

**Câu hỏi trắc nghiệm**

Lưu ý: chỉ chọn một đáp án duy nhất.

**Câu 1.** Cho list **arr = [9, 7, 4, 2, 1]**. Thuật toán Counting Sort sẽ sử dụng tổng cộng bao nhiêu phép so sánh để hoàn tất sắp xếp *arr*?

1. 0
2. 4
3. 5
4. 16

**Câu 2.** Thuật toán Counting Sort trong bài học sử dụng thêm bao nhiêu không gian nhớ, không tính mảng *arr* ban đầu? Gọi n là số phần tử cần sắp xếp và k là giá trị của phần tử lớn nhất.

1. O(n)
2. O(k)
3. O(n+k)
4. O(n.k)

**Câu 3.** Counting Sort là thuật toán sắp xếp nhanh nhất vì:

1. Thuật toán này không hoạt động dựa trên phương pháp so sánh các phần tử
2. Thuật toán này có độ phức tạp về thời gian là O(n+k)
3. Thuật toán này được thiết kế đặc biệt để sắp xếp danh sách các số nguyên
4. Counting Sort không được coi là thuật toán sắp xếp nhanh nhất

**Câu 4.** Counting Sort không thể thực thi được trên danh sách chứa các phần tử là:

1. Số thực
2. Số âm
3. Object bất kỳ
4. Tất cả các ý trên

**Câu 5.** Các thuật toán nào dưới đây phải được viết dưới dạng đệ quy?

1. Merge Sort
2. Quicksort
3. A và B
4. Không thuật toán nào

**Câu 6.** Với cách cài đặt trong bài học, thuật toán nào phải sử dụng thêm bộ nhớ trong quá trình sắp xếp?

1. Bubble Sort, Quicksort
2. Merge Sort, Quicksort
3. Insertion Sort, Merge Sort
4. Counting Sort, Merge Sort

**Câu 7.** Counting Sort và Merge Sort có điểm chung là:

1. Số phép toán phải thực hiện được giảm thiểu khi dữ liệu đầu vào gần như đã được sắp xếp
2. Rơi vào trường hợp xấu nhất khi dữ liệu đầu vào được sắp xếp ngẫu nhiên
3. Rơi vào trường hợp xấu nhất khi dữ liệu đầu vào được sắp xếp ngược với thứ tự mong muốn
4. Số phép toán phải thực hiện không phụ thuộc vào trạng thái sắp xếp của dữ liệu đầu vào

**Câu 8.** Thuật toán sắp xếp nhanh nhất hiện tại là:

1. Quicksort
2. Merge Sort
3. Timsort trong hàm *sorted()* của Python
4. Không thể xác định do tốc độ của thuật toán còn phụ thuộc vào đặc trưng của dữ liệu đầu vào

# Đáp án

| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | D | C | D | D | D | D |