|  |  |
| --- | --- |
| Chọn phát biểu đúng trong các phát biểu dưới đây: bằng cách chạy thử một thuật toán với một bộ dữ liệu, ta có thể? | |
| A | Khẳng định thuật toán đúng nếu nó cho kết quả đúng. |
| B | Khẳng định thuật toán sai nếu nó cho kết quả sai. |
| C | Khẳng định thuật toán tốt nếu nó cho kết quả nhanh. |
| D | Khẳng định thuật toán hiệu quả nếu nó cho kết quả đúng. |

|  |  |
| --- | --- |
| Tìm mệnh đề SAI trong các mệnh đề sau? | |
| A | Một cấu trúc dữ liệu bao gồm các thành phần đặt sát nhau trong bộ nhớ |
| B | Một cấu trúc dữ liệu bao gồm các kiểu dữ liệu cơ sở, được liên kết với nhau thành một cấu trúc thống nhất, tiện lợi cho quá trình xử lý |
| C | Một cấu trúc dữ liệu bao gồm một tập hợp nào đó các thành phần dữ liệu |
| D | Một cấu trúc dữ liệu bao gồm các kiểu dữ liệu phức tạp, được xây dựng nên từ các kiểu dữ liệu đã có, đơn giản hơn bằng phương pháp liên kết nào đó |

|  |  |
| --- | --- |
| Mối liên hệ giữa Cấu trúc dữ liệu và Giải thuật được minh họa bằng đẳng thức nào? | |
| A | Cấu trúc dữ liệu + Chương trình = Giải thuật |
| B | Cấu trúc dữ liệu = Chương trình |
| C | Cấu trúc dữ liệu + Giải thuật = Chương trình |
| D | Chương trình + Giải thuật = Cấu trúc dữ liệu |

|  |  |
| --- | --- |
| Để đánh giá một cấu trúc dữ liệu, chúng ta thường dựa vào tiêu chí nào trong các tiêu chí dưới đây? | |
| A | Cấu trúc dữ liệu phải tiết kiệm tài nguyên (bộ nhớ trong) |
| B | Cấu trúc dữ liệu phải phản ánh đúng thực tế của bài toán |
| C | Cấu trúc dữ liệu phải dễ dàng trong việc thao tác dữ liệu |
| D | Cấu trúc dữ liệu phải tiết kiệm tài nguyên, phản ánh đúng thực tế của bài toán và dễ dàng trong việc thao tác dữ liệu |

|  |  |
| --- | --- |
| Một cấu trúc dữ liệu tốt cần thỏa mãn điều kiện nào trong các điều kiện sau? | |
| A | Phản ánh đúng thực tế; tốc độ xử lý của máy tính phải nhanh; tiết kiệm tài nguyên hệ thống. |
| B | Các thao tác xử lý dữ liệu phải đơn giản; phù hợp với các thao tác xử lý; phản ánh đúng thực tế. |
| C | Các thao tác xử lý dữ liệu phải đơn giản; phù hợp với các thao tác xử lý; tiết kiệm tài nguyên hệ thống. |
| D | Phản ánh đúng thực tế; phù hợp với các thao tác xử lý; tiết kiệm tài nguyên hệ thống. |
| Trong các tiêu chí đánh giá một cấu trúc dữ liệu, tiêu chí nào dưới đây KHÔNG đúng? | |
| A | Phù hợp với các thao tác xử lý |
| B | Phản ánh đúng thực tế |
| C | Tốc độ xử lý dữ liệu phải nhanh, chiếm ít bộ nhớ |
| D | Tiết kiệm tài nguyên của hệ thống |

|  |  |
| --- | --- |
| 5 đặc trưng cơ bản của giải thuật là gì? | |
| A | Đầu vào; đầu ra; tính khả thi; tính dừng; dữ liệu |
| B | Đầu vào; đầu ra; tính xác định; tính khả thi; tính dừng |
| C | Dữ liệu; đầu vào; đầu ra; giải thuật; tính dừng |
| D | Đầu vào; đầu ta; dữ liệu; giải thuật; tính khả thi |

|  |  |
| --- | --- |
| Các cấu trúc nào dưới đây thường được dùng để thiết kế giải thuật? | |
| A | Cấu trúc tuần tự; cấu trúc chọn lựa; cấu trúc lặp |
| B | Cấu trúc dữ liệu; cấu trúc lặp; cấu trúc tuần tự |
| C | Cấu trúc chọn lựa; cấu trúc lặp; cấu trúc dữ liệu |
| D | Cấu trúc tuần tự; cấu trúc lặp; cấu trúc dữ liệu |

|  |  |
| --- | --- |
| Để lưu trữ một danh sách số liệu dân số 100 hộ gia đình kèm các thông số kèm theo (như: Giới tính, ngày sinh, nghề nghiệp, số con, ...), sử dụng kiểu dữ liệu nào trong các kiểu dữ liệu dưới đây là tối ưu nhất (tiết kiệm bộ nhớ)? | |
| A | Cấu trúc |
| B | Mảng |
| C | Ngăn xếp |
| D | Con trỏ |

|  |  |
| --- | --- |
| Để lưu trữ số lượng thí sinh đến dự thi trung học phổ thông của trường, sử dụng kiểu dữ liệu nào trong các kiểu dữ liệu dưới đây là tối ưu nhất (tối ưu bộ nhớ nhất)? | |
| A | Mảng |
| B | Cây |
| C | Danh sách liên kết |
| D | Hàng đợi |

|  |  |
| --- | --- |
| Kiểu dữ liệu T được xác định bởi 1 bộ <V, O>, trong đó V, O là gì? | |
| A | V: là tập các giá trị hợp lệ mà một đối tượng kiểu T có thể lưu trữ  O: là tập các thao tác xử lý có thể thi hành trên đối tượng kiểu T |
| B | V: là tập các thao tác xử lý có thể thi hành trên đối tượng kiểu T  O: là tập các giá trị hợp lệ mà đối tượng kiểu T có thể lưu trữ |
| C | V: là tập các thao tác xử lý mà một đối tượng kiểu T có thể lưu trữ  O: là tập các kết quả dữ liệu đầu vào của chương trình |
| D | V: là tập các giá trị hợp lệ mà một đối tượng kiểu T có thể lưu trữ  O: là tập các kết quả dữ liệu đầu ra của chương trình |

|  |  |
| --- | --- |
| Một kiểu dữ liệu cần phải có những thuộc tính nào trong các thuộc tính dưới đây? | |
| A | Tên kiểu; kích thước lưu trữ; tập các toán hạng, toán tử tác động lên kiểu dữ liệu |
| B | Tên kiểu; miền giá trị; tập các toán hạng, toán tử tác động lên kiểu dữ liệu |
| C | Miền giá trị; kích thước lưu trữ; tập các toán hạng, toán tử tác động lên kiểu dữ liệu |
| D | Tên kiểu; miền giá trị; kích thước lưu trữ; tập các toán tử tác động lên kiểu dữ liệu |

|  |  |
| --- | --- |
| Bài toán liệt kê là bài toán gì? | |
| A | Là bài toán yêu cầu đưa ra danh sách các cấu hình |
| B | Là bài toán phân tích đối tượng thành nhiều thành phần |
| C | Là bài toán số tổ hợp chập k của n theo công thức truy hồi |
| D | Là bài toán chia bài toán lớn thành các bài toán nhỏ hơn để tìm kết quả |

|  |  |
| --- | --- |
| Để lưu trữ số lượng khách hàng hàng ngày ở một siêu thị, ta nên sử dụng dữ liệu kiểu gì? Vì sao? | |
| A | Kiểu danh sách liên kết, vì số lượng khách hàng không cố định |
| B | Kiểu bản ghi, vì khách hàng cần lưu trữ nhiều thông tin |
| C | Kiểu mảng, vì dễ truy cập đến một khách hàng bất kỳ |
| D | Kiểu cấu trúc, vì khách hàng cần lưu trữ nhiều thông tin |

|  |  |
| --- | --- |
| Khi muốn lưu trữ điểm trung bình chung của sinh viên (số sinh viên cố định), dùng dữ liệu kiểu nào là tối ưu nhất? Vì sao? | |
| A | Kiểu bản ghi, vì sinh viên cần lưu trữ nhiều thông tin |
| B | Kiểu con trỏ, vì số lượng sinh viên không cố định |
| C | Kiểu mảng, vì biết trước số lượng các phần tử |
| D | Kiểu số thực (float), vì điểm trung bình nhận các số thực |

|  |  |
| --- | --- |
| Khi lưu trữ điểm của môn học CTDL và GT (theo thang điểm 0, 1, 2,… 10); trong các kiểu dữ liệu sau, kiểu nào gây lãng phí nhiều nhất? | |
| A | char |
| B | long |
| C | int |
| D | float |

|  |  |
| --- | --- |
| Để kiểm tra tính đúng đắn của giải thuật chúng ta có thể? | |
| A | Cài đặt thuật toán đó và cho thực hiện trên máy với một số bộ dữ liệu mẫu |
| B | Đo lường thời gian thực hiện của hoạt động trên một máy tính xác định hợp với dữ liệu đầu vào được chọn |
| C | Cài đặt giải thuật đó và cho thực hiện trên máy với một số bộ dữ liệu mẫu rồi lấy kết quả đó so sánh với kết quả đã biết |
| D | Xác định bởi số các lệnh được thực hiện trong một máy tính lý tưởng |

|  |  |
| --- | --- |
| Thuật toán là gì? | |
| A | Thuật toán là một dãy các bước, mỗi bước mô tả chính xác các phép toán thực hiện trên dữ liệu đưa vào |
| B | Thuật toán là một dãy các bước, mỗi bước mô tả chính xác các phép toán hoặc hành động cần thực hiện để giải quyết một vấn đề |
| C | Thuật toán là một dãy hữu hạn các bước, mỗi bước mô tả chính xác các phép toán hoặc hành động cần thực hiện để giải quyết một vấn đề |
| D | Thuật toán là một dãy hữu hạn các bước, thực hiện xử lý dữ liệu. |

|  |  |
| --- | --- |
| Để lưu trữ điểm trung bình chung học tập của sinh viên (điểm trung bình được tính theo công thức trung bình cộng của các môn học có hệ số), sử dụng kiểu dữ liệu nào trong các kiểu dữ liệu dưới đây là tối ưu nhất? | |
| A | float |
| B | int |
| C | double |
| D | longint |

|  |  |
| --- | --- |
| Để lưu trữ 50 phần tử các số nguyên, sử dụng kiểu cấu trúc dữ liệu nào trong các kiểu cấu trúc dữ liệu dưới đây là tối ưu nhất? | |
| A | int |
| B | Mảng |
| C | Cấu trúc |
| D | Cây |

|  |  |
| --- | --- |
| Trong các đặc trưng của giải thuật, đầu vào (Input) có nghĩa là gì? | |
| A | Là các giá trị cần đưa vào khi thuật toán thực hiện xong |
| B | Là các giá trị cần đưa vào khi viết chương trình |
| C | Là các giá trị cần đưa vào khi thuật toán bắt đầu làm việc |
| D | Là các giá trị đưa vào khi chương trình chạy |

|  |  |
| --- | --- |
| Trong các đặt trưng của giải thuật, đầu ra (Output) có nghĩa là gì? | |
| A | Là các dòng thông báo hiển thị trên màn hình |
| B | Là kết quả của sự thực hiện thuật toán |
| C | Là các dòng thông báo in ra máy in |
| D | Là kết quả trung gian khi thực hiện giải thuật |

|  |  |
| --- | --- |
| Trong các đặc trưng của giải thuật, tính dừng nghĩa là gì? | |
| A | Với mọi bộ dữ liệu vào thỏa mãn các điều kiện của dữ liệu, thuật toán phải dừng lại sau một số hữu hạn bước thực hiện |
| B | Với mọi bộ dữ liệu ra thỏa mãn các điều kiện của dữ liệu, thuật toán dừng lại sau thời gian chạy nhất định |
| C | Với mọi bộ dữ liệu ra thỏa mãn các điều kiện của dữ liệu, thuật toán phải dừng lại sau một số hữu hạn bước thực hiện |
| D | Với mọi bộ dữ liệu thỏa mãn các điều kiện của dữ liệu, thuật toán dừng lại sau một thời gian chạy nhất định |

|  |  |
| --- | --- |
| Trong các đặc trưng của giải thuật, tính xác định nghĩa là gì? | |
| A | Phải xác định rõ dữ liệu đầu vào và dữ liệu đầu ra của thuật toán |
| B | Mỗi bước của thuật toán có thể có nhiều cách hiểu khác nhau |
| C | Phải xác định chính xác từng dòng lệnh khi thực hiện chương trình |
| D | Mỗi bước của thuật toán cần được mô tả một cách chính xác và chỉ có một cách hiểu duy nhất |

|  |  |
| --- | --- |
| Trong các đặc trưng của giải thuật, tính khả thi có nghĩa là gì? | |
| A | Các phép toán có mặt trong các bước của thuật toán phải đơn giản, có ít nhất về nguyên tắc để có thể thực hiện được trong thời gian hữu hạn |
| B | Chương trình phải thực hiện được trên tất cả các máy tính |
| C | Với mọi bộ dữ liệu đầu vào, thỏa mãn các điều kiện của dữ liệu, thuật toán dừng lại sau một số hữu hạn bước thực hiện |
| D | Chương trình phải thực hiện được ở mọi nơi, mọi thời điểm |

|  |  |
| --- | --- |
| Trong một đơn vị có 20 nhân viên, mỗi nhân viên phải đóng 200000đ tiền bảo hiểm trong một tháng. Sử dụng kiểu dữ liệu nào trong các kiểu dữ liệu dưới đây để lưu trữ tổng tiền quĩ là tối ưu nhất? | |
| A | int |
| B | float |
| C | longint |
| D | unsigned int |

|  |  |
| --- | --- |
| Khi lưu trữ điểm thi cuối kỳ một môn học (theo thang điểm 0, 1, .., 10) kiểu dữ liệu nào dưới đây gây lãng phí nhất? | |
| A | char |
| B | int |
| C | longint |
| D | float |

|  |  |
| --- | --- |
| Trong các đặc trưng của giải thuật, tính khả thi nghĩa là gì? | |
| A | Các phép toán có mặt trong các bước của thuật toán phải đủ đơn giản, có ít nhất về nguyên tắc để có thể thực hiện được trong thời gian hữu hạn |
| B | Chương trình phải thực hiện được trên tất cả các máy tính |
| C | Chương trình phải thực hiện được ở mọi nơi, mọi thời điểm |
| D | Trong thời gian hữu hạn, chương trình phải dừng và cho kết quả theo ý muốn |

|  |  |
| --- | --- |
| Trong một giải thuật, khi thay đổi cấu trúc dữ liệu thì giải thuật có cần thay đổi theo hay không? Tại sao? | |
| A | Không cần, vì thuật toán không phụ thuộc vào cấu trúc dữ liệu |
| B | Có, vì để tránh việc xử lý gượng ép, thiếu tự nhiên trên một cấu trúc không phù hợp |
| C | Có, vì cấu trúc dữ liệu và giải thuật có một mối quan hệ mật thiết với nhau |
| D | Có, vì thay đổi sẽ dễ dàng thực hiện các thao tác hơn |

|  |  |
| --- | --- |
| Trong các phương pháp biểu diễn sau phương pháp nào sử dụng các hình vẽ để biểu diễn các bước trong thuật toán | |
| A | Phương pháp liệt kê từng bước |
| B | Phương pháp biểu diễn bằng sơ đồ khối |
| C | Phương pháp sử dụng một ngôn ngữ lập trình |
| D | Cả 3 phương pháp liệt kê từng bước, biểu diễn bằng sơ đồ khối, biểu diễn bằng ngôn ngữ lập trình |

|  |  |
| --- | --- |
| Giải thuật phản ánh gì? | |
| A | Giải thuật phản ánh các phép xử lý |
| B | Giải thuật phản ánh các đối tượng xử lý |
| C | Giải thuật phản ánh các thông tin xử lý |
| D | Giải thuật phản ánh các bước thực hiện |

|  |  |
| --- | --- |
| Khi lựa chọn cấu trúc dữ liệu vấn đề nào cần quan tâm nhất? | |
| A | Dữ liệu nhập vào có phù hợp không |
| B | Những thao tác nào sẽ tác động đến nó |
| C | Có dễ xử lý không |
| D | Cấu trúc dữ liệu có tiết kiệm được tài nguyên hệ thống không |

|  |  |
| --- | --- |
| Đối tượng được xử lý của giải thuật là gì? | |
| A | Dữ liệu |
| B | Cấu trúc dữ liệu |
| C | Các phép toán trên dữ liệu |
| D | Kiểu dữ liệu |

|  |  |
| --- | --- |
| Trong các tiêu chí đánh giá cấu trúc dữ liệu dưới đây, tiêu chí nào quan trọng nhất? | |
| A | Phù hợp với các thao tác xử lý |
| B | Tiết kiệm tài nguyên hệ thống |
| C | Phản ánh đúng thực tế |
| D | Dễ dàng trong việc thao tác dữ liệu |

