ĐÁP ÁN QUIZ

1. Tính nào sau đây không phải đặc trưng của thuật toán.

A. Tính đúng đắn.

B. Tính dừng.

C. Tính đa hình.

D. Tính xác định.

Đáp án: C

Các đặc trưng của thuật toán là: đúng đắn, dừng, xác định, hiệu quả, phổ quát.

Còn đa hình là tính chất của lập trình hướng đối tượng.

2. Độ phức tạp của thuật toán có khái niệm khác là?

A. Big O

B. Small O

C. Big D

D. Small D

Đáp án: A

Khái niệm Big O hoạc với tên gọi khác trong tiếng Việt là “độ phức tạp của thuật toán” là thuật ngữ thường dùng để chỉ khoảng thời gian tiêu hao để chạy một thuật toán.

3. Cho đoạn chương trình sau:

int sum(int n)

{

return n\*(n+1)/2;

}

Độ phức tạp của thuật toán này là:

A. O(n).

B. O(1).

C. O(n^2).

D. O(n-1).

Đáp án: B

Với đoạn chương trình này với mọi giá trị đầu vào chương trình chỉ thực thi đúng 1 câu lệnh return. Do đó, ta nói độ phức tạp của chương trình trên là O(1).

4. Điền vào chỗ trống:

Khi nói về Big O thì thời gian chạy nhanh như thế nào, còn tùy thuộc vào [....], vì giá trị [...] sẽ lớn dần lên khi chương trình chạy.

A. input - input.

B. output - ouput.

C. input - output.

D. output - input.

Đáp án: A

Khi bắt đầu chạy chương trình mọi người thường nghĩ sẽ thật tốn công sức nếu người viết một thuật toán phức tạp để xử lý giá trị đầu vào nhỏ, nhưng qua từng bước của thuật toán đó, giá trị ban đầu vốn nhỏ lại dần trở nên lớn.

5. Chọn đáp án sai trong các phát biểu sau:

A. O(1) có độ phức tạp về thời gian luôn giống nhau

B. Thuật toán được đánh giá bằng thời gian và không gian

C. O(n) có thời gian tăng dần theo lượng Input

D. Khi giải quyết độ phức tạp, chúng ta luôn xét đến thời gian chạy tốt nhất

Đáp án D.

Khi giải quyết độ phức tạp, chúng ta luôn xét đến thời gian chạy xấu nhất

6. Thuật toán Interchange được hiện thực hiện như sau:

void interchangeSort(int \*a, int n)

{

for (int i = 0; i < n-1; i++)

for (int j = i+1; j < n; j++)

if (a[i] > a[j])

{

int temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

Cho biết độ phức tạp của thuật toán:

A. O(1)

B. O(n)

C. n

D. O(n²)

Đáp án: D

Trong hàm này, i sẽ chạy từ 0 đến n - 2. Ứng với mỗi giá trị của i vòng for trong sẽ chạy n - i - 1 lần (j từ i + 1 đến n - 1), do đó số phép so sánh cần thực hiện sẽ là Σ(n - i - 1) (với i chạy từ 0 đến n - 2) = (n - 1) + (n - 2) + (n - 3) +...+ 2 + 1 = (n - 1)\*n/2 = n²/2 - n/2. Ta nói độ phức tạp của thuật toán trên là O(n²) (thực ra là O(n²/2 - n/2)).

7. Chọn biểu thức đúng với quy tắc bỏ hằng số

A. T(n) = O(c.f(n)) = O(f(n)) với c là một hằng số bất kì

B. T(n) = O(c.f(n)) = O(f(n)) với c là một hằng số dương

C. T(n) = O(c + f(n)) = O(f(n)) với c là một hằng số bất kì

D. T(n) = O(c + f(n)) = O(f(n)) với c là một hằng số dương

Đáp án: B

Quy tắc bỏ hằng: Nếu đoạn chương trình P có thời gian thực hiện T(n) = O(c1.f(n)) với c1 là một hằng số dương thì có thể coi đoạn chương trình đó có độ phức tạp tính toán là O(f(n)).

8. Trong quy tắc tổng quát, thời gian thực hiện mỗi lệnh gán là

A. O(1).

B. O(n).

C. O(n-1).

D. O(n²)

Đáp án: A

Trong quy tắc tổng quát, thời gian thực hiện mỗi lệnh gán là O(1)

9. Trong quy tắc tổng quát, thời gian thực hiện chuỗi lệnh tuần tự được xác định bằng:

A. Thời gian thực hiện mỗi lệnh gán.

B. Quy tắc nhân.

C. Quy tắc cộng.

D. Không xác định.

Đáp án: C

Thời gian thực hiện của một chuỗi tuần tự các lệnh được xác định bằng qui tắc cộng => thời gian này là thời gian thi hành một lệnh nào đó lâu nhất trong chuỗi lệnh.

10. Cho thuật toán sắp xếp nổi bọt (Bubble Sort) được cài đặt như sau:

def bubble\_sort(arr):

n = len(arr)

# Duyệt qua tất cả các phần tử của mảng

for i in range(n): # {1}

for j in range(0, n-i-1): # {2}

# duyệt qua các phần tử của mảng từ 0 đén n-i-1

# Đổi chỗ nếu tìm thấy phần tử lớn hơn

if arr[j] > arr[j+1] : # {3}

arr[j], arr[j+1] = arr[j+1], arr[j] # {4}

Áp dụng quy tắc tổng quát, xác định lần lượt thời gian thực hiện lệnh {3} và thời gian chạy vòng lặp {2}

A. O(1) và O(n -1)

B. O(n²) và O(n - 1)

C. O(n² - 1) và O(n - i)

D. O(1) và O(n - i)

Đáp án: D

Ta thấy trong chương trình gồm một lệnh lặp {1}, lồng trong lệnh lặp {1} là lệnh lặp {2}. Lồng trong lệnh lặp 2 là lệnh điều kiện {3}, trong lệnh điều kiện {3} là khối lệnh {4}.

Khối lệnh 4 thực ra là ba lệnh:

temp = arr[j+1]

arr[j+1] = arr[j]

arr[j] = temp

Khối 3 lệnh trên tốn thời gian là O(1), câu lệnh điều kiện cũng tốn thời gian là O(1). Do đó thời gian thực hiện lệnh {3} là O(1).

Vòng lặp {2} chạy n-i lần, mỗi lần O(1) nên thời gian chạy vòng lặp 2 là O((n-i) \* 1) = O(n-i).