# CÂU HỎI TRẮC NGHIỆM - THUẬT TOÁN SẮP XẾP

1. Câu 1: Merge Sort có độ phức tạp thời gian tốt nhất, trung bình và tệ nhất là gì?

* A. O(n), O(n log n), O(n²)
* B. O(n log n), O(n log n), O(n log n)
* C. O(n log n), O(n²), O(n²)
* D. O(log n), O(n log n), O(n²)

Đáp án đúng: B

Giải thích: Merge Sort luôn thực hiện chia và trộn độc lập với dữ liệu, nên độ phức tạp là O(n log n) trong mọi trường hợp.

1. Câu 2: Khi áp dụng Merge Sort, tại sao cần phải dùng bộ nhớ phụ?

* A. Vì Merge Sort sử dụng quay lui
* B. Vì cần lưu mảng con đã sắp xếp
* C. Vì việc chia mảng đòi hỏi mảng phụ
* D. Vì trộn các phần tử cần vùng nhớ tạm để lưu kết quả

Đáp án đúng: D

Giải thích: Trộn hai mảng con cần bộ nhớ phụ để giữ các phần tử trước khi ghi lại vào mảng chính.

1. Câu 3: Merge Sort có thể hiệu quả hơn Quick Sort trong trường hợp nào?

* A. Khi dữ liệu có phân phối đồng đều
* B. Khi không gian bộ nhớ rất hạn chế
* C. Khi cần độ ổn định và dữ liệu lớn
* D. Khi dữ liệu gần như đã được sắp xếp

Đáp án đúng: C

Giải thích: Merge Sort ổn định và có hiệu năng tốt trên dữ liệu lớn hoặc cần bảo toàn thứ tự phần tử.

1. Câu 4: Natural Merge Sort khác gì với Merge Sort cổ điển?

* A. Dùng thêm cấu trúc heap
* B. Không cần chia đều mảng
* C. Tận dụng các chuỗi tăng tự nhiên trong dữ liệu
* D. Không cần bộ nhớ phụ

Đáp án đúng: C

Giải thích: Natural Merge Sort phát hiện các chuỗi đã có thứ tự tăng để giảm số lần trộn.

1. Câu 5: Trong Natural Merge Sort, “chuỗi tăng tự nhiên” là gì?

* A. Một đoạn có giá trị tăng đều
* B. Một dãy con đã được sắp xếp tăng
* C. Một phần của mảng có giá trị giống nhau
* D. Một chuỗi có giá trị xen kẽ

Đáp án đúng: B

Giải thích: Là đoạn liên tiếp trong mảng đã có thứ tự tăng.

1. Câu 6: Ưu điểm của Natural Merge Sort so với Merge Sort thường là gì?

* A. Luôn có độ phức tạp tốt hơn
* B. Không cần dùng bộ nhớ phụ
* C. Sử dụng đệ quy ít hơn
* D. Giảm số lần trộn khi dữ liệu đã gần sắp xếp

Đáp án đúng: D

Giải thích: Natural Merge Sort tận dụng các chuỗi tăng có sẵn để giảm chi phí trộn.

1. Câu 7: Trong Natural Merge Sort, nếu toàn bộ mảng là một chuỗi tăng, ta cần bao nhiêu lượt trộn?

* A. log n
* B. 1
* C. n/2
* D. 0

Đáp án đúng: B

Giải thích: Vì mảng đã sắp, chỉ cần nhận diện chuỗi tăng duy nhất và không cần trộn lại.

1. Câu 8: Trong K-way Merge Sort, “K” đề cập đến?

* A. Số lần chia mảng
* B. Số phần tử trong một block
* C. Số luồng xử lý song song
* D. Số đoạn con được trộn cùng lúc

Đáp án đúng: D

Giải thích: K là số mảng con được trộn cùng lúc trong mỗi bước.

1. Câu 9: Cấu trúc dữ liệu phù hợp nhất để trộn K mảng đã sắp xếp là?

* A. Mảng
* B. Stack
* C. Heap (Priority Queue)
* D. Hash Table

Đáp án đúng: C

Giải thích: Min-Heap giúp chọn phần tử nhỏ nhất từ K mảng hiệu quả.

1. Câu 10: Độ phức tạp thời gian của K-way Merge bằng cách dùng Min-Heap là?

* A. O(n log n)
* B. O(n log K)
* C. O(K log n)
* D. O(nK)

Đáp án đúng: B

Giải thích: Vì mỗi phần tử được thêm hoặc loại khỏi heap có log K thao tác, tổng cộng n phần tử.