|  |  |
| --- | --- |
| Tại sao không sử dụng một ngôn ngữ lập trình cụ thể để mô tả giải thuật mà lại sử dụng giả mã (giả ngôn ngữ)? | |
| A | Vì phải phụ thuộc vào cấu trúc dữ liệu tiền định của ngôn ngữ |
| B | Vì phải luôn luôn tuân thủ theo các qui tắc chặt chẽ về mặt cú pháp của ngôn ngữ đó |
| C | Vì ngôn ngữ nào được chọn cũng không hẳn đã được mọi người ưa thích |
| D | Vì phải tuân theo qui tắc chặt chẽ về mặt cú pháp của ngôn ngữ, phụ thuộc vào cấu trúc dữ liệu tiền định và không phải ngôn ngữ nào được chọn cũng không hẳn đã được mọi người ưa thích. |

|  |  |
| --- | --- |
| Áp dụng giải thuật đệ qui, viết chương trình tính tổng S = 1 + 3 + … + (2n+1). Đoạn chương trình nào dưới đây là đúng? | |
| A | int S(int n)  {  for (i = 1; i<=n; i++)  if (n%2 !=0) S = S +n;  } |
| B | int S(int n)  {  if (n==1) S = 1;  else if (n%2 == 0) n = n-1;  else S = S(n-2) + n;  } |
| C | int S(int n)  {  if (n==1) S = 1;  else  if (n%2 != 0) n = n-1;  else S = S(n-2) + n;  } |
| D | int S(int n)  {  for (i = 1; i<=n; i++)  if (n%2 ==0) S = S +n;  } |

|  |  |
| --- | --- |
| Áp dụng giải thuật đệ qui, viết chương trình tính tổng S = 0 + 2 + 4 + … + 2n  Đoạn chương trình nào dưới đây là đúng? | |
| A | int S(int n)  {  for (i = 0; i<=n; i++)  if (n%2 !=0) S = S +n;  } |
| B | int S(int n)  {  if (n==0) S = 0;  else  if (n%2 == 0) n = n-1;  else S = S(n-2) + n;  } |
| C | int S(int n)  {  if (n==0) S = 0;  else  if (n%2 != 0) n = n-1;  else S = S(n-2) + n;  } |
| D | int S(int n)  {  for (i = 0; i<=n; i++)  if (n%2 ==0) S = S +n;  } |

|  |  |
| --- | --- |
| Trong các đoạn chương trình tính tổng S=1+2+…+n dưới đây, đoạn chương trình nào được thiết kế dạng đệ qui? | |
| A | int S(int n)  {  for (i=1; i<=n; i++)  s+ = i;  } |
| B | int S(int n)  {  S = (n\*(n+1))/2;  } |
| C | int S(int n)  {  if (n=1)  return(1);  else  return (S(n-1)+n);  } |
| D | int S(int n)  {  S(n-1) + n;  } |

|  |  |
| --- | --- |
| Trong đệ qui, phần suy biến (phần NEO) được dùng để làm gì? | |
| A | Quyết định tới tính hữu hạn dừng của lời giải |
| B | Quyết định xem thuật toán có tiếp tục thực hiện hay không |
| C | Để thể hiện tính đệ qui của giải thuật |
| D | Quyết định tới tính hữu hạn của lời giải |

|  |  |
| --- | --- |
| Trong định nghĩa đệ qui, phần đệ qui (phần qui nạp) được thực hiện khi nào? | |
| A | Khi bài toán quá đơn giản có thể giải trực tiếp mà không cần phải nhờ đến một thuật toán nào |
| B | Khi chương trình gọi đến các thủ tục khác |
| C | Khi bài toán không thể giải bằng phần neo, ta xác định những bài toán con và gọi đệ quy giải những bài toán con đó. |
| D | Khi cần giải những bài toán con |

|  |  |
| --- | --- |
| Trong định nghĩa đệ qui, phần suy biến (phần NEO) được thực hiện khi nào? | |
| A | Khi đã có lời giải của những bài toán con |
| B | Khi bài toán quá đơn giản có thể giải trực tiếp mà không cần phải nhờ đến một thuật toán nào |
| C | Khi phối hợp các lời giải con của bài toán |
| D | Khi chương trình gọi đến các thủ tục khác |

|  |  |
| --- | --- |
| Trong các ví dụ dưới đây, ví dụ nào KHÔNG phải là đệ qui? | |
| A | Hình ảnh khi soi một chiếc gương |
| B | Đặt hai chiếc gương cầu đối diện nhau |
| C | Nếu n = 0 thì n! = 1; nếu n > 0 thì n! = n.(n-1)! |
| D | a) 1 là 1 số tự nhiên.  b) x là số tự nhiên nếu x -1 là số tự nhiên |

|  |  |
| --- | --- |
| Khi giải một phương trình đệ quy không thuộc dạng phương trình tổng quát thì sử dụng phương pháp nào? | |
| A | Truy hồi, kí hiệu ô lớn |
| B | Đoán nghiệm, kí hiệu ô lớn |
| C | Ký hiệu ô lớn |
| D | Truy hồi, đoán nghiệm |

|  |  |
| --- | --- |
| Các bước để giải phương trình đệ quy là gì? | |
| A | Dùng phương pháp truy hồi, giải phương trình đệ quy, sau đó nghiệm của phương trình đệ quy chính là độ phức tạp của giải thuật |
| B | Dùng phương pháp đoán nghiệm, giải phương trình đệ quy, sau đó nghiệm của phương trình đệ quy chính là độ phức tạp của giải thuật |
| C | Giải phương trình đệ quy,thành lập phương trình đệ quy, sau đó nghiệm của phương trình đệ quy chính là độ phức tạp của giải thuật |
| D | Top of Form  Thành lập phương trình đệ quy, sau đó giải phương trình đệ quy, nghiệm của phương trình đệ quy chính là độ phức tạp của giải thuật |

|  |  |
| --- | --- |
| Kết quả nào đúng khi thực hiện giải thuật đệ qui sau đây?  long lt(int n)  { if (n==0) return 1;  else return (2\*lt(n-1);  } | |
| A | lt(12) = 2010 |
| B | lt(12) = 1024 |
| C | lt(7) = 720 |
| D | lt(6) = 64 |

|  |  |
| --- | --- |
| Nhược điểm nổi bật nhất của đệ qui là gì? | |
| A | Đôi khi tạo ra các tính toán thừa, không cần thiết do tính chất tự động gọi thực hiện thủ tục khi chưa gặp điều kiện dừng của đệ quy. |
| B | Độ phức tạp tính toán cao |
| C | Chương trình Khó hiểu, không rõ ràng |
| D | Khó thiết kế chương trình. |

|  |  |
| --- | --- |
| Khi nào không nên sử dụng đệ qui? | |
| A | Khi chương trình đơn giản |
| B | Khi chương trình quá lớn, không thể thiết kế được dưới dạng lặp |
| C | Khi chương trình có thể viết dưới dạng lặp hoặc dưới dạng các cấu trúc khác |
| D | Khi chương trình lặp vô hạn. |

|  |
| --- |
| Cho biểu thức sau: *a*n = *a*n-1 + 6*a*n-2 + 10n(-2)n - 3(-2)n-1, với a0 = 8, a1 = 5.  Hãy tính a3 (với n = 3)? |

|  |  |
| --- | --- |
| A | -139 |
| B | -83 |
| C | 83 |
| D | 139 |

|  |  |
| --- | --- |
| Một người gửi 100 triệu đồng vào một quĩ đầu tư vào ngày đầu của một năm. Ngày cuối cùng của năm người đó được hưởng hai khoản tiền lãi. Khoản thứ nhất là 20% tổng số tiền có trong tài khoản cả năm, khoản lãi thứ hai là 45% của tổng số tiền có trong tài khoản của năm trước đó.  Hàm đệ qui nào dưới đây tính số tiền có trong tài khoản vào cuối năm thứ n? | |
| A | int S(int n)  {  if (n==0) return 100;  if (n==1) return 120;  else S = 1.2 \* S(n-1) + 0.45 \* S(n-2);  } |
| B | int S(int n)  {  if (n==0) return 100;  if (n==1) return 120;  else S = 0.45 \* S(n-1) + 1.2 \* S(n-2);  } |
| C | int S(int n)  {  S0 = 100; S1 = 120;  for (i=2; i<=n; i++)  Sn = 0.45 \* S(i-1) + 1.2 \* S(i-2);  } |
| D | int S(int n)  {  S0 = 100; S1 = 120;  for (i=2; i<=n; i++)  Sn = 1.2 \* S(n-1) + 0.45 \* S(n-2);  } |

|  |  |
| --- | --- |
| Cho đoạn chương trình đệ qui sau:  double tinh(int x, int n)  { if (n==1) return pow(x,n);  else return tinh(x,n-1) + pow(x,n+1);  }  Hãy cho biết kết quả chạy chương trình với x = 2, n = 2? | |
| A | 11 |
| B | 10 |
| C | 12 |
| D | 13 |

|  |  |
| --- | --- |
| Giả sử số tôm hùm bị đánh bắt trong một năm bằng trung bình cộng số bị đánh bắt trong 2 năm trước đó. Giả sử số tôm hùm năm thứ nhất bị đánh bắt là 100000 con, năm thứ 2 là 300000 con.  Hàm đệ qui nào dưới đây tính số tôm hùm bị đánh bắt trong năm thứ n? | |
| A | S1 = 100000;  S2 = 300000;  for (i=1; i<=n; i++)  S = S(i-1) + S(i-2); |
| B | if (n==1) return 100000;  if (n==2) return 300000;  else return (S(n-1)+S(n-2)); |
| C | S1 = 100000;  S2 = 300000;  for (i=1; i<=n; i++)  S = (S(i-1) + S(i-2))/2; |
| D | if (n==1) return 100000;  if (n==2) return 300000;  else return (S(n-1)+S(n-2))/2; |

|  |  |
| --- | --- |
| Cho đoạn chương trình sau:  float Tinh(float n)  { if (n==1) return 0.5;  else return Tinh(n-1) + 1/(n\*(n+1));  }  Hãy cho biết kết quả chạy chương trình với n = 3? | |
| A | 0.75 |
| B | 0.67 |
| C | 0.5 |
| D | 0.8 |

|  |  |
| --- | --- |
| Giải thuật đệ qui nào dưới đây đếm số các xâu nhị phân có độ dài n và không có 2 số 0 liên tiếp? | |
| A | int F(int n)  {  if (n==1) return 2;  if (n==2) return 3;  else return F(n-1) + F(n-2);  } |
| B | int F(int n)  {  if (n<=2) return 3;  else return F(n-1);  } |
| C | int F(int n)  {  if (n==1) return 2;  if (n==2) return 3;  else return F(n-2);  } |
| D | int F(int n)  {  if (n==1) return 2;  else return F(n-1) + F(n-2);  } |

|  |  |
| --- | --- |
| Hàm đệ qui sau tính gì?  double S(int n)  {  if(n==1) return 1;  else if(n%2!=0)  return S(n-1);  else return n+S (n-2);  } | |
| A | 2 + 4 + ...+ 2n |
| B | 1 + 2 + ... + n |
| C | 1 + 3 + ... + (2n+1) |
| D | 1.2....4 |

|  |  |
| --- | --- |
| Hàm đệ qui sau tính gì?  double S(int n)  {  if(n<=0) return 0;  else if(n%2==0)  return S(n-1);  else return n+S (n-2);  } | |
| A | 1+2+...+n |
| B | 1+3+5+…+(2n+1) |
| C | 0+2+4+…+2n |
| D | 1.2....n |

|  |  |
| --- | --- |
| Cho đoạn chương trình sau:  int S(int n)  { if(n == 0) return 0;  else return S(n/10) + n % 10;  }  Hãy cho biết kết quả chạy chương trình với n = 54782? | |
| A | 25 |
| B | 26 |
| C | 27 |
| D | 28 |

|  |  |
| --- | --- |
| Hàm đệ qui sau tính gì?  float S(float x , int n)  { if(n == 0) return 1;  if(n < 0) return S(x,n+1) \* 1/x;  else return S(x,n-1) \* x;  } | |
| A | 1+2+…+n |
| B | n! |
| C | n2 |
| D | xn |

|  |  |
| --- | --- |
| Hàm đệ qui sau tính gì?  int F(int n)  {  if(n<=2) return 1;  else return F(n-1)+F(n-2);  } | |
| A | Tìm dãy số Fibonacci |
| B | Tính tổng các số <=n-2 |
| C | Tính tổng các số chẵn |
| D | Tính tổng các số lẻ |

|  |  |
| --- | --- |
| Hàm đệ qui sau tính gì?  int Tinh(int n)  {  if (n==1) return 1;  else return Tinh(n-1) + n\*n;  } | |
| A | Tính tổng các số chẵn |
| B | Tính dãy số Fibonacci |
| C | Tính tổng các số lẻ |
| D | Tính tổng bình phương các số từ 1 đến n |

|  |  |
| --- | --- |
| Hàm đệ qui sau tính gì?  void S(long n)  {  if(n==0)return;  else  {  printf(“%d”, k%10);  S(k/10);  }  } | |
| A | Tìm số nguyên tố |
| B | In đảo ngược dãy số |
| C | Tìm số dư |
| D | Đổi số nguyên từ hệ 10 sang hệ 2 |

|  |  |
| --- | --- |
| Cho đoạn chương trình sau:  if(k==0) return;  else  { printf(“%d”, k%10);  S(k/10);  }  Hãy cho biết kết quả thực hiện của đoạn chương trình trên với k = 56978? | |
| A | 78 |
| B | 978 |
| C | 87965 |
| D | 965 |

|  |  |
| --- | --- |
| Trong các giải thuật đảo ngược một dãy số nguyên dưới đây, giải thuật nào được viết bằng phương pháp đệ qui? (ví dụ: 12345 🡪 54321) | |
| A | if(n==0)  {  printf(“%d”, n/10);  DaoNguoc(n%10);  } |
| B | if(n==0) return;  else  {  printf(“%d”, n%10);  DaoNguoc(n/10);  } |
| C | if(n==0) return;  else  {  printf(“%d”, n/10);  DaoNguoc(n%10);  } |
| D | if(n==0)  {  printf(“%d”, n%10);  DaoNguoc(n/10);  } |

|  |  |
| --- | --- |
| Cho đoạn chương trình sau :  if(a==b) return a;  else  if(a>b) return S(a-b,b);  else return S(a,b-a);  Hãy cho biết kết quả thực hiện của đoạn chương trình trên với a = 45; b=60? | |
| A | 15 |
| B | 3 |
| C | 9 |
| D | 30 |

|  |  |
| --- | --- |
| Trong các đoạn chương trình dưới đây, đoạn chương trình nào KHÔNG tính USCLN của 2 số a và b? | |
| A | if (b==0) S= a;  else  S = S(b, a%b); |
| B | do  { r=a%b;  if (r!=0)  { a=b; b=r;  }  } while (r!=0);  return b; |
| C | do  { if (a==b) return a;  else  if (a>b) return S(a-b,b);  else return S(a,b-a);  } while (a>b); |
| D | if (a==b) return a;  else  if(a>b) return S(a-b,b);  else return S(a,b-a); |

|  |  |
| --- | --- |
| Khi nói một đối tượng là đệ qui, phát biểu nào dưới đây KHÔNG đúng? | |
| A | Nó được định nghĩa dưới dạng của chính nó hoặc một đối tượng khác cùng dạng với chính nó bằng quy nạp |
| B | Một đối tượng được gọi là đệ qui nếu nó bao gồm chính nó như một bộ phận |
| C | Nếu một bài toán T được thực hiện bằng lời giải của một bài toán T’ có dạng giống T |
| D | Nếu bài toán được tách thành nhiều bài toán nhỏ hơn |

|  |  |
| --- | --- |
| Cho đoạn chương trình sau :  if (n=1) return(1);  else  return (S(n-1)+n);  Hãy cho biết kết quả thực hiện của đoạn chương trình trên với n = 10? | |
| A | 25 |
| B | 50 E 55 |
| C | 45 |
| D | 60 |

|  |  |
| --- | --- |
| Trong các giải thuật dưới đây, giải thuật nào được viết dưới dạng đệ qui? | |
| A | int F(int i)  { for (i=1; i<=n; i++)  F = F(i - 1) + F(i - 2)  } |
| B | do  { r=p%q;  if (r!=0)  { p=q; q=r;  }  } while (r!=0);  return q; |
| C | for (i = 0; i<=n; i++)  if (n%2 ==0) S = S +n; |
| D | int S(int a, int b)  { if (b==0) S= a;  else  S = S(b, a%b);  } |

|  |  |
| --- | --- |
| Trong các giải thuật dưới đây, giải thuật nào tính USCLN của 2 số nguyên dương p và q bằng phương pháp đệ qui? | |
| A | int USCLN(int p, int q)  {  if (q==0) return (p);  else  return ( USCLN(p%q, q));  } |
| B | do  { if (q==0) return (p);  else  return ( USCLN(q, p%q));  } while (r!=0);  return q; |
| C | do  { r=p%q;  if (r!=0)  { p=q; q=r;  }  } while (r!=0);  return q; |
| D | int USCLN(int p, int q)  {  if (q==0) return (p);  else  return ( USCLN(q, p%q));  } |

