

Question	Option 1	Option 2	Option 3	Option 4	Correct option number
Thuật toán là:	Thuật toán là một hệ thống chặt chẽ và rõ ràng các quy tắc nhằm xác định một dãy thao tác trên cấu trúc dữ liệu sao cho: với một dữ liệu đầu vào, sau một số hữu hạn bước thực hiện các thao tác đã chỉ ra, ta đạt được mục tiêu nhất định.	Thuật toán là một hệ thống lỏng lẻo và mập mờ các quy tắc nhằm xác định một dãy thao tác trên cấu trúc dữ liệu sao cho: với một dữ liệu đầu vào, sau một số hữu hạn bước thực hiện các thao tác đã chỉ ra, ta không đạt được mục tiêu nhất định.	Thuật toán là một hệ thống chặt chẽ và rõ ràng các quy tắc nhằm xác định một dãy thao tác trên cấu trúc dữ liệu sao cho: với một dữ liệu đầu vào, sau một số hữu hạn bước thực hiện các thao tác đã chỉ ra, ta không đạt được mục tiêu nhất định.	Thuật toán là một hệ thống lỏng lẻo và mập mờ các quy tắc nhằm xác định một dãy thao tác trên cấu trúc dữ liệu sao cho: với một dữ liệu đầu vào, sau một số hữu hạn bước thực hiện các thao tác đã chỉ ra, ta đạt được mục tiêu nhất định.	1
Đặc tính nào dưới đây không phải là đặc trưng của thuật toán?	Tính khả thi: + Thuật toán yêu cầu không gian lưu trữ đủ nhỏ để có thể lưu trữ được trên máy tính. + Thuật toán yêu cầu thời gian thực thi đủ nhỏ để máy tính có thể thực hiện. + Thuật toán đủ dễ hiểu để thể cài đặt được	Tính dừng: + Sau khi thực hiện tất cả các bước của thuật toán theo đúng trình tự đã định, ta phải được kết quả mong muốn với mọi dữ liệu đầu vào	Tính đơn điệu: + Thời gian thực thi và không gian yêu cầu liên tục tăng khi kích thước của bài toán tăng.	Tính dừng: + Thuật toán không được rơi vào vô hạn, phải dừng lại và cho kết quả sau một số hữu hạn bước	3
Để lập trình hiệu quả cao, ta cần kỹ thuật lập trình tốt. <i>Vậy kỹ thuật lập trình tốt thể</i> Có những loại lỗi nào khi lập trình?	Các thiết bị phần cứng tốt và chạy nhanh.	Phần mềm hỗ trợ lập trình tốt và tiện dụng.	Tốc độ gõ code nhanh.	Khả năng viết chương trình, khả năng gỡ rối và thao tác nhanh.	4
	Lỗi cú pháp.	Lỗi cài đặt.	Lỗi thuật toán	Cả ba lỗi trên.	4
Các hợp lý để xây dựng bộ test là:	1. Kiểm tra tính đúng đắn của chương trình với các bộ test nhỏ, đơn giản, làm bằng tay có thể ra được đáp án. 2. Kiểm tra tính đúng đắn của chương trình bằng các bộ test nhỏ chứa những giá trị đặc biệt hoặc bất thường. 3. Kiểm tra tính chịu đựng của chương trình bằng các bộ test lớn.	1. Kiểm tra tính đúng đắn của chương trình với các bộ test nhỏ, đơn giản, làm bằng tay có thể ra được đáp án. 2. Kiểm tra tính đúng đắn của chương trình bằng các bộ test nhỏ chứa những giá trị đặc biệt hoặc bất thường. 3. Kiểm tra tính chịu đựng và tính đúng đắn của chương trình bằng các bộ test lớn.	1. Kiểm tra tính đúng đắn của chương trình với các bộ test nhỏ, đơn giản, làm bằng tay có thể ra được đáp án. 2. Kiểm tra tính chịu đựng của chương trình bằng các bộ test nhỏ chứa những giá trị đặc biệt hoặc bất thường. 3. Kiểm tra tính chịu đựng của chương trình bằng các bộ test lớn.	1. Kiểm tra tính đúng đắn của chương trình với các bộ test nhỏ, đơn giản, làm bằng tay có thể ra được đáp án. 2. Kiểm tra tính đúng đắn của chương trình bằng các bộ test nhỏ chứa những giá trị đặc biệt hoặc bất thường. 3. Kiểm tra tính đúng đắn của chương trình bằng các bộ test lớn.	1
