O.

Câu 4: Chọn **D.** $\boldsymbol{O(n2)}$

Vì: Ở đây có 2 vòng lặp lồng vào nhau, xét vòng lặp bên trong thì sẽ có $n - i - 1$ phép gán nên vòng lặp trong sẽ có độ phức tạp là $\boldsymbol{O(n)}$. Vòng lặp ngoài lặp $n - 1$ lần nên thuật toán trên sẽ có độ phức tạp là $\boldsymbol{O(n * n)} = \boldsymbol{O(n2)}$.

Câu 5: Chọn **D.** $\boldsymbol{O(1)}$ **và** $\boldsymbol{O(n - i)}$

Vì: Xét lệnh {3} là phép so sánh nên lệnh {3} sẽ có độ phức tạp là $\boldsymbol{O(1)}$. Xét vòng lặp {2}, ta thấy nó lặp lại $n - i$ lần nên sẽ có độ phức tạp là $\boldsymbol{O(n - i)}$.