|  |  |
| --- | --- |
| Áp dụng giải thuật đệ qui, viết đoạn chương trình tính xn. Giải thuật nào dưới đây là đúng? | |
| A | int F(int x, int n)  {  if (n= = 0) F = 1;  else            F = x \* F(x,n-1);  } |
| B | F = pow(x, n); |
| C | int F(int x, int n)  {  F = 1;  for (i=1; i<=n; i++)  F = F \* x;  } |
| D | int F(int x, int n)  {  for (i=1; i<=n; i++)  if (n= = 0) F = 1;  else            F = x \* F(n-1);  } |

|  |  |
| --- | --- |
| Áp dụng giải thuật đệ qui, viết chương trình tìm ước số lẻ lớn nhất của số nguyên dương n. Giải thuật đệ qui nào dưới đây là đúng?(Ví dụ: n = 100 ước số lẻ lớn nhất của 100 là 25) | |
| A | int UocLeMax(int n)  { if(n % 2 == 1) return n;  Else return UocLeMax(n/2);  } |
| B | int UocLeMax(int n)  { for(i=1; i<=n; i++)  UocLeMax = i/2;  } |
| C | int UocLeMax(int n)  { if(n==1) UocLeMax = 1;  else UocLeMax = i/2;  } |
| D | int UocLeMax(int n)  { UocLeMax = 0;  for(i=1; i<=n; i++)  UocLeMax = UocLeMax(i-1) + i/2;  } |

|  |  |
| --- | --- |
| Cho đoạn chương trình đệ qui dưới đây:  double tinh(int x, int n)  { if (n==1) return pow(x,n);  else return tinh(x,n-1) + pow(x,n-1);  }  Hãy cho biết kết quả chạy chương trình với n = 2, x = 2? | |
| A | 6 |
| B | 4 |
| C | 5 |
| D | 3 |

|  |  |
| --- | --- |
| Cho đoạn chương trình đệ qui sau:  double tong(int x, int n)  {  if (n==1) return pow(x,2\*n);  else return tong(x,n-1) + pow(x,2\*n);  }  Hãy cho biết kết quả khi chạy chương trình với x=2, n=2? | |
| A | 20 |
| B | 30 |
| C | 16 |
| D | 40 |

|  |  |
| --- | --- |
| Viết đoạn chương trình tính số Fibonacci, áp dụng giải thuật đệ qui. Giải thuật nào dưới đây là đúng? | |
| A | int F(int n)  { for (i=1; i<=n; i++)  if (n <=2) return (1);  else return (F(n - 1) + F(n - 2));  } |
| B | int F(int n)  { if (n <=2) return (1);  else return (F(n - 1) + F(n - 2));  } |
| C | int F(int n)  { for (i=1; i<=n; i++)  F = (i-1) + (i-2);  } |
| D | int F(int n)  { for (i=1; i<=n; i++)  F = F(i - 1) + F(i - 2);  } |

|  |  |
| --- | --- |
| Cho đoạn chương trình sau, hãy cho biết kết quả chạy chương trình với n = 4?  float Tinh(float n)  {  if (n==1)  return 1;  return Tinh(n-1) + 1/n;  } | |
| A | ~2.2833 |
| B | ~2.0833 |
| C | ~2.45 |
| D | ~2.59286 |

|  |  |
| --- | --- |
| Áp dụng giải thuật đệ qui, viết đoạn chương trình tính tổng bình phương của n số tự nhiên. Giải thuật nào dưới đây là đúng? | |
| A | int tinh(int n)  { if(n==1) return 1;  else return tinh(n-1) + n\*n;  } |
| B | int S(int n)  { S = 0;  for(i=1; i<=n; i++)  S = S + i\*i;  } |
| C | int S(int n)  { S = 0;  if(n==0) return 0;  else return S(n-1) + n\*n;  } |
| D | int S(int n)  { S = 0;  if(n==0) return 0;  for(i=1; i<=n; i++)  S = S + i\*i;  } |

|  |  |
| --- | --- |
| Cho hàm đệ qui sau:  long xn(int n)  { if (n==0) return 1;  else return xn(n-1) + yn(n-1);  }  long yn(int n)  { if (n==0) return 0;  else 3\*xn(n-1) + 2\*yn(n-1);  }  Hãy cho biết kết quả chạy chương trình với n=2? | |
| A | x2 = 4; y2 = 9 |
| B | x2 = 1; y2 = 3 |
| C | x2 = 2; y2 = 6 |
| D | x2 = 1; y2 = 0 |

|  |  |
| --- | --- |
| Một người gửi ngân hàng 1000USD vào tài khoản của mình tại ngân hàng với lãi suất kép 8% mỗi năm. Giải thuật đệ qui nào dưới đây tính tổng tiền trong tài khoản của anh ta sau n năm? | |
| A | int F(int n)  { if(n==0) return 1000;  else return pow(1.08, n)\*F(0);  } |
| B | int F(int n)  { for(i=1; i<=n; i++)  if(n==0) return 1000;  else return 1.08\*F(n-1);  } |
| C | int F(int n)  { if(n==0) return 1000;  else return 1.08\*F(n-1);  } |
| D | int F(int n)  { for(i=1; i<=n; i++)  F=1.08\*F(n-1);  } |

|  |  |
| --- | --- |
| Vi trùng cứ 1 giờ lại nhân đôi. Vậy sau n giờ sẽ có mấy con vi trùng nếu ban đầu có 2 con. Giải thuật đệ qui nào dưới đây tính số vi trùng sau n giờ? | |
| A | int V(int n)  { if(n ==0) return 2;  else 2\*V(n-1);  } |
| B | int V(int n)  { for(i=1; i<=n; i++)  if(n == 0) return 2;  else return 2\*V(n-1);  } |
| C | int V(int n)  { if(n == 0) return 2;  else 2\*V(n-1) + 2;  } |
| D | int V(int n)  { for(i=1; i<=n; i++)  V(n) = 2\*V(n-1);  } |

|  |  |
| --- | --- |
| Áp dụng giải thuật đệ qui, viết đoạn chương trình tính tổng n số tự nhiên. Giải thuật nào dưới đây là đúng? | |
| A | int S(int n)  { S = 0;  for(i=0; i<=n; i++)  S =S + i;  } |
| B | int S(int n)  { S = 0;  for(i=0; i<=n; i++)  if(n==0) return 0;  else S = S + i;  } |
| C | int S(int n)  { for(i=0; i<=n; i++)  S = S(i-1) + i;  } |
| D | int S(int n)  { if(n == 0) return 0;  else S(n-1) + n;  } |

|  |  |
| --- | --- |
| Đoạn chương trình đệ qui sau tính gì?  int gt(int n)  { if(n==0) return 1;  else return (n\*gt(n-1));  } | |
| A | Tính dãy số Fibonacci |
| B | Tính giai thừa |
| C | Tìm ước số chung lớn nhất |
| D | Tính số lần chuyển của bài toán tháp Hà Nội |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức T, nếu phép duyệt tiền tố cho kết quả: \* 2 + 4 + 5 3; phép duyệt hậu tố cho kết quả là: 2 4 5 3 + + \*. Hãy cho biết các nút của cây con trái? | |
| A | 2 |
| B | 2 4 5 3 |
| C | 2 + 4 |
| D | 2 + 4 + 5 |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức T, nếu phép duyệt tiền tố cho kết quả: \* 2 + 4 + 5 3; phép duyệt trung tố cho kết quả là: 2 \* 4 + 5 + 3. Hãy cho biết các nút của cây con phải? | |
| A | 2 \* 4 |
| B | 5 + 3 |
| C | 4 5 + 3 |
| D | 4 + 5 + 3 |

|  |  |
| --- | --- |
| Cho cây biểu thức sau:  Biểu thức nào dưới đây tương ứng với cây? | |
| A | ((a2 - 2 )\* c) + 3 |
| B | (a2 - 2 )\* (c + 3) |
| C | (a - 2)2 \* (c + 3) |
| D | (a - 2)2 + (c \* 3) |

|  |  |
| --- | --- |
| Cho cây biểu thức sau:  Hãy cho biết kết quả duyệt cây theo thứ tự trước (tiền tố)? | |
| A | ^ - \* a 2 2 + c 3 |
| B | a 2 2 c 3 ^ - + \* |
| C | \* - ^ a 2 2 + c 3 |
| D | \* - ^ + a 2 2 c 3 |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức như hình vẽ:  Hãy cho biết kết quả duyệt cây theo thứ tự giữa (trung tố)? | |
| A | a ^ 2 - 2 c 3 \* + |
| B | a 2 - 2 ^\* c + 3 |
| C | a 2 ^- 2 c \*+ 3 |
| D | a ^ 2 - 2 \* c + 3 |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức như hình vẽ:  Hãy cho biết kết quả duyệt cây theo thứ tự sau (hậu tố)? | |
| A | a 2 2 ^ - c 3 + \* |
| B | a 2 ^ 2 - c 3 + \* |
| C | a 2 2 - c 3 + \*^ |
| D | a 2 2 c 3 + \*-^ |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức sau:  Biểu thức nào dưới đây tương ứng với cây nhị phân trên? | |
| A | (x2 - 4x + 7) |
| B | (x2 -4) \* (x + 7) |
| C | (x - 4)2 \* (x + 7) |
| D | (x - 4)2 + 7x |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức sau:  Hãy cho biết kết quả duyệt cây theo thứ tự trước (tiền tố)? | |
| A | + - ^ x 2 \* 4 x 7 |
| B | + - ^ \* x 2 4 x 7 |
| C | + - x 2 ^ \* 4 x 7 |
| D | + - ^ x 2 4 x \* 7 |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức sau:  Hãy cho biết kết quả duyệt cây theo thứ tự giữa (trung tố)? | |
| A | x ^ 2 - 4 x \* + 7 |
| B | x ^ 2 - 4 \* x + 7 |
| C | x ^ -2 4 x \*+ 7 |
| D | x ^ 2 - 4 x 7 \* + |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức sau:  Hãy cho biết kết quả duyệt cây theo thứ tự sau (hậu tố)? | |
| A | x 2 4 x 7 ^ \* - + |
| B | x 2 4 x ^ \* - 7 + |
| C | x 2 ^ 4 x \* - + 7 |
| D | x 2 ^ 4 x \* - 7 + |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức T, nếu phép duyệt tiền tố cho kết quả: + - ^ x 2 \* 4 x 7; phép duyệt trung tố cho kết quả là: x ^ 2 - 4 \* x + 7. Hãy cho biết các nút của cây con phải? | |
| A | 7 |
| B | 4 \* x + 7 |
| C | x + 7 |
| D | x 7 |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức T, nếu phép duyệt hậu tố cho kết quả: x 2 ^ 4 x \* - 7 +; phép duyệt trung tố cho kết quả là: x ^ 2 - 4 \* x + 7. Hãy cho biết các nút của cây con trái? | |
| A | X |
| B | x ^ 2 – 4 |
| C | x ^ 2 - 4 \* x |
| D | x ^ 2 |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức T, nếu phép duyệt tiền tố cho kết quả: \* - ^ a 2 2 + c 3; phép duyệt trung tố cho kết quả là: a ^ 2 - 2 \* c + 3. Hãy cho biết các nút của cây con phải? | |
| A | \* c + 3 |
| B | c + 3 |
| C | 3 |
| D | 2 \* c + 3 |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức T, nếu phép duyệt tiền tố cho kết quả: \* - ^ a 2 2 + c 3; phép duyệt hậu tố cho kết quả là: a 2 ^ 2 - c 3 + \*. Hãy cho biết các nút của cây con trái? | |
| A | a ^ 2 – 2 |
| B | a ^ 2 |
| C | a ^ 2 - 2 \* |
| D | a |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức T, nếu phép duyệt hậu tố cho kết quả: a 2 ^ 2 - c 3 + \*; phép duyệt trung tố cho kết quả là: a ^ 2 - 2 \* c + 3. Hãy cho biết nút của gốc? | |
| A | \* |
| B | - |
| C | ^ |
| D | + |
| Khi vẽ cây nhị phân dưới đây thành cây nhị phân nối vòng, những nút nào trên cây có mối nối không trỏ đến đầu cây (Head)? | |
| A | 9, 15, 87, 19, 5 |
| B | 15, 87, 5 |
| C | 9, 5, 19 |
| D | 9, 19 |

|  |  |
| --- | --- |
| Cấp (hay còn gọi là bậc) của 1 nút trong cây bằng bao nhiêu? | |
| A | Số các con của một nút |
| B | Số các con của một nút -1 |
| C | Số các con của một nút +1 |
| D | 1 |

|  |  |
| --- | --- |
| Cấp của cây bằng bao nhiêu? | |
| A | 1 |
| B | Số các con của nút |
| C | Cấp cao nhất của nút trên cây |
| D | Cấp bé nhất của nút trên cây |

|  |  |
| --- | --- |
| Nút lá có cấp bằng bao nhiêu? | |
| A | 0 |
| B | 1 |
| C | 2 |
| D | 3 |

|  |  |
| --- | --- |
| Nếu nút cha có mức là i thì nút con có mức là bao nhiêu? | |
| A | i -1 |
| B | i +1 |
| C | i + 2 |
| D | i -2 |

|  |  |  |
| --- | --- | --- |
| Cho cây sau:  Những nút nào là anh em của nút C? | | |
| A | A, B, D, E | |
| B | B, C, D, E | |
| C | A, H | |
| D | B, D, E | |
| Cho cây sau:  Các nút nào là nút lá? | |
| A | F, G, H, K, L, M | |
| B | B, C, D, E | |
| C | E, F, G, H, K, L, M | |
| D | E, F, G, H | |
| Cho cây sau:  Cha của nút I là nút nào? | | |
| A | A | |
| B | D | |
| C | K, L, M | |
| D | H | |

|  |  |
| --- | --- |
| Cho cây sau  Mức của B là bao nhiêu? | |
| A | 1 |
| B | 2 |
| C | 3 |
| D | 4 |
| Cho cây sau  Mức của cây bằng bao nhiêu? | |
| A | 1 |
| B | 2 |
| C | 3 |
| D | 4 |

|  |  |
| --- | --- |
| Cho cây sau:  Có bao nhiêu đường đi có độ dài bằng 3 trong cây? | |
| A | 2 |
| B | 3 |
| C | 4 |
| D | 0 |

|  |  |
| --- | --- |
| Cho cây sau:  Cấp của D bằng bao nhiêu? | |
| A | 1 |
| B | 2 |
| C | 3 |
| D | 4 |
| Cho cây sau:  Cấp của cây bằng bao nhiêu? | |
| A | 2 |
| B | 3 |
| C | 4 |
| D | 5 |

|  |  |
| --- | --- |
| Khi cài đặt cây, lệnh nào dưới đây có chức năng khởi tạo cây rỗng? | |
| A | T.MaxNode == 0; |
| B | (\*T).MaxNode=0; |
| C | T.MaxNode = 0; |
| D | (\*T).MaxNode==0; |

|  |  |
| --- | --- |
| Khi cài đặt cây, lệnh nào dưới đây có chức năng kiểm tra cây rỗng? | |
| A | T.MaxNode == 0; |
| B | (\*T).MaxNode=0; |
| C | T.MaxNode = 0; |
| D | (\*T).MaxNode==0; |

|  |  |
| --- | --- |
| Khi duyệt cây theo thứ tự trước, nút nào trong cây được duyệt đầu tiên? | |
| A | Nút bên phải cùng |
| B | Nút bên trái cùng |
| C | Nút giữa |
| D | Nút gốc |

|  |  |
| --- | --- |
| Khi duyệt cây theo thứ tự giữa, nút nào trong cây được duyệt đầu tiên? | |
| A | Nút bên phải cùng |
| B | Nút bên trái cùng |
| C | Nút giữa |
| D | Nút gốc |

|  |  |
| --- | --- |
| Duyệt cây theo thứ tự sau được thực hiện theo qui tắc nào? | |
| A | Trái 🡪 Phải 🡪 Gốc |
| B | Trái 🡪 Gốc 🡪 Phải |
| C | Gốc 🡪 Trái 🡪 Phải |
| D | Gốc 🡪 Phải 🡪 Trái |

|  |  |
| --- | --- |
| Duyệt cây theo thứ tự trước được thực hiện theo qui tắc nào? | |
| A | Trái 🡪 Phải 🡪 Gốc |
| B | Trái 🡪 Gốc 🡪 Phải |
| C | Gốc 🡪 Trái 🡪 Phải |
| D | Gốc 🡪 Phải 🡪 Trái |

|  |  |
| --- | --- |
| Duyệt cây theo thứ tự giữa được thực hiện theo qui tắc nào? | |
| A | Trái 🡪 Phải 🡪 Gốc |
| B | Trái 🡪 Gốc 🡪 Phải |
| C | Gốc 🡪 Trái 🡪 Phải |
| D | Gốc 🡪 Phải 🡪 Trái |

|  |  |
| --- | --- |
| Khi duyệt cây theo thứ tự sau, nút nào trong cây được duyệt đầu tiên? | |
| A | Nút trái cùng |
| B | Nút phải cùng |
| C | Nút gốc |
| D | Nút lá |