Tìm kiếm nhị phân được áp dụng khi nào?	Khi các khóa tìm kiếm đã được sắp xếp.	Khi các giá trị cần tìm kiếm đã được sắp xếp.	Khi cả khóa tìm kiếm và giá trị tìm kiếm đều đã được sắp xếp.	Khi một trong hai khóa tìm kiếm hoặc giá trị tìm kiếm đã được sắp xếp.	1
Việc chọn "khóa" cho việc tìm kiếm trong tìm kiếm nhị phân dẫn tới việc:	Phải thiết lập cấu trúc dữ liệu sao cho giá trị tìm kiếm được sắp xếp và dễ dàng truy cập tới khóa tìm kiếm.	Phải thiết lập cấu trúc dữ liệu sao cho khóa tìm kiếm được sắp xếp và dễ dàng truy cập tới giá trị tìm kiếm.	Code khó khăn hơn.	Code dễ dàng hơn.	2
Độ phức tạp trung bình của tìm kiếm nhị	O(n)	O(n ²)	O(nlogn)	O(logn)	4
Bài toán tối ưu A yêu cầu tìm kết quả lớn nhất phù hợp điều kiện X. Anh Jong Bức Chỉ đã dùng phương pháp vét cạn và đưa ra được giải thuật như sau: 1. Lập i có giá trị tăng từ 1 đến N. 2. Sinh cấu c với giá trị i tương ứng. 3. Kiểm tra nếu cấu hình c không thỏa mãn thì kết quả là i - 1. Để tăng tốc thuật toán, anh Jong Bức Chỉ đã nghĩ đến việc dùng thuật toán tìm kiếm nhị phân (binary search). Theo bạn, cách áp dụng phù hợp nhất là:	Không áp dụng được tìm kiếm nhị phân cho bài toán A.	1. Khởi tạo kết quả i = 1 và bước tăng step = 1. 2. Sinh cấu hình c với giá trị i tương ứng. 3. Kiểm tra tính hợp lệ của cấu hình c 3.1 Nếu c hợp lệ, tăng gấp đôi bước tăng step = step x 2. 3.2 Nếu c không hợp lệ. Reset i về giá trị cũ hợp lệ i = i - step. Reset step về 1 step = 1. Giá trị i = i + step. 4. Dừng khi c không hợp lệ và step = 1. Kết quả là i - 1.	1. Lập i có giá trị giảm dần từ N về 1. 2. Sinh cấu c với giá trị i tương ứng. 3. Dừng khi cấu hình c thỏa mãn. Kết quả là i. 4. Kiểm tra tính hợp lệ của cấu hình c. 4.1 Nếu c hợp lệ, gán giá trị low = i. 4.2 Nếu c không hợp lệ, gán giá trị high = i. 5. Lập lại bước 2 cho đến khi low = high - 1.	1. Bắt đầu bài toán có kích thước low = 1, high = N. 2. Tính giá trị trung bình i = (low + high) / 2. 3. Sinh cấu hình c cho giá trị i ở bước 2. 4. Kiểm tra tính hợp lệ của cấu hình c. 4.1 Nếu c hợp lệ, gán giá trị low = i. 4.2 Nếu c không hợp lệ, gán giá trị high = i. 5. Lập lại bước 2 cho đến khi low = high - 1.	4
Cách tăng tốc thuật toán mà bạn đã áp dụng ở câu hỏi trên là:	Tăng tốc độ thực thi.	Giảm không gian tìm kiếm.	Đó là hai thuật toán không liên quan gì đến nhau.	Sắp xếp lại không gian tìm kiếm.	4
Các thành phần theo thứ tự thực hiện của thuật toán chia để trị là:	1. Conquer (trị) 2. Combine (kết hợp) 3. Divide (chia)	1. Divide (chia) 2. Combine (kết hợp) 3. Conquer (trị)	1. Divide (chia) 2. Conquer (trị) 3. Combine (kết hợp)	1. Conquer (trị) 2. Divide (chia) 3. Combine (kết hợp)	3
"Chia" (divide) trong thuật toán chia để trị là bước:	Chia bài toán thành các bài toán nhỏ hơn. Bước này thường thực hiện một cách đệ quy để chia bài toán nhỏ cho đến khi bài toán nhỏ không thể giải được nữa.	Chia đôi bài toán thành hai bài toán nhỏ hơn. Bước này thường thực hiện một cách đệ quy để chia bài toán nhỏ cho đến khi không chia nhỏ được nữa.	Chia bài toán thành các bài toán nhỏ hơn. Bước này thường thực hiện một cách đệ quy để chia bài toán nhỏ cho đến khi không chia nhỏ được nữa.	Chia đôi bài toán thành hai bài toán nhỏ hơn. Bước này thường thực hiện một cách đệ quy để chia bài toán nhỏ cho đến khi bài toán nhỏ không thể giải được nữa.	3
