

CÂU TẬP CHỦ ĐỀ 1: ĐỘ PHỨC TẬP THUẬT TOÁN

Phản trắc nghiệm (chọn 1 câu trả lời đúng nhất)

Câu 1. Tại sao Big O đóng vai trò quan trọng?

- A. Big-O giúp đo lường độ lớn, hay độ phức tạp của một thuật toán.
- B. Big-O thể hiện hiệu suất hoạt động của một thuật toán dựa trên độ lớn của tập input.
- C. Big-O giúp so sánh, đánh giá hiệu quả của các thuật toán khác nhau.
- D. Big-O giúp đảm bảo thuật toán sẽ dừng.
- E. Tất cả các câu trên (A, B, C, D)
- F. Không có câu trả lời nào đúng

Câu 2. Ký hiệu nào sau đây là ký hiệu của độ phức tạp hằng số (constant time)

- A. $O(n)$
- B. $O(n^2)$
- C. $O(1)$
- D. $O(\log_2 n)$

Câu 3. Thời gian chạy lâu nhất trong các độ phức tạp dưới đây là:

- E. $O(n)$
- F. $O(n^2)$
- G. $O(n!)$
- H. $O(2^n)$

Câu 4. Danh sách nào được sắp xếp từ nhanh nhất đến chậm nhất

- A. $O(n!), O(n^3), O(n^2), O(n \log_2 n), O(\log_2 n), O(1)$
- B. $O(n!), O(n^3), O(n \log_2 n), O(n^2), O(\log_2 n), O(1)$
- C. $O(1), O(\log_2 n), O(n^2), O(n \log_2 n), O(n^3), O(n!)$
- D. $O(1), O(\log_2 n), O(n \log_2 n), O(n^2), O(n^3), O(n!)$

Câu 5: Phát biểu nào ĐÚNG? (1 điểm)

- A. $5^{n+5} = O(5^n)$
- B. $5^{5n} = O(5^n)$
- C. $5^{5n} = O(n^5)$
- D. Không có phát biểu nào đúng

Câu 6: Độ phức tạp thời gian trong tình huống xấu nhất của Mergesort để sắp xếp mảng gồm n phần tử là bao nhiêu? (1 điểm)

- A. $O(n)$
- B. $O(\log_2 n)$
- C. $O(n^2)$
- D. $O(n \log_2 n)$

Câu 7: Độ phức tạp thời gian trong tình huống xấu nhất của Quicksort để sắp xếp mảng gồm n phần tử là bao nhiêu? (1 điểm)

- A. $O(n)$
- B. $O(\log_2 n)$
- C. $O(n^2)$
- D. $O(n \log_2 n)$

Phần tự luận

Câu 1. Cho f và g là hai hàm số thực. Ta nói $f(x)$ là $O(g(x))$ nếu tồn tại hai hằng số c và k sao cho :
 $|f(x)| \leq c|g(x)|$ với mọi $x \geq k$

Hãy chứng minh:

- a. $f(x) = 4x^2 - 5x + 3$ là $O(x^2)$
- b. $f(x) = (x + 5) \log_2(3x^2 + 7)$ là $O(x \log_2 x)$
- c. $f(x) = (x^2 + 5 \log_2 x)/(2x + 1)$ là $O(x)$

Câu 2. Các hàm sau đây có là $O(x)$ hay không?

- a. $f(x) = 10$
- b. $f(x) = 3x + 7$
- c. $f(x) = 2x^2 + 2$

Câu 3. Mô tả thuật toán tìm số nhỏ nhất trong dãy hữu hạn các số tự nhiên. Có bao nhiêu phép so sánh, bao nhiêu phép gán trong thuật toán?

Câu 4. Phân tích độ phức tạp của thuật toán tính tổng dãy số sau:

$$S = 1 + \frac{1}{2} + \frac{1}{6} + \dots + \frac{1}{n!}$$

Câu 5. Cho biết số phép gán, số phép so sánh trong đoạn code sau đây theo n :

```
sum = 0;
for (i = 0; i < n; i++)
{
    scanf("%d", &x);
    sum = sum + x;
}
```

Câu 6. Tính số phép gán, số phép so sánh, chi phí giới hạn trên (big-O) cho các giải thuật sau:

```
a. for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
        b[i][j] += c;
```

```
b. for (i = 0; i < n; i++)
    if (a[i] == k)
        return 1;
    return 0;
```

```
c. for (i = 0; i < n; i++)
    for (j = i+1; j < n; j++)
        b[i][j] -= c;
```

Câu 7. Hãy cho biết các hàm $g(n)$ cho các hàm $f(n)$ dưới đây ($f(n) = O(g(n))$).