|  |  |
| --- | --- |
| Cho cây sau:  Hãy cho biết kết quả duyệt cây theo thứ tự giữa (trung tố)? | |
| A | F B G A H C D I K L M E |
| B | F G H K L M I B C D E A |
| C | F G B A H C D K I L M E |
| D | F B G A C H D K I L M E |
| Cho cây sau:  Hãy cho biết kết quả duyệt cây theo thứ tự trước(tiền tố)? | |
| A | A B C D E E G H I K L M |
| B | A B F G C H D I K L M E |
| C | A B F G H C I K L M D E |
| D | A B F G H C D I K L M E |
| Cho cây sau:  Hãy cho biết kết quả duyệt cây theo thứ tự sau (hậu tố)? | |
| A | F B G H C K L M I D E A |
| B | F B G C H D K L M I E A |
| C | F G B H C K L M I D E A |
| D | F G B C H D K L M I E A |

|  |  |
| --- | --- |
| Cho cây sau:  Độ dài đường đi từ A đến I bằng bao nhiêu? | |
| A | 4 |
| B | 1 |
| C | 3 |
| D | 2 |

|  |  |
| --- | --- |
| Lưu trữ kế tiếp một cây 5 phân (mỗi nút chiếm 1 ô). Nút có địa chỉ 21 thì 5 con ở địa chỉ nào? | |
| A | 99, 100, 101, 102, 103 |
| B | 105, 106, 107, 108, 109 |
| C | 102, 103, 104, 105, 106 |
| D | 100, 101, 102, 103, 104 |

|  |  |
| --- | --- |
| Độ cao của cây gọi là gì? | |
| A | Số lượng nút của cây |
| B | Mức lớn nhất của cây |
| C | Cấp lớn nhất của cây |
| D | Số cây con của cây |

|  |  |
| --- | --- |
| Lưu trữ kế tiếp một cây 8 phân (mỗi nút chiếm 1 ô), con thứ 7 của nút 20 nằm ở ô nào? | |
| A | 158 |
| B | 160 |
| C | 161 |
| D | 159 |

|  |  |
| --- | --- |
| Lưu trữ kế tiếp một cây 4 phân (mỗi nút chiếm 1 ô), nút có địa chỉ 25 thì 4 con ở địa chỉ nào? | |
| A | 98, 99, 100, 101 |
| B | 99, 100, 101, 102 |
| C | 100, 101, 102, 103 |
| D | 97, 98, 99, 100 |

|  |  |
| --- | --- |
| Nút có Khóa lớn nhất trong cây nhị phân tìm kiếm khác rỗng là nút nào? | |
| A | Nút con bên phải nhất |
| B | Nút con bên trái nhất |
| C | Nút gốc |
| D | Tất cả các nút trên cây |

|  |  |
| --- | --- |
| Trong phép duyệt cây nhị phân có 24 nút theo thứ tự sau, nút gốc có thứ tự là bao nhiêu? | |
| A | Thứ 1 |
| B | Thứ 2 |
| C | Thứ 23 |
| D | Thứ 24 |

|  |  |
| --- | --- |
| Nút có Khóa nhỏ nhất trong cây nhị phân tìm kiếm khác rỗng là? | |
| A | Nút con bên phải nhất |
| B | Nút con bên trái nhất |
| C | Nút gốc |
| D | Tất cả các nút trên cây |

|  |  |
| --- | --- |
| Cây nhị phân khác rỗng là cây như thế nào? | |
| A | Mỗi nút (trừ nút lá) đều có hai nút con |
| B | Tất cả các nút đều có nút con |
| C | Mỗi nút có không quá 2 nút con |
| D | Tất cả các nút đều có nút cha |

|  |  |
| --- | --- |
| Cho cây biểu thức sau:  Biểu thức nào dưới đây tương ứng với cây trên? | |
| A | (4 \* (2 + (5 + 3)) |
| B | (2 \* (4 + (5 + 3)) |
| C | (2 \* (3 + (4 + 5)) |
| D | (2 \* (5 + (4 + 3)) |

|  |  |
| --- | --- |
| Cho cây nhị phân T, nếu duyệt cây theo thứ tự trước cho kết quả là ABDEHCFIGJ, duyệt cây theo thứ tự giữa cho kết quả là EDHBCAIFJG. Hãy cho biết các nút của cây con trái? | |
| A | B C D E H |
| B | I F J G |
| C | B D E H F |
| D | I F D H |

|  |  |
| --- | --- |
| Viết các phần tử của cây nhị phân tìm kiếm được tạo từ các nút có Khóa là các số nguyên 2, 10, 15, -5, -2, 13, -12 khi thực hiện phép duyệt cây theo thứ tự sau? | |
| A | -12 -2 -5 13 15 10 2 |
| B | -12 -2 -5 2 13 15 10 |
| C | -12 -2 -5 10 15 13 2 |
| D | -12 -2 -5 10 13 15 2 |

|  |  |
| --- | --- |
| Viết các phần tử của cây nhị phân tìm kiếm được tạo từ các nút có Khóa là các số nguyên 2, 10, 15, -5, -2, 13, -12 khi thực hiện phép duyệt cây theo thứ tự giữa? | |
| A | 12 -5 -2 2 10 13 15 |
| B | 2 -5 -12 -2 10 15 13 |
| C | -12 -2 -5 2 13 15 10 |
| D | 10 -2 -5 13 2 15 -12 |

|  |  |
| --- | --- |
| Viết các phần tử của cây nhị phân tìm kiếm được tạo từ các nút có Khóa là các số nguyên 2, 10, 15, -5, -2, 13, -12 khi thực hiện phép duyệt cây theo thứ tự trước? | |
| A | 13 -2 -5 2 15 -12 10 |
| B | -12 -2 -5 2 13 15 10 |
| C | -12 -2 -5 13 2 15 10 |
| D | 2 -5 -12 -2 10 13 15 |

|  |  |
| --- | --- |
| Cho cây nhị phân T, nếu duyệt cây theo thứ tự trước cho kết quả là ABDEHCFIGJ, duyệt cây theo thứ tự giữa cho kết quả là EDHBCAIFJG. Hãy cho biết các nút của cây con phải? | |
| A | B C D E H | |
| B | I F J G | |
| C | B D E H F | |
| D | I F G H | |

|  |  |
| --- | --- |
| Cho cây nhị phân T, nếu phép duyệt cây theo thứ tự trước cho kết quả là AHBDEFICGJ, duyệt cây theo thứ tự sau cho kết quả là DBEHIGJCFA Hãy cho biết các nút của cây con phải? | |
| A | I G E C F |
| B | J G C H F Bỏ |
| C | I F C G J |
| D | J C F I E |

|  |  |
| --- | --- |
| Cho cây nhị phân T, nếu phép duyệt cây theo thứ tự trước cho kết quả là AHBDEFICGJ, duyệt cây theo thứ tự sau cho kết quả là DBEHIGJCFA Hãy cho biết các nút của cây con trái? | |
| A | D B H C |
| B | D B C E |
| C | D B E I |
| D | D B E H |

|  |  |
| --- | --- |
| Cho cây nhị phân T, nếu phép duyệt cây theo thứ tự giữa cho kết quả là HDIBJEAFCG, duyệt cây theo thứ tự sau cho kết quả là HIDJEBFGCA. Hãy cho biết các nút của cây con trái? | |
| A | H D I B E J |
| B | H D I C E J |
| C | F C G I E D |
| D | F C G J E H |

|  |  |
| --- | --- |
| Cho cây nhị phân T, nếu phép duyệt cây theo thứ tự giữa cho kết quả là HDIBJEAFCG, duyệt cây theo thứ tự sau cho kết quả là HIDJEBFGCA. Hãy cho biết các nút của cây con phải? | |
| A | F C D |
| B | F C G |
| C | F C H |
| D | F H E |

|  |  |
| --- | --- |
| Cho cây nhị phân T được lưu trữ kế tiếp vào mảng, nút có địa chỉ 7 có 2 con ở địa chỉ nào? | |
| A | 8 và 9 |
| B | 10 và 11 |
| C | 12 và 13 |
| D | 14 và 15 |

|  |  |
| --- | --- |
| Cho cây nhị phân T được lưu trữ kế tiếp vào mảng, nút có địa chỉ 19 thì có nút cha ở địa chỉ nào? | |
| A | 17 |
| B | 8 |
| C | 9 |
| D | 18 |

|  |  |
| --- | --- |
| Cho cây nhị phân T được lưu trữ kế tiếp vào mảng, số nút ở mức 7 (nút gốc có mức 1) là bao nhiêu? | |
| A | 31 |
| B | 64 |
| C | 127 |
| D | 255 |

|  |  |  |
| --- | --- | --- |
| Cho cây nhị phân T có chiều cao là 6 (nút gốc có mức 1). Số nút tối đa của cây là bao nhiêu? | | |
| A | 90 |
| B | 31 |
| C | 125 |
| D | 63 |

|  |  |
| --- | --- |
| Nếu lưu trữ kế tiếp một cây nhị phân có chiều cao 8 thì phải dự trù bao nhiêu ô nhớ (nút gốc có mức 1, mỗi nút cần 1 ô nhớ)? | |
| A | 128 ô |
| B | 255 ô |
| C | 64 ô |
| D | 256 ô |

|  |  |
| --- | --- |
| Một cây nhị phân có chiều cao là 7, cây đó chỉ có 50 nút. Nếu lưu trữ kế tiếp thì lãng phí bao nhiêu ô (nút gốc có mức 1, mỗi nút chiếm 1 ô)? | |
| A | 15 ô |
| B | 70 ô |
| C | 25 ô |
| D | 77 ô |

|  |  |
| --- | --- |
| Nếu lưu trữ móc nối thì mỗi nút của cây nhị phân cần 2 khoảng để ghi địa chỉ 2 con. Cây có 72 nút. Vậy lãng phí bao nhiêu khoảng địa chỉ? | |
| A | 72 |
| B | 70 |
| C | 73 |
| D | 75 |

|  |  |
| --- | --- |
| Cây nhị phân T có 30 nút lá (không có con). Cây đó có bao nhiêu nút cấp 2 (có 2 con)? | |
| A | 15 |
| B | 31 |
| C | 30 |
| D | 29 |

|  |  |
| --- | --- |
| Cho cây nhị phân T có 70 nút cấp 2 (có 2 con).Cây đó có bao nhiêu nút lá ( không có con)? | |
| A | 36 |
| B | 35 |
| C | 71 |
| D | 70 |

|  |  |
| --- | --- |
| Cây 5 phân có nghĩa là gì? | |
| A | Cây đó có 5 nút |
| B | Nút có cấp lớn nhất là 5 |
| C | Cây có chiều cao là 5 |
| D | Mức có nhiều nút nhất là 5 |

|  |  |
| --- | --- |
| Muốn lưu trữ kế tiếp một cây 5 phân có chiều cao là 3 thì phải dự trữ bao nhiêu ô nhớ (mỗi nút chiếm 1 ô)? | |
| A | 250 |
| B | 125 |
| C | 31 |
| D | 45 |

|  |  |
| --- | --- |
| Lưu trữ kế tiếp một cây 5 phân có chiều cao 4 mà cây đó chỉ có 70 nút thì lãng phí bao nhiêu ô nhớ (mỗi nút chiếm 1 ô)? | |
| A | 86 |
| B | 400 |
| C | 371 |
| D | 370 |

|  |  |
| --- | --- |
| Lưu trữ móc nối 1 cây 5 phân thì mỗi nút cần 5 khoảng địa chỉ để ghi địa chỉ 5 con. Cây đó chỉ có 40 nút, vậy lãng phí bao nhiêu khoảng địa chỉ? | |
| A | 40 |
| B | 41 |
| C | 39 |
| D | 161 |
| Cho cây biểu thức sau:  Khi duyệt cây theo thứ tự giữa, biểu thức trung tố được viết dưới dạng nào dưới đây? | |
| A | 2 \* + 4 +5 3 |
| B | 2 \* 4 + 5 + 3 |
| C | \* 2 + 4 + 5 3 |
| D | 2 4 5 3 + + \* |
| Cho cây biểu thức sau:  Khi duyệt cây theo thứ tự trước, biểu thức tiền tố được viết dưới dạng nào dưới đây? | |
| A | 2 \* + 4 +5 3 |
| B | 2 \* 4 + 5 + 3 |
| C | \* 2 + 4 + 5 3 |
| D | 2 4 5 3 + + \* |

|  |  |
| --- | --- |
| Cho cây biểu thức sau:  Khi duyệt cây theo thứ tự sau, biểu thức hậu tố được viết dưới dạng nào dưới đây? | |
| A | 2 \* + 4 +5 3 |
| B | 2 \* 4 + 5 + 3 |
| C | \* 2 + 4 + 5 3 |
| D | 2 4 5 3 + + \* |
| Hãy cho biết thứ tự duyệt các đỉnh của cây có gốc a sau theo phương pháp duyệt tiền thứ tự? | |
| A | e b i f a c g d h |
| B | a b e f i c g d h |
| C | e i f b g c h d a |
| D | a b c d e f g h i |
| Hãy cho biết thứ tự duyệt các đỉnh của cây có gốc a sau theo phương pháp duyệt trung thứ tự? | |
| A | e b i f a c g d h |
| B | a b e f i c g d h |
| C | e i f b g c h d a |
| D | a b c d e f g h i |
| Hãy cho biết thứ tự duyệt các đỉnh của cây có gốc a sau theo phương pháp duyệt hậu thứ tự? | |
| A | e b i f a c g d h |
| B | a b e f i c g d h |
| C | e i f b g c h d a |
| D | a b c d e f g h i |

|  |  |
| --- | --- |
| Điền vào chỗ trống cho định nghĩa sau đây: Cây là một đồ thị…. | |
| A | đơn và liên thông |
| B | đơn, liên thông và vô hướng |
| C | đơn, liên thông, có hướng và không có chu trình |
| D | đơn, liên thông, vô hướng và không có chu trình |

|  |  |
| --- | --- |
| Hãy cho biết thứ tự duyệt các đỉnh của cây có gốc a sau theo phương pháp duyệt tiền thứ tự? | |
| A | a b d e g h c f i |
| B | d b g e h a f i c |
| C | d g h e b i f c a |
| D | a b c d e f g h i |

|  |  |
| --- | --- |
| Hãy cho biết thứ tự duyệt các đỉnh của cây có gốc a sau theo phương pháp duyệt trung thứ tự? | |
| A | a b d e g h c f i |
| B | d b g e h a f i c |
| C | d g h e b i f c a |
| D | a b c d e f g h i |
| Hãy cho biết thứ tự duyệt các đỉnh của cây có gốc a sau theo phương pháp duyệt hậu thứ tự? | |
| A | a b d e g h c f i |
| B | d b g e h a f i c |
| C | d g h e b i f c a |
| D | a b c d e f g h i |

|  |  |
| --- | --- |
| Có bao nhiêu đường đi nối các đỉnh trong cây? | |
| A | 1 |
| B | 2 |
| C | 3 |
| D | Nhiều đường đi |

|  |  |
| --- | --- |
| Mỗi nút trong cây (trừ nút gốc) có bao nhiêu nút cha? | |
| A | Nhiều nút cha |
| B | 1 |
| C | 2 |
| D | Không có nút cha, chỉ có các nút con |

|  |  |
| --- | --- |
| Trong cấu trúc cây, các nút nằm cùng cấp được gọi là các nút gì? | |
| A | Cha con |
| B | Nút lá |
| C | Nút anh em |
| D | Nút gốc |

|  |  |
| --- | --- |
| Trong cấu trúc cây, nút không có các nút con được gọi là nút gì? | |
| A | Nút anh em |
| B | Nút rỗng |
| C | Nút gốc |
| D | Nút lá |

|  |  |
| --- | --- |
| Độ dài đường đi trên cây được tính bằng công thức nào? | |
| A | Số nút trên đường đi + 1 |
| B | Số nút trên đường đi – 1 |
| C | Số nút trên đường đi |
| D | Số mức của cây |

|  |  |
| --- | --- |
| Thế nào là cây nhị phân đầy đủ? | |
| A | Là cây nhị phân mà mỗi nút không phải lá đều có đúng 2 nút con và các nút lá phải có cùng độ sâu |
| B | Là cây nhị phân mà mỗi nút không phải lá có nhiều nhất 2 nút con và các nút lá phải có cùng độ sâu |
| C | Là cây nhị phân mà các giá trị của Khóa bên trái < gốc < Khóa bên phải |
| D | Là cây nhị phân mà giá trị của gốc > hơn giá trị của các nút con |

|  |  |  |
| --- | --- | --- |
| Thế nào là cây nhị phân tìm kiếm? | | |
| A | Là cây nhị phân mà giá trị của gốc < hơn giá trị của các nút con |
| B | Là cây nhị phân mà các giá trị của Khóa bên trái > gốc > Khóa bên phải |
| C | Là cây nhị phân mà các giá trị của Khóa bên trái < gốc < Khóa bên phải |
| D | Là cây nhị phân mà giá trị của gốc > hơn giá trị của các nút con |

|  |  |  |
| --- | --- | --- |
| Khi cài đặt cây nhị phân bằng danh sách liên kết, lệnh nào dưới đây có chức năng khởi tạo cây rỗng? | | |
| A | root = = 0; |
| B | (\*root)=0; |
| C | root = = NULL; |
| D | (\*root)=NULL; |

|  |  |  |
| --- | --- | --- |
| Khi cài đặt cây nhị phân bằng danh sách liên kết, lệnh nào dưới đây có chức năng kiểm tra cây rỗng? | | |
| A | root != 0; |
| B | (\*root) !=0; |
| C | root == NULL; |
| D | (\*root)=NULL; |