"Trị" (conquer) trong thuật toán chia để trị là:	Giải quyết bài toán con đã được chia nhỏ có thể giải bằng một thuật toán khác.	Giải quyết bài toán con đã được chia nhỏ đến mức đơn giản có thể giải quyết trực tiếp	Giải quyết bài toán ban đầu một cách trực tiếp.	Giải quyết bài toán ban đầu bằng một thuật toán khác.	2
"Kết hợp" (combine) trong thuật toán chia để trị là:	Kết hợp phần cứng và phần mềm để giải quyết bài toán.	Kết hợp các lời giải của bài toán bằng các thuật toán khác nhau.	Khi các bài toán nhỏ hơn được giải quyết, kết hợp chúng một cách đệ quy cho đến khi chúng là lời giải cho bài toán ban đầu.	Không đáp án nào đúng.	3
Các bước trong thuật toán sắp xếp nhanh (quick sort) là:	1. Phân vùng 2. Gọi đệ quy hai dãy con vừa được phân vùng	1. Kiểm tra kích thước 2. Chọn pivot 3. Phân vùng 4. Gọi đệ quy hai dãy con vừa được phân vùng	1. Chọn pivot 2. Kiểm tra kích thước 3. Phân vùng 4. Gọi đệ quy hai dãy con vừa được phân vùng	1. Phân vùng 2. Kiểm tra kích thước 3. Chọn pivot 4. Gọi đệ quy hai dãy con vừa được phân vùng	2
Độ phức tạp tốt nhất, trung bình và xấu nhất của thuật toán sắp xếp nhanh lần lượt là:	O(n ²) , O(n ²), O(n ²)	O(nlogn) , O(nlogn), O(nlogn)	O(nlogn) , O(n ²), O(n ²)	O(nlogn) , O(nlogn), O(n ²)	4

Các bước tương ứng của thuật toán sắp xếp nhanh với các thành phần của thuật toán chia để trị là:	+ Divide: Kiểm tra kích thước + chọn pivot + phân vùng + Conquer: Gọi đệ quy hai dãy con vừa được phân vùng. + Combine: N/A	+ Divide: Chọn pivot + phân vùng + Conquer: Kiểm tra kích thước. + Combine: Gọi đệ quy hai dãy con vừa được phân vùng	+ Divide: Phân vùng + Conquer: Gọi đệ quy hai dãy con vừa được phân vùng + Combine: Kiểm tra kích thước + Chọn pivot.	+ Divide: Kiểm tra kích thước + Conquer: Chọn pivot + phân vùng + Combine: Gọi đệ quy hai dãy con vừa được phân vùng.	1
Các bước trong thuật toán sắp xếp trộn (merge sort) là:	1. Chia danh sách chưa được sắp xếp thành hai danh sách con, mỗi danh sách chứa 1 nửa số phần tử. 2. Lập lại bước 1 cho đến khi danh sách chỉ còn một phần tử. 3. Liên tục hợp nhất hai danh sách con để tạo ra các danh sách con được sắp xếp mới cho đến khi chỉ còn lại một danh sách con	1. Chia danh sách chưa được sắp xếp thành n danh sách con, mỗi danh sách chứa 1 phần tử 2. Liên tục hợp nhất các danh sách con để tạo ra các danh sách con được sắp xếp mới cho đến khi chỉ còn lại một danh sách con	1. Bắt đầu từ danh sách một phần tử. 2. Liên tục hợp nhất một phần tử mới vào danh sách đang có để có được danh sách được sắp xếp.	Không đáp án nào đúng.	2
Độ phức tạp tốt nhất, trung bình và xấu nhất của thuật toán sắp xếp trộn là:	$O(n \log n)$, $O(n \log n)$, $O(n^2)$	$O(n \log n)$, $O(n \log n)$, $O(n \log n)$	$O(n \log n)$, $O(n^2)$, $O(n^2)$	$O(n^2)$, $O(n^2)$, $O(n^2)$	2
Bài toán con trong thuật toán chia để trị là:	Bài toán con có kích thước bằng một nửa bài toán cha.	Bài toán con có kích thước nhỏ hơn bài toán cha một đơn vị.	Bài toán con có kích thước nhỏ hơn bài toán cha.	Là bài toán con có kích thước giống với bài toán cha nhưng loại bỏ đi một số yêu cầu	3