- a. $f(n) = (2 + n) * (3 + \log_2 n)$
- b. $f(n) = 11 * \log_2 n + \frac{n}{2} - 3542$
- c. $f(n) = n * (3 + n) - 7n$
- d. $f(n) = \log_2(n^2) + n$

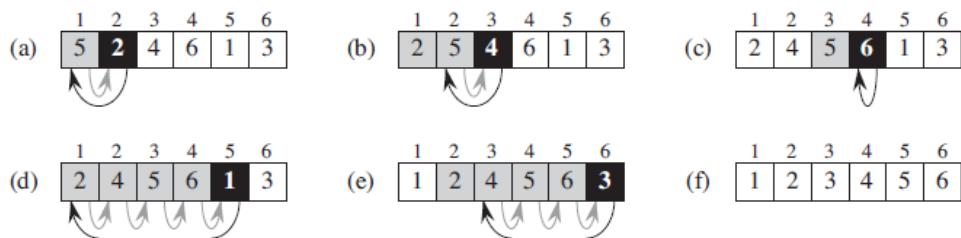
Câu 8. Xác định $O(g(n))$ của các hàm sau đây

- a. $f(n) = 10$
- b. $f(n) = 5n + 3$
- c. $f(n) = 5n + 3$
- d. $f(n) = 10n^2 - 3n + 20$
- e. $f(n) = \log n + 100$
- f. $f(n) = n \log n + \log n + 5$

Câu 9. Phát biểu nào là đúng? Tại sao?

- a. $2^{n+1} = O(2^n)$?
- b. $2^{2n} = O(2^n)$?

Câu 10.



Hình 1. Minh họa thuật toán Insertion Sort

Hình 1 minh họa quá trình thực thi thuật toán **Insertion Sort** (chèn trực tiếp) của mảng $A = <5, 2, 4, 6, 1, 3>$. Chỉ số mảng được viết phía trên mỗi phần tử và giá trị của phần tử được vẽ trong mỗi ô vuông. Hình 1(a)-1(e) là các vòng lặp chèn phần tử của mảng chưa sắp vào đúng vị trí của mảng đã sắp. Trong mỗi vòng lặp, ô tô màu đen giữ giá trị lấy ra từ mảng chưa sắp, giá trị này được so sánh với các giá trị trong ô màu xám (thuộc mảng đã sắp). Các mũi tên màu xám cho biết các

phần tử phải dịch chuyển sang phải 1 vị trí, mũi tên đen cho biết vị trí mà giá trị khóa được chèn vào. Hình 1(f) là mảng sau cùng đã được sắp thứ tự.

Lấy hình 1 làm mẫu, hãy vẽ minh họa quá trình thực thi của thuật toán Insertion Sort cho các mảng sau đây:

- a. $A = < 31, 41, 59, 26, 41, 58 >$
- b. $B = < 100, 77, 49, 29, 51, 100 >$
- c. $C = < 14, 23, 10, 9, 9, 10 >$

Câu 11. Viết lại thuật toán Insertion Sort để sắp xếp mảng theo thứ tự giảm dần.

Câu 12. Xét thuật toán sắp xếp n phần tử trong mảng A như sau: Tìm phần tử có giá trị nhỏ nhất trong mảng A và hoán vị phần tử đó với phần tử A[0]. Tiếp tục tìm phần tử có giá trị nhỏ thứ hai trong mảng A và hoán vị nó với phần tử A[1]. Cứ như vậy cho đến khi hết n-1 phần tử đầu tiên trong A. Đây chính là thuật toán **Selection Sort** (Chọn trực tiếp). Viết thuật toán này bằng C và trả lời các câu hỏi sau:

- a. Tại sao thuật toán chỉ chạy cho n-1 phần tử đầu tiên trong mảng A?
- b. Hãy cho biết độ phức tạp của thuật toán (Big-O) này trong trường hợp tốt nhất? xấu nhất?
- c. So sánh thuật toán Selection Sort với Insertion Sort về thời gian chạy trên cùng 1 máy tính. Thuật toán nào tốt hơn? Trong trường hợp nào?