|  |  |
| --- | --- |
| Khi cài đặt cây nhị phân bằng danh sách liên kết, lệnh nào dưới đây có chức năng tính số nút trên cây T? (Trong đó nb\_nodes(LeftChild(T)) là số lượng các nút của cây con trái; nb\_nodes(RightChild(T)) là số lượng các nút của cây con phải) | |
| A | 1+nb\_nodes(LeftChild(T))+ nb\_nodes(RightChild(T)); |
| B | nb\_nodes(LeftChild(T))+ nb\_nodes(RightChild(T)); |
| C | nb\_nodes(LeftChild(T))+ nb\_nodes(RightChild(T)) - 1; |
| D | 1 - nb\_nodes(LeftChild(T))+ nb\_nodes(RightChild(T)); |

|  |  |
| --- | --- |
| Trên cây nhị phân nối vòng, nếu có n nút thì có bao nhiêu móc nối không? | |
| A | N |
| B | n+1 |
| C | 2n+1 |
| D | n-1 |

|  |  |
| --- | --- |
| Tại sao gọi là cây nhị phân nối vòng? | |
| A | Thay mối không bằng một cung trỏ lên nút cha |
| B | Thay mối không bằng một cung trỏ lên đầu cây |
| C | Thay mối không bằng mối con trỏ đến một nút qui định |
| D | Ở các nút lá đều có các mối nối trỏ lên đầu cây |

|  |  |
| --- | --- |
| Giả sử có 8 đồng tiền vàng trong đó có 1 đồng vàng giả có trọng lượng nhỏ hơn 7 đồng còn lại. Nếu sử dụng cây quyết định xây dựng cân thăng bằng thì cần ít nhất bao nhiêu lần cân để xác định đồng vàng giả? | |
| A | 1 |
| B | 2 |
| C | 3 |
| D | 4 |

|  |  |
| --- | --- |
| Giả sử có 24 đồng tiền vàng trong đó có 1 đồng vàng giả có trọng lượng nhỏ hơn 23 đồng còn lại. Nếu sử dụng cây quyết định xây dựng cân thăng bằng thì cần ít nhất bao nhiêu lần cân để xác định đồng vàng giả? | |
| A | 5 |
| B | 4 |
| C | 3 |
| D | 2 |

|  |  |
| --- | --- |
| Cây biểu thức sau tương ứng với biểu thức toán học nào dưới đây? | |
| A | (a2 - b )3 -5 |
| B | 5- (a2 - b )3 |
| C | (a2 - b3) -5 |
| D | 5- (a2 - b3) |

|  |  |
| --- | --- |
| Biểu thức tiền tố nào dưới đây tương ứng với cây nhị phân biểu thức sau đây? | |
| A | a 2 b 3 5 ^ - ^ - |
| B | a 2 b 3 5 - ^ - ^ |
| C | ^ - ^- a 2 b 3 5 |
| D | - ^ - ^ a 2 b 3 5 |
| Biểu thức hậu tố nào dưới đây tương ứng với cây nhị phân biểu thức sau đây? | |
| A | a 2 ^ b3 ^ 5 - - |
| B | a 2 b ^ - 3 ^ 5 - |
| C | a 2 ^ b - 3 ^ 5 - |
| D | a 2 ^ b - 3 5 ^ - |
| Biểu thức trung tố nào dưới đây tương ứng với cây nhị phân biểu thức sau đây? | |
| A | a ^ 2 - b ^ 3 – 5 |
| B | a 2 ^- b ^ 3 – 5 |
| C | a ^ 2 - b ^ 3 5 - |
| D | a 2^ - b 3 ^ - 5 |

|  |  |
| --- | --- |
| Cây biểu thức sau tương ứng với biểu thức toán học nào dưới đây? | |
| A | (x2 - 4x + 73) |
| B | (x2 - 4x + 7)3 |
| C | (x2 - 4)\*x + 7)3 |
| D | (x2 - 4\*(x + 7))3 |

|  |  |
| --- | --- |
| Biểu thức tiền tố nào dưới đây tương ứng với  cây nhị phân biểu thức sau đây? | |
| A | ^ - ^ x 2 + \* 4 x + 3 |
| B | ^ - ^ + \*+ x 2 4 x 3 |
| C | ^ - ^ x 2 + \* +4 x 3 |
| D | ^ - ^ +x 2 \* +4 x 3 |
| Biểu thức hậu tố nào dưới đây tương ứng với cây nhị phân biểu thức sau đây? | |
| A | x 2 4 x ^ \* 7 + 3- ^ | |
| B | x 2 ^ 4 x \* 7 3+ - ^ | |
| C | x 2 ^ 4 x \* 7 + 3- ^ | |
| D | x 2 ^ 4 x \* 7 + - 3 ^ | |

|  |  |
| --- | --- |
| Biểu thức trung tố nào dưới đây tương ứng với  cây nhị phân biểu thức sau đây? | |
| A | x ^ - 2 4 \* x 7 + ^ 3 |
| B | x ^ 2 - 4 x \* + 7 ^ 3 |
| C | x ^ 2 - 4 \* x + 7 ^ 3 |
| D | x 2 ^ - 4 \* x 7 + ^ 3 |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức T, nếu phép duyệt cây theo thứ tự giữa cho kết quả là, x ^ 2 - 4 \* x + 7 ^ 3; duyệt cây theo thứ tự sau cho kết quả là x 2 ^ 4 x \* 7 + - 3 ^. Hãy cho biết các nút của cây con phải? | |
| A | 3 |
| B | 7 ^ 3 |
| C | 4 \* x + 7 |
| D | 4 \* x + 7 ^ 3 |

|  |  |
| --- | --- |
| Cho cây nhị phân biểu thức T, nếu phép duyệt cây theo thứ tự giữa cho kết quả là, x ^ 2 - 4 \* x + 7 ^ 3; duyệt cây theo thứ tự sau cho kết quả là x 2 ^ 4 x \* 7 + - 3 ^. Hãy cho biết các nút của cây con trái? | |
| A | x ^ 2 |
| B | x^2- 4 \* x |
| C | x^2- 4 \* x + 7 |
| D | x^2- 4 \* x + 7 - |

|  |  |
| --- | --- |
| Gốc của cây có mức bằng bao nhiêu? | |
| A | 0 |
| B | 1 |
| C | 2 |
| D | Tùy vào chiều cao của cây |

|  |  |
| --- | --- |
| Cho mảng 2 chiều: A=(aij) i là chỉ số hàng, j là chỉ số cột. Mảng A có 6 hàng, 6 cột. Lưu trữ liên tiếp mảng A ưu tiên hàng. Nếu phần tử a11 có địa chỉ 65, mỗi phần tử chiếm 4 ô thì phần tử a6,5 có địa chỉ là bao nhiêu? | |
| A | 201 |
| B | 200 |
| C | 181 |
| D | 180 |

|  |  |
| --- | --- |
| Cho mảng 2 chiều A=(a ij): i là chỉ số hàng, j là chỉ số cột. Mảng A có 8 hàng, 7 cột. Lưu trữ liên tiếp mảng A ưu tiên cột, nếu phần tử a11 có địa chỉ 155, mỗi phần tử chiếm 4 ô thì phần tử a6,4 có địa chỉ là bao nhiêu? | |
| A | 270 |
| B | 271 |
| C | 327 |
| D | 328 |

|  |  |
| --- | --- |
| Các phần tử trong danh sách được bố trí như thế nào? | |
| A | Các phần tử được bố trí rời rạc |
| B | Các phần tử được bố trí ngẫu nhiên |
| C | Các phần tử sắp xếp theo thứ tự tăng dần |
| D | Các phần tử được sắp tuyến tính |

|  |  |
| --- | --- |
| Điền vào dấu ba chấm cho phát biểu sau: Cho n là độ dài của danh sách, với n>1, khi đó ta nói phần tử ai sẽ…. | |
| A | “đi sau” phần tử ai+1. |
| B | “đi trước” phần tử ai-1. |
| C | “đi trước” phần tử ai+1. |
| D | là phần tử thứ i+1 của danh sách |

|  |  |
| --- | --- |
| Khi cài đặt danh sách bằng mảng, khai báo một danh sách như sau:  #define MaxLength ...  typedef ... ElementType;  typedef int Position;  typedef struct {  ElementType Elements[MaxLength];  Position Last;  } List;  trong đó, khai báo Position Last; được dùng để mô tả gì? | |
| A | Giữ độ dài của danh sách |
| B | Chỉ phần tử cuối cùng trong danh sách |
| C | Chỉ phần tử đầu tiên trong danh sách |
| D | Chỉ vị trí các phần tử trong danh sách |

|  |  |
| --- | --- |
| Trong các câu lệnh dưới đây, câu lệnh nào có chức năng khởi tạo danh sách rỗng? | |
| A | L.Last = NULL; |
| B | L.Last=0; |
| C | L->Last=0; |
| D | L 🡪 Last = NULL; |

|  |  |
| --- | --- |
| Trong các câu lệnh dưới đây, câu lệnh nào có chức năng kiểm tra danh sách rỗng? | |
| A | L.Last !=0; |
| B | L.Last=0; |
| C | L->Last=0; |
| D | L 🡪 Last != 0; |

|  |  |
| --- | --- |
| Trong các điều kiện dưới đây, điều kiện nào KHÔNG cho phép chèn thêm một phần tử vào mảng danh sách? | |
| A | Mảng rỗng |
| B | Mảng đầy |
| C | Phần tử chèn vào không đúng vị trí |
| D | Không tìm thấy vị trí cần chèn |

|  |  |
| --- | --- |
| Trong các điều kiện dưới đây, điều kiện nào KHÔNG cho phép xóa một phần tử khỏi mảng danh sách? | |
| A | Không tìm thấy phần tử cần xóa |
| B | Phần tử cần xóa có giá trị lớn hơn vị trí của mảng |
| C | Mảng đầy |
| D | Mảng rỗng |

|  |  |
| --- | --- |
| Đoạn chương trình dưới đây mô tả thuật toán chèn thêm 1 phần tử vào mảng danh sách:  1. for(Q=(L->Last-1)+1;Q>P-1;Q--)  1.1. L->Elements[Q]=L->Elements[Q-1];  2. L->Elements[P-1]=X;  3. L->Last++;  Câu lệnh 1 và 1.1 có chức năng là gì? | |
| A | Dời các phần tử từ vị trí p đến cuối danh sách sang phải 1 vị trí |
| B | Dời các phần tử từ vị trí p đến cuối danh sách sang trái 1 vị trí |
| C | Dời các phần tử từ vị trí p về đầu danh sách sang phải 1 vị trí |
| D | Dời các phần tử từ vị trí p về đầu danh sách sang trái 1 vị trí |

|  |  |
| --- | --- |
| Đoạn chương trình dưới đây mô tả thuật toán loại bỏ 1 phần tử ra khỏi mảng danh sách:  1. for(Q=P-1;Q<L->Last-1;Q++)  1.1. L->Elements[Q]=L->Elements[Q+1];  2. L->Last--;  Câu lệnh 1 và 1.1 có chức năng là gì? | |
| A | Dời các phần tử từ vị trí p+1 (chỉ số trong mảng là p) về đầu danh sách sang phải 1 vị trí |
| B | Dời các phần tử từ vị trí p+1 (chỉ số trong mảng là p) về đầu danh sách sang trái 1 vị trí |
| C | Dời các phần tử từ vị trí p+1 (chỉ số trong mảng là p) đến cuối danh sách sang trái 1 vị trí |
| D | Dời các phần tử từ vị trí p+1 (chỉ số trong mảng là p) đến cuối danh sách sang phải 1 vị trí |

|  |  |
| --- | --- |
| Cho mảng một chiều A=(a1,a2,...,an) được lưu trữ liên tiếp. Giả sử mỗi phần tử của mảng chiếm 5 ô và phần tử đầu tiên a1 có địa chỉ 17 thì phần tử a8 có địa chỉ là bao nhiêu? | |
| A | 51 |
| B | 52 |
| C | 55 |
| D | 60 |

|  |  |
| --- | --- |
| Khi chèn phần tử có nội dung x vào vị trí p của danh sách L, trường hợp nào p KHÔNG hợp lệ? | |
| A | p<L.last+1 hoặc p<1 |
| B | p> L.last+1 hoặc p<1 |
| C | p> L.last+1 hoặc p>1 |
| D | p< L.last+1 hoặc p>1 |

|  |  |  |
| --- | --- | --- |
| Khi chèn phần tử có nội dung x vào vị trí p của danh sách L, trường hợp nào vị trí của p là hợp lệ? | | |
| A | p<= L.last+1 và p<=1 |
| B | p>= L.last+1 và p<=1 |
| C | p>= L.last+1 và p>=1 |
| D | p<= L.last+1 và p>=1 |

|  |  |
| --- | --- |
| Khi xóa 1 phần tử có nội dung x tại vị trí p của danh sách L, trường hợp nào p KHÔNG hợp lệ? | |
| A | P<L.last hoặc p>1 |
| B | p>L.last hoặc p>1 |
| C | p>L.last hoặc p<1 |
| D | P<L.last hoặc p<1 |

|  |  |
| --- | --- |
| Khi xóa 1 phần tử có nội dung x tại vị trí p của danh sách L, trường hợp nào vị trí của p là hợp lệ? | |
| A | p<=L.last và p>=1 |
| B | p<=L.last và p<=1 |
| C | p>=L.last và p<=1 |
| D | p>=L.last và p>=1 |

|  |  |
| --- | --- |
| Đoạn chương trình dưới đây mô tả thuật toán chèn thêm 1 phần tử vào mảng danh sách:  B1. for(n=(L->Last-1)+1; n>p-1; n--)  B1.1. L->A[n]=L->A[n-1];  B2. //Đưa X vào vị trí p  ……………………… (1)  3. L->Last++;  Hãy chọn câu lệnh đúng vào vị trí (1) cho B2? | |
| A | L->Elements[p-1]=X; |
| B | L->Elements[p+1]=X; |
| C | X = L->Elements[p-1]; |
| D | X = L->Elements[p+1] |

|  |  |
| --- | --- |
| Đoạn chương trình dưới đây mô tả thuật toán loại bỏ 1 phần tử ra khỏi mảng danh sách:  B1. for(n=i-1;n<L->Last-1;n++)  B1.1. Gán giá trị ở vị trí n+1 cho n  ............................... (1)  B2. L->Last--;  Hãy chọn câu lệnh đúng vào vị trí (1) cho B1.1? | |
| A | L->Elements[n-1]=L->Elements[n]; |
| B | L->Elements[n]=L->Elements[n-1]; |
| C | L->Elements[n]=L->Elements[n+1]; |
| D | L->Elements[n+1]=L->Elements[n]; |

|  |  |
| --- | --- |
| Cho mảng một chiều A=(a1,a2,...,an) và được lưu trữ liên tiếp. Giả sử mỗi phần tử của mảng chiếm 4 ô và phần tử đầu tiên a1 có địa chỉ 11 thì phần tử a9 có địa chỉ là bao nhiêu? | |
| A | 52 |
| B | 43 |
| C | 41 |
| D | 37 |

|  |  |
| --- | --- |
| Cho một ma trận thưa, hàng 1 có 3 phần tử a11 , a12, a1,3. Từ hàng thứ 2 chỉ có 5 phần tử  ak,k-2 , ak,k-1; ak,k; ak,k+1, ak,k+2, hàng cuối cùng cũng chỉ có 3 phần tử : an,n-1, an,n , an,n+1  Hãy lưu trữ liên tiếp ưu tiên hàng của ma trận này thành một mảng một chiều: thí dụ a11 là b1; a12 là b2 ; a21 là b3 … Tính bk nếu phần tử ai,j  là a5,6? | |
| A | 21 |
| B | 22 |
| C | 23 |
| D | 24 |

|  |  |
| --- | --- |
| Trong các phát biểu về những ưu điểm của cấu trúc dữ liệu mảng, phát biểu nào dưới đây là đúng? | |
| A | Việc thêm bớt các phần tử trong mảng được thực hiện thường xuyên |
| B | Việc truy xuất và tìm kiếm các phần tử của mảng tương đối dễ dàng thông qua chỉ số của mảng |
| C | Mật độ sử dụng bộ nhớ của mảng là rời rạc. |
| D | Mảng là một tập hợp không cố định các phần tử có cùng kiểu dữ liệu |

|  |  |
| --- | --- |
| Trong các phát biểu về những nhược điểm của cấu trúc dữ liệu mảng, phát biểu nào dưới đây là đúng? | |
| A | Việc thêm bớt các phần tử trong mảng có nhiều Khó khăn do phải di dời các phần tử khác |
| B | Việc truy xuất và tìm kiếm các phần tử của mảng phải theo thứ tự tuần tự |
| C | Mật độ sử dụng bộ nhớ của mảng là rời rạc. |
| D | Mảng là một tập hợp không cố định các phần tử có cùng kiểu dữ liệu |

|  |  |
| --- | --- |
| Trong các phát biểu về những ưu điểm của cấu trúc dữ liệu mảng, phát biểu nào dưới đây là đúng? | |
| A | Việc bổ sung, loại bỏ các phần tử thường xuyên được tác động lên mảng |
| B | Việc truy xuất và tìm kiếm các phần tử của mảng được thực hiện tuần tự |
| C | Mật độ sử dụng bộ nhớ của mảng là tối ưu tuyệt đối. |
| D | Mảng là một tập hợp các phần tử linh hoạt về kích thước. |

|  |  |  |
| --- | --- | --- |
| Trong số các phép toán sau đây, phép toán nào không được dùng thường xuyên đối với mảng? | | |
| A | Tạo mảng |
| B | Bổ sung một phần tử vào mảng |
| C | Lưu trữ mảng |
| D | Tìm kiếm trên mảng |

|  |  |
| --- | --- |
| Trong số các phép toán sau đây, phép toán nào không được dùng thường xuyên đối với mảng? | |
| A | Loại bỏ 1 phần tử ra khỏi mảng |
| B | Tìm kiếm trên mảng |
| C | Lưu trữ mảng |
| D | Sắp xếp dữ liệu trên mảng |

|  |  |
| --- | --- |
| Khi nói về nhược điểm của mảng, phát biểu nào dưới đây là đúng? | |
| A | Truy cập vào các phần tử của mảng phải thực hiện tuần tự |
| B | Việc tạo lập các phần tử của mảng rất Khó khăn |
| C | Mật độ sử dụng bộ nhớ của mảng là rời rạc. |
| D | Không linh hoạt về kích thước. |

|  |  |  |
| --- | --- | --- |
| Trong các phát biểu dưới đây về những nhược điểm của mảng, phát biểu nào KHÔNG đúng? | | |
| A | Không linh hoạt về kích thước. |
| B | Mật độ sử dụng bộ nhớ của mảng là rời rạc. |
| C | Việc bổ sung các phần tử rất Khó khăn do phải di dời các phần tử khác |
| D | Việc loại bỏ các phần tử rất Khó khăn do phải di dời các phần tử khác |

|  |  |
| --- | --- |
| Trong các phát biểu dưới đây về những ưu điểm của mảng, phát biểu nào SAI? | |
| A | Là cấu trúc dữ liệu dễ sử dụng |
| B | Tốc độ truy cập các phần tử cao |
| C | Các thao tác bổ sung, loại bỏ các phần tử được thực hiện dễ dàng. |
| D | Mật độ sử dụng bộ nhớ của mảng là tối ưu tuyệt đối |

|  |  |
| --- | --- |
| Điểm nào dưới đây nói về sự khác nhau giữa mảng và danh sách? | |
| A | Khác với mảng, danh sách là một tập sắp thứ tự nhưng bao gồm một số biến động các phần tử |
| B | Khác với mảng, danh sách là một tập sắp thứ tự cố định số lượng phần tử |
| C | Khác với mảng, phép toán bổ sung, loại bỏ các phần tử không được tác động lên danh sách. |
| D | Mảng truy cập các phần tử theo thứ tự, còn danh sách truy cập ngẫu nhiên |

|  |  |
| --- | --- |
| Cho mảng 2 chiều: A=(aij) i là chỉ số hàng, j là chỉ số cột. Mảng A có 7 hàng, 9 cột. Lưu trữ liên tiếp mảng A ưu tiên hàng. Nếu phần tử a11 có địa chỉ 70, mỗi phần tử chiếm 5 ô thì phần tử a7,9 có địa chỉ là bao nhiêu? | |
| A | 461 |
| B | 460 |
| C | 390 |
| D | 380 |

|  |  |
| --- | --- |
| Cho mảng 2 chiều: A=(aij) i là chỉ số hàng, j là chỉ số cột. Mảng A có 7 hàng, 9 cột. Lưu trữ liên tiếp mảng A ưu tiên cột. Nếu phần tử a11 có địa chỉ 75, mỗi phần tử chiếm 5 ô thì phần tử a7,9 có địa chỉ là bao nhiêu? | |
| A | 385 |
| B | 395 |
| C | 345 |
| D | 346 |

|  |  |
| --- | --- |
| Cho mảng một chiều A=(a1,a2,...,an) được lưu trữ liên tiếp. Giả sử mỗi phần tử của mảng chiếm 3 ô và phần tử đầu tiên a1 có địa chỉ 23 thì phần tử a7 có địa chỉ là bao nhiêu? | |
| A | 52 |
| B | 15 |
| C | 41 |
| D | 70 |

|  |  |
| --- | --- |
| Cho mảng 2 chiều: A=(aij) i là chỉ số hàng, j là chỉ số cột. Mảng A có 8 hàng, 9 cột. Lưu trữ liên tiếp mảng A ưu tiên hàng. Nếu phần tử a11 có địa chỉ 50, mỗi phần tử chiếm 3 ô thì phần tử a57 có địa chỉ là bao nhiêu? | |
| A | 148 |
| B | 152 |
| C | 158 |
| D | 176 |

|  |  |
| --- | --- |
| Cho mảng 2 chiều A=(a ij): i là chỉ số hàng, j là chỉ số cột. Mảng A có 8 hàng, 9 cột. Lưu trữ liên tiếp mảng A ưu tiên cột nếu phần tử a11 có địa chỉ 230, mỗi phần tử chiếm 3 ô thì phần tử a37 có địa chỉ là bao nhiêu? | |
| A | 378 |
| B | 380 |
| C | 382 |
| D | 390 |

|  |  |
| --- | --- |
| Cho mảng 2 chiều: A=(aij). Mảng có m hàng, n cột. Công thức tính địa chỉ của phần tử aij trong việc lưu trữ liên tiếp là  L(ai j) = L0 + C [(i – 1)n + (j – 1)]. Dùng trong trường hợp nào? | |
| A | Ưu tiên hàng |
| B | Ưu tiên cột |
| C | Ưu tiên đường chéo chính |
| D | Mọi trường hợp |

|  |  |
| --- | --- |
| Cho mảng 2 chiều A=(ai j), mảng có m hàng, n cột, và được lưu trữ liên tiếp. Công thức tính địa chỉ của phần tử a i j  L( ai j ) = L0 + C [(j – 1)m + (i – 1)]. Dùng trong trường hợp nào? | |
| A | Ưu tiên hàng |
| B | Ưu tiên cột |
| C | Ưu tiên đường chéo chính |
| D | Mọi trường hợp |

|  |  |
| --- | --- |
| Dùng phương pháp lưu trữ liên tiếp để lưu trữ một ma trận (mảng hai chiều) có nhược điểm lớn nhất là gì? | |
| A | Khó tìm kiếm |
| B | Cần một lượng lớn ô nhớ |
| C | Lãng phí ô nhớ khi ma trận thưa |
| D | Khó sắp xếp |

|  |  |
| --- | --- |
| Cho một ma trận thưa, hàng 1 có 2 phần tử a1,1 , a1,2. Từ hàng thứ 2 chỉ có 3 phần tử ak,k-1; ak,k; ak,k+1, hàng cuối cùng cũng chỉ có 2 phần tử : an,n-1; an,n. Hãy lưu trữ liên tiếp ưu tiên hàng của ma trận này thành một mảng một chiều: thí dụ a11 là b1; a1,2 là b2 ; a2,1 là b3 …Tính bk nếu phần tử ai,j  là a6,7? | |
| A | b17 |
| B | b18 |
| C | b19 |
| D | b20 |

|  |  |
| --- | --- |
| Biểu thức toán học ((A - B) /((C+D)\*(U -V))) viết dưới dạng hậu tố ký pháp Balan có dạng nào? | |
| A | A B C D U V - \* + / - |
| B | A B + C D U V - \* + / |
| C | A B - C D + / U V - \* |
| D | A B - C D + U V - \* / |

|  |  |
| --- | --- |
| Cho dãy thao tác: WEL\*LCO\*\*\*M\*. Với một chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào ngăn xếp, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong ngăn xếp in lên màn hình.  Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình? | |
| A | WLCOM |
| B | MOCLL |
| C | LOCLM |
| D | WELLC |

|  |  |
| --- | --- |
| Cho dãy thao tác: WEL\*LCO\*\*\*M\*. Với một chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào hàng đợi, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong hàng đợi in lên màn hình.  Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình? | |
| A | LOCLM |
| B | WELLC |
| C | MCOLL |
| D | OCLML |

|  |  |
| --- | --- |
| Biểu thức hậu tố ký pháp Balan có dạng: E=A B + U V - C \* / Với A=7; B=9; U= 7; V=5; C =2; E có giá trị là bao nhiêu? | |
| A | 2 |
| B | 4 |
| C | 5 |
| D | 8 |

|  |  |
| --- | --- |
| Biểu thức hậu tố ký pháp Balan có dạng E=10 4 - 2 \* 9 6 - 3 / + có giá trị là bao nhiêu? | |
| A | 11 |
| B | 13 |
| C | 19 |
| D | 7 |

|  |  |
| --- | --- |
| Biểu thức hậu tố ký pháp Balan có dạng E=10 5 - 3 \* 6 2 - 3 / + tương ứng với biểu thức toán học nào? | |
| A | (((10 - 5)\*3)+((6-2)/3)) |
| B | ((10 - (5\*3))+((6-2)/3)) |
| C | (((10 - 5)\*3)/((6-2)+3)) |
| D | ((10 - (5\*3))+(6-(2/3)) |

|  |  |
| --- | --- |
| Cho dãy thao tác: DHT\*\*N\*\*M\*\*THN Với một chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào hàng đợi, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong hàng đợi in lên màn hình. Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình? | |
| A | THNDM |
| B | HNTMNT |
| C | DHTNM |
| D | DHTNMT |

|  |  |
| --- | --- |
| Cho dãy thao tác: DHT\*\*N\*\*M\*\*THN Với một chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào ngăn xếp, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong ngăn xếp in lên màn hình.  Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình? | |
| A | THNDMT |
| B | HTDNM |
| C | HTDNMT |
| D | THNDM |

|  |  |
| --- | --- |
| Biểu thức hậu tố ký pháp Balan có dạng E= 8 1 3 + / 3 4 + 2 - \*có giá trị là bao nhiêu? | |
| A | 10 |
| B | 16 |
| C | 35 |
| D | 19 |

|  |  |
| --- | --- |
| Biểu thức hậu tố ký pháp Balan có dạng E= 8 1 3 + / 3 4 + 2 - \* tương ứng với biểu thức toán học nào? | |
| A | (((8/1)+3)\*((3+4)-2)) |
| B | (((8/(1+3))\*((3+4)-2)) |
| C | (((8/1)+(3\*3)+(4-2))) |
| D | (((8/(1+3))\*((3+(4-2)) |

|  |  |
| --- | --- |
| Biểu thức toán học E = (((5\*(3+4))-(2+(6/3)))tương ứng với biểu thức hậu tố ký pháp Balan nào? | |
| A | 5 3 4 + \* 2 6 3 + / - |
| B | 5 3 4 \* + 2 6 + 3 / - |
| C | 5 3 4 + \* 2 6 3 / + - |
| D | 5 3 4 \* + 2 6 3 - + / |

|  |  |
| --- | --- |
| Cho dãy thao tác: KH\*\*\*OA\*CNT\*T\* Với một chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào ngăn xếp, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong ngăn xếp in lên màn hình.  Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình? | |
| A | HKATTN |
| B | HKATT |
| C | KHOACN |
| D | KHOAC |

|  |  |
| --- | --- |
| Cho dãy thao tác: KH\*\*\*OA\*CNT\*T\* Với một chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào hàng đợi, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong hàng đợi in lên màn hình.  Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình? | |
| A | TTCNAO |
| B | TTCNA |
| C | KHOACN |
| D | KHOAC |

|  |  |
| --- | --- |
| Biểu thức hậu tố ký pháp Balan có dạng E= A B C - + U V + D \* / có giá trị là bao nhiêu? (Với A = 11; B=7; C = 2; U = 1; V=2; D=2) | |
| A | 3 |
| B | 2.57 E 2,66 |
| C | 2 |
| D | 5 |

|  |  |
| --- | --- |
| Biểu thức toán học E = (((A+B)-C)/((U+V)\*D))) tương ứng với biểu thức hậu tố ký pháp Balan nào? | |
| A | A B C - + U V + D \* / |
| B | A B + C - U V + D \* / |
| C | A B + C - U V D + \* / |
| D | A B C + - U V + D \* / |

|  |  |
| --- | --- |
| Biểu thức hậu tố ký pháp Balan có dạng E= A B C - + U V + D \* / tương ứng với biểu thức toán học nào? | |
| A | (((A+(B-C))/((U+V)\*D))) |
| B | (((A+B)-C)/((U+V)\*D))) |
| C | (((A+(B-C))/(U+(V\*D))) |
| D | ((A+B)-C))/((U+(V\*D))) |

|  |  |
| --- | --- |
| Dấu hiệu nào dưới đây cho biết hàng đợi đã có thao tác thêm và loại bỏ phần tử là rỗng? | |
| A | Lối sau nhận giá trị = 0 |
| B | Lối trước có giá trị < giá trị của lối sau |
| C | Lối trước có giá trị > giá trị của lối sau |
| D | Lối trước nhận giá trị = 0 |

|  |  |
| --- | --- |
| Thao tác nào dưới đây thực hiện trên ngăn xếp (stack)? | |
| A | Thêm phần tử vào vị trí bất kỳ |
| B | Loại bỏ phần tử tại vị trí bất kỳ |
| C | Thêm và loại bỏ phần tử luôn thực hiện tại vị trí đỉnh (top) |
| D | Thêm và loại bỏ phần tử có thể thực hiện tại vị trí bất kỳ |

|  |  |
| --- | --- |
| Kết quả của biểu thức dạng hậu tố E= 6 27 25 - \* 15 8 - 3\* - khi ứng dụng ngăn xếp để tính: E = bao nhiêu? | |
| A | E = 627 |
| B | E = -546 E -9 |
| C | E = 546 |
| D | E = -158 |

|  |  |
| --- | --- |
| Thao tác nào dưới đây đặc trưng trên hàng đợi (queue)? | |
| A | Thêm và loại bỏ phần tử tại vị trí bất kỳ |
| B | Thêm phần tử vào lối sau |
| C | Loại bỏ phần tử ở lối sau |
| D | Thêm phần tử vào lối trước |

|  |  |
| --- | --- |
| Cho một danh sách liên kết đơn lưu trữ các số nguyên, hãy xóa bỏ các phần tử chia hết cho 3 trong danh sách. Đoạn chương trình sau đây mô tả thuật toán trên:  while (r !=NULL)  { if (q 🡪 item %3 ==0) (1)  { ………………………………… ..(2)  …………………………………...(3)  ………………………………….. (4)  free(q); (5)  }  r = r 🡪 next; (6)  }  Hãy chọn câu lệnh đúng điền vào vị trí (2) (3) và (4)? | |
| A | q = r 🡪 next; (2)  r 🡪 next = q 🡪 next; (3)  q🡪 next = NULL; (4) |
| B | r = q🡪 next; (2)  r 🡪 next = q 🡪 next; (3)  q🡪 next = NULL; (4) |
| C | q = r 🡪 next; (2)  q 🡪 next = r 🡪 next; (3)  r🡪 next = NULL; (4) |
| D | q = r 🡪 next; (2)  r 🡪 next = q 🡪 next; (3)  r🡪 next = NULL; (4) |

|  |  |
| --- | --- |
| Ý tưởng của thuật toán xóa bỏ một phần tử có nội dung là x được trỏ tới bởi con trỏ q nào dưới đây là đúng? | |
| A | 1. Tìm địa chỉ của phần tử có nội dung là x trong danh sách liên kết.  2. Loại bỏ phần tử này theo địa chỉ. |
| B | 1. Tìm địa chỉ của phần tử có nội dung là x trong danh sách liên kết.  2. Ngắt liên kết của phần tử này |
| C | 1. Tìm địa chỉ của phần tử có nội dung là x trong danh sách liên kết.  2. Cho liên kết của nút này trỏ tới NULL |
| D | 1. Tìm địa chỉ của phần tử có nội dung là x trong danh sách liên kết.  2. Cho liên kết của nút này trỏ tới liên kết kế tiếp. |

|  |  |
| --- | --- |
| Dấu hiệu nào dưới đây cho biết danh sách liên kết đơn L là rỗng? | |
| A | (L->next == NULL) |
| B | (L == NULL) |
| C | (L->infor == NULL) |
| D | (L->left == NULL) |

|  |  |
| --- | --- |
| Khi nói về danh sách liên kết, phát biểu nào dưới đây là đúng? | |
| A | Danh sách là tập hợp các phần tử có kiểu dữ liệu xác định và giữa chúng có một mối liên hệ nào đó |
| B | Số phần tử của danh sách cộng 1 được gọi là chiều dài của danh sách |
| C | Một danh sách có chiều dài bằng -1 là danh sách rỗng |
| D | Tất cả các phương án trên đều đúng |

|  |  |
| --- | --- |
| Khi nói về danh sách liên kết, phát biểu nào dưới đây là SAI? | |
| A | Danh sách là tập hợp các phần tử có kiểu dữ liệu xác định và giữa chúng có một mối liên hệ nào đó |
| B | Số phần tử của danh sách được gọi là chiều dài của danh sách |
| C | Một danh sách có chiều dài bằng 0 là danh sách rỗng |
| D | Số phần tử của danh sách cộng 1 được gọi là chiều dài của danh sách |

|  |  |
| --- | --- |
| Cho một danh sách liên kết đơn lưu trữ các số nguyên, hãy xóa bỏ các phần tử âm trong danh sách. Đoạn chương trình sau đây mô tả thuật toán trên:  while (r !=NULL)  { ……………………… (1)  { q = r 🡪 next; (2)  r 🡪 next = q 🡪 next; (3)  q🡪 next = NULL; (4)  free(q); (5)  }  r = r 🡪 next; (6)  }  Hãy chọn câu lệnh đúng điền vào vị trí (1)? | |
| A | if (q 🡪 next <0) |
| B | while (q 🡪 next <0) |
| C | while (q🡪item <0) |
| D | if (q🡪 item<0) |

|  |  |
| --- | --- |
| Định nghĩa nào dưới đây là đúng với danh sách liên kết? | |
| A | Danh sách liên kết là cấu trúc dữ liệu dạng cây |
| B | Danh sách liên kết là cấu trúc dữ liệu tự định nghĩa |
| C | Danh sách liên kết là tập hợp các phần tử mà giữa chúng có một sự nối kết với nhau thông qua vùng liên kết của chúng |
| D | Danh sách liên kết là tập hợp các phần tử đặt kề cận với nhau trong vùng nhớ |

|  |  |  |
| --- | --- | --- |
| Khi nói về ưu điểm của danh sách liên kết, phát biểu nào dưới đây là đúng? | | |
| A | Việc bổ sung, loại bỏ các phần tử thường xuyên được tác động lên danh sách |
| B | Việc truy xuất và tìm kiếm các phần tử của danh sách được thực hiện ngẫu nhiên |
| C | Mật độ sử dụng bộ nhớ của danh sách là tối ưu tuyệt đối. |
| D | Danh sách là một tập hợp các phần tử cố định về về kích thước. |

|  |  |
| --- | --- |
| Thủ tục xóa bỏ một phần tử có nội dung là x trong danh sách liên kết đơn được mô tả như sau:  r = \*p;  ………………………….(1)  q = r 🡪 next;  r 🡪 next = q 🡪 next;  q 🡪 next = NULL;  free(q);  Hãy chọn câu lệnh đúng điền vào vị trí (1)? | |
| A | while ((r 🡪 next != NULL)&& (q🡪Item !=x))  r = r🡪 next; |
| B | while (q🡪Item !=x)  r = r🡪 next; |
| C | while (r 🡪 next != NULL)  r = r🡪 next; |
| D | while (r 🡪 next != NULL)&& (q🡪Item ==x)  r = r🡪 next; |

|  |  |
| --- | --- |
| Cho một danh sách liên kết đơn lưu trữ các số nguyên, kiểm tra xem ở vị trí thứ 5 trong danh sách có phải là phần tử chẵn hay không? Nếu đúng hãy xóa bỏ phần tử này được mô tả trong đoạn chương trình sau đây:  while (r !=NULL)  { if (count == 5 && q🡪 item%2==0) (1)  { q = r 🡪 next; (2)  r 🡪 next = q 🡪 next; (3)  q🡪 next = NULL; (4)  free(q); (5)  }  count ++; r = r 🡪 next; (6)  }  Lệnh nào trong các lệnh trên có chức năng tìm và kiểm tra điều kiện phần tử ở vị trí thứ 5 trong danh sách? | |
| A | (1), (2) |
| B | (1), (6) |
| C | (1) |
| D | (2), (6) |

|  |  |
| --- | --- |
| Định nghĩa cấu trúc dữ liệu của danh sách liên kết đơn được mô tả như sau:  struct node  { int item;  struct node \*next;  } ;  typedef struct node \*listnode;  Trong đó, khai báo node \* next; được dùng để mô tả gì? | |
| A | Con trỏ tới vùng dữ liệu |
| B | Vùng liên kết quản lý địa chỉ phần tử kế tiếp |
| C | Con trỏ trỏ tới vùng nhớ của phần tử trước đó trong danh sách liên kết đơn |
| D | Con trỏ trỏ tới vùng nhớ của phần tử đầu tiên trong danh sách liên kết đơn |

|  |  |  |
| --- | --- | --- |
| Với cấu trúc dữ liệu của DSLK đơn lưu trữ thông tin về phòng máy được cho như sau:  typedef struct PM  { int maPM;  int tongsoMay;  } PHONGMAY;  typedef struct Node  { PHONGMAY Data;  Node \* NextNode;  } OneNode;  typedef OneNode \*SLLPointer;  Để quản lý danh sách liên kết đơn bằng phần tử đầu và phần tử cuối, cần định nghĩa thêm kiểu dữ liệu nào dưới đây? | | |
| A | SLLPinter DanhSach; |
| B | typedef struct SSLLIST  { SLLPointer First;  SLLPointer Last;  } LIST;  LIST DanhSach; |
| C | typedef struct SSLLIST  { SLLPointer First;  SLLPointer Last;  int total;  } LIST;  LIST DanhSach; |
| D | typedef struct SSLLIST  { SLLPointer First;  int total;  } LIST;  LIST DanhSach; |

|  |  |
| --- | --- |
| Đoạn giả mã thuật toán bổ sung một phần tử có giá trị thành phần dữ liệu là NewData vào trong danh sách liên kết đơn SLList vào ngay sau nút có địa chỉ InsNode:  B1: NewNode -> Next = NULL;  B2: NewNode -> Data = NewData;  B3: if (InsNode -> Next = NULL)  B3.1: InsNode -> Next = NewNode;  B3.2: Thực hiện BKT;  B4: //Nối các nút kế sau InsNode vào sau NewNode  ………………………………….  B5: //Chuyển mối liên kết giữa InsNode với nút kế của nó về NewNode  ……………………………………..  BKT: Kết thúc  Hãy chọn câu lệnh đúng cho B6 và B7? | |
| A | B6: InsNode ->Next = NewNode ->Next  B7: NewNode = InsNode -> Next |
| B | B6: InsNode ->Next = NewNode ->Next  B7: InsNode -> Next = NewNode |
| C | B6: NewNode ->Next = InsNode ->Next  B7: NewNode = InsNode -> Next |
| D | B6: NewNode ->Next = InsNode ->Next  B7: InsNode -> Next = NewNode |

|  |  |  |
| --- | --- | --- |
| Với tổ chức dữ liệu cho 1 danh sách liên kết đơn như sau:  struct node  { int item;  struct node \*next;  };  typedef struct node \*listnode;  đoạn chương dưới đây bổ sung 1 phần tử có giá trị là x vào ngay sau nút r trong danh sách:  while ((r !=NULL) && (found == 0))  { q 🡪 item = x;  …………………………… (1)  …………………………….(2)  }  Hãy chọn câu lệnh đúng điền vào vị trí (1) và (2) | | |
| A | (1): r 🡪 next = q 🡪 next;  (2): r 🡪 next = q; |
| B | (1): r 🡪 next = q 🡪 next;  (2): q 🡪 next = r; |
| C | (1): q 🡪 next = r 🡪 next;  (2): r 🡪 next = q; |
| D | (1): q 🡪 next = r 🡪 next;  (2): q = r 🡪 next ; |

|  |  |
| --- | --- |
| Cho cấu trúc kiểu dữ liệu sinh viên như sau:  typedef struct TagSV  { char MSSV[8];  char Ten[30];  char Nsinh[11];  float ĐTB;  } SV;  SV sv1, \*sv2;  Để gán giá trị cho mã sinh viên của sv1 và sv2, câu lệnh nào dưới đây là đúng? | |
| A | sv1.MSSV = “DH0002001”  sv2.MSSV = “DH0002002” |
| B | sv1.MSSV = “DH0002001”  sv2->MSSV = “DH0002002” |
| C | sv1->MSSV = “DH0002001”  sv2.MSSV = “DH0002002” |
| D | sv1->MSSV = “DH0002001”  sv2->MSSV = “DH0002002” |

|  |  |
| --- | --- |
| Khi dùng danh sách liên kết có cần phải khai báo kích thước hay không? | |
| A | Phải khai báo trước khi sử dụng |
| B | Phải khai báo ở đầu chương trình |
| C | Có thể khai báo có thể không |
| D | Không cần phải khai báo trước |

|  |  |
| --- | --- |
| Lệnh nào dưới đây chỉ con trỏ đang ở cuối danh sách liên kết đơn L? | |
| A | L 🡪 Next = 0; |
| B | L 🡪 Next = NULL; |
| C | L. Next = 0; |
| D | L. Next = NULL; |

|  |  |
| --- | --- |
| Các nút của danh sách liên kết đơn được tạo ra khi nào? | |
| A | Chỉ được tạo ra khi cần. |
| B | Được tạo ra ở đầu danh sách |
| C | Được tạo ra ở cuối danh sách |
| D | Số lượng các phần tử trong danh sách là cố định, nên không thể tạo thêm. |

|  |  |
| --- | --- |
| Lệnh nào dưới đây có chức năng tạo và cấp phát bộ nhớ cho một nút trong danh sách liên kết đơn? | |
| A | \*p == NULL; |
| B | free(p); |
| C | p = (listnode) malloc(sizeof(struct node)); |
| D | p🡪 Next = NULL; |

|  |  |
| --- | --- |
| Lệnh nào dưới đây có chức năng giải phóng bộ nhớ đã cấp phát cho nút p trong danh sách liên kết đơn? | |
| A | p.Next = 0; |
| B | free(p); |
| C | p = (listnode) malloc(sizeof(struct node)); |
| D | p🡪 Next = NULL; |

|  |  |
| --- | --- |
| Chèn thêm 1 nút vào đầu danh sách liên kết đơn, thực hiện tuần tự theo các bước nào dưới đây? (giả sử con trỏ p đang trỏ vào đầu danh sách) | |
| A | 1. Cho p trỏ đến nút mới tạo.  2. Tạo và cấp phát bộ nhớ cho 1 nút  3. Cho con trỏ tiếp của nút mới trỏ đến phần tử đầu tiên danh sách |
| B | 1. Cho p trỏ đến nút mới tạo.  2. Tạo và cấp phát bộ nhớ cho 1 nút  3. Cho con trỏ tiếp của nút mới trỏ đến phần tử đầu tiên danh sách |
| C | 1. Tạo và cấp phát bộ nhớ cho 1 nút  2. Cho p trỏ đến nút mới tạo.  3. Cho con trỏ tiếp của nút mới trỏ đến phần tử đầu tiên danh sách |
| D | 1. Tạo và cấp phát bộ nhớ cho 1 nút  2. Cho con trỏ tiếp của nút mới trỏ đến phần tử đầu tiên danh sách  3. Cho p trỏ đến nút mới tạo. |

|  |  |
| --- | --- |
| Đoạn chương trình nào dưới đây bổ sung thêm 1 nút có giá trị x và danh sách liên kết p? | |
| A | q = (listnode) malloc(sizeof(struct node));  q 🡪 item = x;  q 🡪 next = \*p; |
| B | q 🡪 next = \*p;  q = (listnode) malloc(sizeof(struct node));  q 🡪 item = x; |
| C | q = (listnode) malloc(sizeof(struct node));  q 🡪 next = \*p;  q 🡪 item = x; |
| D | q 🡪 item = x;  q = (listnode) malloc(sizeof(struct node));  q 🡪 next = \*p; |

|  |  |
| --- | --- |
| Đoạn chương trình sau mô tả giải thuật chèn 1 nút vào cuối danh sách liên kết đơn:  if (\*p == NULL) \*p = q;  else {  r = \*p;  while (r🡪next != NULL) r = r 🡪 next;  // Cho con trỏ tiếp của nút cuối (được trỏ tới bởi r) trỏ đến nút mới tạo là q  ……………………………… (1)  // cho con trỏ tiếp của q trỏ tới NULL  …………………………. (2)  }  Hãy chọn câu lệnh đúng điền vào vị trí (1) và (2) | |
| A | (1) q 🡪 next = r;  (2) q 🡪 next = NULL; |
| B | (1) r 🡪 next = q;  (2) q 🡪 next = NULL; |
| C | (1) r 🡪 next = q 🡪 next;  (2) q 🡪 next = NULL; |
| D | (1) q 🡪 next = r 🡪 next;  (2) r 🡪 next = NULL; |

|  |  |
| --- | --- |
| Ý tưởng của giải thuật chèn một nút vào cuối danh sách liên kết đơn nào dưới đây là đúng? | |
| A | 1. Tạo và cấp phát bộ nhớ cho 1 nút mới  2. Dịch chuyển con trỏ tới nút cuối của danh sách  3. Cho con trỏ tiếp của nút cuối (được trỏ tới bởi r) trỏ đến nút mới tạo là q, và cho con trỏ tiếp của q trỏ tới NULL |
| B | 1. Dịch chuyển con trỏ tới nút cuối của danh sách  2. Tạo và cấp phát bộ nhớ cho 1 nút mới  3. Cho con trỏ tiếp của nút cuối (được trỏ tới bởi r) trỏ đến nút mới tạo là q, và cho con trỏ tiếp của q trỏ tới NULL |
| C | 1. Tạo và cấp phát bộ nhớ cho 1 nút mới  2. Cho con trỏ tiếp của nút cuối (được trỏ tới bởi r) trỏ đến nút mới tạo là q, và cho con trỏ tiếp của q trỏ tới NULL  3. Dịch chuyển con trỏ tới nút cuối của danh sách |
| D | 1. Cho con trỏ tiếp của nút cuối (được trỏ tới bởi r) trỏ đến nút mới tạo là q, và cho con trỏ tiếp của q trỏ tới NULL  2. Dịch chuyển con trỏ tới nút cuối của danh sách  3. Tạo và cấp phát bộ nhớ cho 1 nút mới |

|  |  |
| --- | --- |
| Ý tưởng của giải thuật chèn một nút vào sau nút r trong danh sách liên kết đơn được trỏ tới bởi con trỏ q nào dưới đây là đúng? | |
| A | 1. Tạo và cấp phát bộ nhớ cho 1 nút mới.  2. Cho con trỏ tiếp của nút r trỏ đến q  3. Cho con trỏ tiếp của q trỏ tới NULL |
| B | 1. Tạo và cấp phát bộ nhớ cho 1 nút mới.  2. Cho con trỏ tiếp của nút r trỏ đến q  3. Cho con trỏ tiếp theo của nút mới trỏ đến nút kế tiếp của nút r. |
| C | 1. Tạo và cấp phát bộ nhớ cho 1 nút mới.  2. Cho con trỏ tiếp theo của nút mới trỏ đến nút kế tiếp của nút r.  3. Cho con trỏ tiếp của nút r trỏ đến q |
| D | 1. Tạo và cấp phát bộ nhớ cho 1 nút mới.  2. Cho con trỏ tiếp theo của nút mới trỏ đến nút kế tiếp của nút r.  3. Cho con trỏ tiếp của r trỏ tới NULL |

|  |  |  |
| --- | --- | --- |
| Đoạn chương trình sau mô tả giải thuật chèn một nút mới vào sau nút r trong danh sách liên kết đơn được trỏ tới bởi con trỏ q:  while ((r !=NULL) && (found == 0))  { q = (listnode)malloc(sizeof(struct node));  q 🡪 item = x;  // Cho con trỏ tiếp theo của nút mới trỏ đến nút kế tiếp của nút r.  ………………………………. (1)  // Cho con trỏ tiếp của nút r trỏ đến q  ………………………………….. (2)  found = 1;  }  Hãy chọn câu lệnh đúng điền vào vị trí (1) và (2)? | | |
| A | (1) q 🡪 next = r 🡪 next;  (2) r 🡪 next = NULL; |
| B | (1) q 🡪 next = r 🡪 next;  (2) q= r 🡪 next; |
| C | (1) r 🡪 next = q🡪 next;  (2) r 🡪 next = q; |
| D | (1) q 🡪 next = r 🡪 next;  (2) r 🡪 next = q; |

|  |  |
| --- | --- |
| Thuật toán tìm phần tử có nội dung bằng x được mô tả như sau:  listnode Search\_Item(listnode First, int x);  { listnode p;  p = First;  while ((p!= NULL)&&( p🡪 item !=x))  …………… (1)  return p;  }  Hãy chọn câu lệnh đúng điền vào vị trí (1)? | |
| A | p = p🡪 next; |
| B | if (p🡪 item ==x) |
| C | p🡪 next = NULL; |
| D | x = p🡪next; |

|  |  |  |
| --- | --- | --- |
| Thuật toán kiểm tra xem phần tử ở vị trí thứ 3 có giá trị bằng x hay không được mô tả như sau:  listnode Search\_Item(listnode First, int x);  { listnode p; int count =0; Kt=0;  p = First;  while ((p!= NULL)  { ……………….. (1)  if ((count ==3) && (p🡪item ==x))  Kt=1;  p = p🡪 next;  }  return Kt;  }  Hãy chọn câu lệnh đúng điền vào vị trí (1)? | | |
| A | p🡪 next = NULL; |
| B | count++; |
| C | count !=3; |
| D | Kt = 0; |

|  |  |  |
| --- | --- | --- |
| Thuật toán đếm xem có bao nhiêu phần tử có giá trị chẵn trong danh sách được mô tả như sau:  listnode Search\_Item(listnode First, int item);  { listnode p; int count =0;  p = First;  while ((p!= NULL)  { ………………….. (1)  …………………… (2)  p = p🡪 next;  }  return count;  }  Hãy chọn câu lệnh đúng điền vào vị trí (1) và (2)? | | |
| A | if (p🡪item ==x && p🡪item %2==0)  count ++; |
| B | if (p🡪item !=0&& p🡪item %2==0)  count ++; |
| C | if (p🡪item %2 !=0))  count ++; |
| D | if (p🡪item %2 ==0))  count ++; |

|  |  |  |
| --- | --- | --- |
| Thuật toán tính tổng các phần tử có giá trị chẵn, dương trong danh sách được mô tả như sau:  listnode Search\_Item(listnode First, int item);  { listnode p; int s=0;  p = First;  while ((p!= NULL)  { ………………….. (1)  …………………… (2)  p = p🡪 next;  }  return s;  }  Hãy chọn câu lệnh đúng điền vào vị trí (1) và (2)? | | |
| A | if (p🡪 item %2 ==0 && p🡪 item >0)  s = s + p🡪 item; |
| B | if (p🡪 item %2 ==0 && p🡪 item >0)  { s = s + p🡪 item;  p = p🡪 next;  } |
| C | if (p🡪 item %2 ==0 && p🡪 item !=0)  s = s + p🡪 item; |
| D | if (p🡪 item %2 ==0 && p🡪 item >0)  s = s + p🡪 next; |

|  |  |
| --- | --- |
| Thuật toán tính tổng các phần tử chẵn và dương trong danh sách được mô tả như sau:  listnode Search\_Item(listnode First, int item);  { int s=0;  p = First;  while ((p!= NULL)  { if (p🡪 item %2 ==0 && p🡪 item >0)  s = s + p🡪 item;  p = p🡪 next;  }  return s;  }  Trong các biến được sử dụng trong chương trình, biến nào chưa được khai báo? | |
| A | s |
| B | x |
| C | p |
| D | item |

|  |  |  |
| --- | --- | --- |
| Ý tưởng của giải thuật xóa một nút ở đầu danh sách liên kết đơn được trỏ tới bởi con trỏ q nào dưới đây là đúng? | | |
| A | 1. Dùng 1 con trỏ tạm q trỏ đến đầu danh sách  2. Ngắt liên kết của biến tạm q với nút tiếp theo, giải phóng bộ nhớ cho q  3. Dịch chuyển con trỏ p qua phần tử đầu tiên đến phần tử kế tiếp |
| B | 1. Dùng 1 con trỏ tạm q trỏ đến đầu danh sách  2. Dịch chuyển con trỏ p qua phần tử đầu tiên đến phần tử kế tiếp  3. Ngắt liên kết của biến tạm q với nút tiếp theo, giải phóng bộ nhớ cho q |
| C | 1. Dùng 1 con trỏ tạm q trỏ đến đầu danh sách  2. Ngắt liên kết của biến tạm q với nút tiếp theo, giải phóng bộ nhớ cho q |
| D | 1. Ngắt liên kết của biến tạm q với nút tiếp theo, giải phóng bộ nhớ cho q  2. Dịch chuyển con trỏ p qua phần tử đầu tiên đến phần tử kế tiếp |

|  |  |  |
| --- | --- | --- |
| Đoạn chương trình dưới đây mô tả thuật toán xóa bỏ một nút ở đầu danh sách liên kết đơn:  if (\*p == NULL) return;  q = \* p;  \*p = (\*p) 🡪 next;  //Ngắt liên kết của biến tạm q với nút tiếp theo  …………………………....... (1)  // giải phóng bộ nhớ cho q  ………………………………(2)  Hãy chọn câu lệnh đúng điền vào vị trí (1) và (2)? | | |
| A | (1) q 🡪 next = NULL;  (2) free(q); |
| B | (1) q 🡪 item = NULL;  (2) free(q) |
| C | (1) q 🡪 next = p 🡪 next;  (2) free(q); |
| D | (1) q 🡪 next = NULL;  (2) q = (listnode)malloc(sizeof(struct node)); |

|  |  |
| --- | --- |
| Ý tưởng của giải thuật xóa một nút ở cuối danh sách liên kết đơn được trỏ tới bởi con trỏ q nào dưới đây là đúng? | |
| A | 1. Dịch chuyển con trỏ tới nút gần nút cuối của danh sách  2. Giải phóng bộ nhớ của nút cuối cùng |
| B | 1. Cho con trỏ tiếp của nút gần nút cuối cùng nhất  2. Dịch chuyển con trỏ tới nút gần nút cuối của danh sách  3. Giải phóng bộ nhớ của nút cuối cùng |
| C | 1. Dịch chuyển con trỏ tới nút gần nút cuối của danh sách  2. Cho con trỏ tiếp của nút gần nút cuối cùng nhất  3. Giải phóng bộ nhớ của nút cuối cùng |
| D | 1. Cho con trỏ tiếp của nút gần nút cuối cùng nhất  2. Giải phóng bộ nhớ của nút cuối cùng |

|  |  |  |
| --- | --- | --- |
| Đoạn chương trình dưới đây mô tả thuật toán xóa bỏ một nút ở cuối danh sách liên kết đơn:  //Dịch chuyển con trỏ tới gần nút cuối của danh sách  ……………………………(1)  q🡪 next = NULL;  free(r);  Hãy chọn câu lệnh đúng điền vào vị trí (1)? | | |
| A | while (r 🡪 next != NULL)  { r = r 🡪 next;  q = r;  } |
| B | while (r 🡪 next != NULL)  { r = r 🡪 next;  q = r;  } |
| C | while (r 🡪 next == NULL)  { q = r;  r = r 🡪 next;  } |
| D | while (r 🡪 next != NULL)  { q = r;  r = r 🡪 next;  } |

|  |  |
| --- | --- |
| Ý tưởng của giải thuật xóa một nút ở sau nút r trong danh sách liên kết đơn được trỏ tới bởi con trỏ q nào dưới đây là đúng? | |
| A | 1. Cho con trỏ tiếp của nút r trỏ tới nút đứng sau nút q  2. Sử dụng 1 biến tạm q trỏ đến nút đứng sau nút r  3. Ngắt liên kết của nút q và giải phóng bộ nhớ cho q |
| B | 1. Sử dụng 1 biến tạm q trỏ đến nút đứng sau nút r  2. Cho con trỏ tiếp của nút r trỏ tới nút đứng sau nút q  3. Ngắt liên kết của nút q và giải phóng bộ nhớ cho q |
| C | 1. Cho con trỏ tiếp của nút r trỏ tới nút đứng sau nút q  2. Ngắt liên kết của nút q và giải phóng bộ nhớ cho q |
| D | 1. Sử dụng 1 biến tạm q trỏ đến nút đứng sau nút r  2. Ngắt liên kết của nút q và giải phóng bộ nhớ cho q  3. Cho con trỏ tiếp của nút r trỏ tới nút đứng sau nút q |

|  |  |  |
| --- | --- | --- |
| Đoạn chương trình dưới đây mô tả thuật toán xóa bỏ một nút ở sau nút r trong danh sách liên kết đơn:  while ((r !=NULL) && (found == 0))  { if (count == position)  { q = r 🡪 next;  // Cho con trỏ tiếp của nút r trỏ tới nút đứng sau nút q  ………………………….(1)  // Ngắt liên kết của nút q  ………………………….(2)  free(q)  }  count ++; r = r 🡪 next;  }  Hãy chọn câu lệnh đúng điền vào vị trí (1) và (2)? | | |
| A | (1) q🡪 next = NULL;  (2) r 🡪 next = q 🡪 next; |
| B | (1) r 🡪 next = q 🡪 next;  (2) q🡪 next = NULL; |
| C | (1) r 🡪 next = q 🡪 next;  (2) r🡪 next = NULL; |
| D | (1) q 🡪 next = r 🡪 next;  (2) r🡪 next = NULL; |

|  |  |  |
| --- | --- | --- |
| Biểu thức toán học () viết dưới dạng hậu tố ký pháp Balan có dạng nào? | | |
| A | A B C D E \* + / - |
| B | A B + C D E \* + / |
| C | A B C D / - E + |
| D | A B - C D + E \* / |

|  |  |
| --- | --- |
| Biểu thức toán học viết dưới dạng hậu tố ký pháp Balan có dạng nào? | |
| A | A B + C D - A \* E F - + / |
| B | A B + C D - A \* / E F - + |
| C | A B + C D A \* /- E F - + |
| D | A B + C D A \*- E F - + / |

|  |  |
| --- | --- |
| Biểu thức toán học ( + F) viết dưới dạng hậu tố ký pháp Balan có dạng nào? | |
| A | A + BCE \* + CD – F / + |
| B | A B + C E \* + C D - / F + |
| C | A B + C \* E + C D - / F + |
| D | A B + C \* E C D + - / F + |

|  |  |  |
| --- | --- | --- |
| Biểu thức hậu tố ký pháp Balan có dạng: AB + CDA + - \*. Với A=1; B=5; C=8; D=4 có giá trị là bao nhiêu? | | |
| A | 8 |
| B | 10 |
| C | 17 |
| D | 18 |

|  |  |
| --- | --- |
| Trong các phát biểu dưới đây về hàng đợi (QUEUE), phát biểu nào là đúng? | |
| A | Hàng đợi còn được gọi là danh sách hoạt động theo kiểu LIFO và cấu trúc dữ liệu này được gọi là cấu trúc LIFO (Last In First Out) |
| B | Hàng đợi là một danh sách mà trong đó thao tác bổ sung và loại bỏ một phần tử trong hàng đợi được thực hiện ở 1 đầu khác nhau. |
| C | Hàng đợi là một kiểu danh sách đặc biệt mà trong đó thao tác bổ sung và loại bỏ một phần tử trong hàng đợi được thực hiện ở 2 đầu |
| D | Hàng đợi phải là 1 danh sách liên kết đơn |

|  |  |
| --- | --- |
| Cấu trúc dữ liệu nào dưới đây tương ứng với FIFO (First In First Out)? | |
| A | Hàng đợi (Queue) |
| B | Danh sách liên kết (Linked List) |
| C | Cây (Tree) |
| D | Ngăn xếp (Stack) |

|  |  |
| --- | --- |
| Trong các phát biểu dưới đây về ngăn xếp (STACK), phát biểu nào là đúng? | |
| A | Ngăn xếp còn được gọi là danh sách hoạt động theo kiểu FIFO và cấu trúc dữ liệu này được gọi là cấu trúc FIFO (First In First Out) |
| B | Ngăn xếp là 1 kiểu danh sách đặt biệt mà việc thêm vào một phần tử hoặc loại bỏ 1 phần tử chỉ được thực hiện ở đầu danh sách gọi là đỉnh |
| C | Ngăn xếp là một kiểu danh sách đặc biệt mà trong đó thao tác bổ sung và loại bỏ một phần tử trong ngăn xếp được thực hiện ở 2 đầu |
| D | Ngăn xếp phải là 1 danh sách liên kết đơn |

|  |  |
| --- | --- |
| Cấu trúc dữ liệu nào dưới đây tương ứng với LIFO (Last In First Out)? | |
| A | Hàng đợi (QUEUE) |
| B | Ngăn xếp (STACK) |
| C | Cây (TREE) |
| D | Danh sách liên kết (Linked List) |

|  |  |
| --- | --- |
| Khi bổ sung 1 phần tử vào ngăn xếp (Stack), phần tử mới sẽ được đưa vào vị trí nào? | |
| A | Đỉnh ngăn xếp |
| B | Một vị trí bất kì |
| C | Đáy ngăn xếp |
| D | Vị trí kế tiếp |

|  |  |
| --- | --- |
| Loại bỏ 1 phần tử ra khỏi ngăn xếp, khi đó, phần tử ở vị trí nào sẽ được lấy ra khỏi ngăn xếp? | |
| A | Đỉnh ngăn xếp |
| B | Một vị trí bất kì |
| C | Đáy ngăn xếp |
| D | Vị trí kế tiếp |

|  |  |
| --- | --- |
| Khi Stack rỗng, số phần tử thực sự đang chứa trong mảng là bao nhiêu? | |
| A | -1 |
| B | 0 |
| C | 1 |
| D | NULL |

|  |  |  |
| --- | --- | --- |
| Khi cài đặt ngăn xếp bằng mảng, lệnh nào dưới đây có chức năng kiểm tra ngăn xếp rỗng? | | |
| A | s->top = -1; |
| B | s.top ==-1 |
| C | s.top ==NULL |
| D | s.top == NULL |

|  |  |  |
| --- | --- | --- |
| Khi cài đặt ngăn xếp bằng mảng, lệnh nào dưới đây có chức năng khởi tạo ngăn xếp rỗng? | | |
| A | s->top = -1; |
| B | s.top == -1 |
| C | s.top == NULL |
| D | s🡪top = 0 |

|  |  |
| --- | --- |
| Khi cài đặt ngăn xếp bằng mảng, lệnh nào dưới đây có chức năng kiểm tra ngăn xếp đầy? | |
| A | s->top == MAX; | |
| B | s.top == MAX | |
| C | S.Top == MAX - 1 | |
| D | s🡪top = =MAX -1 | |

|  |  |
| --- | --- |
| Khi cài đặt ngăn xếp bằng danh sách liên kết, lệnh nào dưới đây có chức năng khởi tạo ngăn xếp rỗng? | |
| A | s🡪top == -1 |
| B | s.top == -1 |
| C | s.top = = NULL |
| D | s🡪top == NULL |

|  |  |
| --- | --- |
| Khi cài đặt ngăn xếp bằng danh sách liên kết, lệnh nào dưới đây có chức năng kiểm tra ngăn xếp rỗng? | |
| A | s🡪top == 0 |
| B | s.top == 0 |
| C | s.top = = NULL |
| D | s🡪top == NULL |

|  |  |  |
| --- | --- | --- |
| Trong các ứng dụng dưới đây, ứng dụng nào KHÔNG nên thực hiện theo cơ chế ngăn xếp? | | |
| A | Nạp và tháo đạn ra khỏi súng |
| B | Xếp hàng mua vé |
| C | Tính giá trị một biểu thức hậu tố |
| D | Đảo ngược xâu kí tự |

|  |  |
| --- | --- |
| Khi bổ sung 1 phần tử vào hàng đợi (Queue), phần tử mới sẽ được đưa vào vị trí nào? | |
| A | Lối sau của hàng đợi |
| B | Một vị trí bất kì |
| C | Lối trước của hàng đợi |
| D | Vị trí kế tiếp |

|  |  |
| --- | --- |
| Khi loại bỏ 1 phần tử ra khỏi hàng đợi, khi đó, phần tử ở vị trí nào sẽ được lấy ra? | |
| A | Lối sau của hàng đợi |
| B | Một vị trí bất kì |
| C | Lối trước của hàng đợi |
| D | Vị trí kế tiếp |

|  |  |
| --- | --- |
| Khi hàng đợi rỗng, số phần tử thực sự đang chứa trong mảng là bao nhiêu? | |
| A | -1 |
| B | 0 |
| C | 1 |
| D | NULL |

|  |  |
| --- | --- |
| Khi cài đặt hàng đợi bằng mảng, lệnh nào dưới đây có chức năng kiểm tra hàng đợi rỗng? (trong đó count cho biết số phần tử hiện tại của hàng đợi) | |
| A | q.count <=0 |
| B | q.count !=0 |
| C | q.count == 0 |
| D | q.count = NULL |

|  |  |  |
| --- | --- | --- |
| Khi cài đặt hàng đợi bằng mảng, lệnh nào dưới đây có chức năng khởi tạo hàng đợi rỗng? (trong đó First lưu chỉ số phần tử đầu, Last lưu chỉ số cuối, count cho biết số phần tử hiện tại của hàng đợi) | | |
| A | q🡪 First = 0;  q🡪 Last = Max-1;  q🡪 Count = 0; |
| B | q. First = 0;  q. Last = Max-1;  q. Count = 0; |
| C | q🡪 First = NULL;  q🡪 Last = Max-1;  q🡪 Count = 0; |
| D | q. First = NULL;  q. Last = Max-1;  q. Count = 0; |

|  |  |
| --- | --- |
| Khi cài đặt hàng đợi bằng danh sách liên kết, lệnh nào dưới đây có chức năng kiểm tra hàng đợi rỗng? | |
| A | q.Last == 0 | |
| B | q.First ==0 | |
| C | q.Last = NULL | |
| D | q.First ==NULL | |

|  |  |  |
| --- | --- | --- |
| Khi cài đặt hàng đợi bằng danh sách liên kết, lệnh nào dưới đây có chức năng khởi tạo hàng đợi rỗng | | |
| A | q 🡪 First = q 🡪 Last ;  q🡪 Last= NULL; |
| B | q 🡪 First = 0 ;  q🡪 Last= 0; |
| C | q 🡪 First = q 🡪 Last = NULL; |
| D | q 🡪 First = q 🡪 Last = 0; |

|  |  |
| --- | --- |
| Khi cài đặt hàng đợi bằng mảng, lệnh nào dưới đây có chức năng kiểm tra hàng đợi đầy? | |
| A | q 🡪 count == Max | |
| B | q🡪count = Max | |
| C | q.count ==Max | |
| D | q.count = Max | |

|  |  |
| --- | --- |
| Trong các ứng dụng dưới đây, ứng dụng nào thực hiện theo cơ chế hàng đợi? | |
| A | Đổi cơ số từ hệ 10 sang các hệ khác |
| B | Một tập các tác vụ chờ phục vụ bởi hệ điều hành máy tính |
| C | Tính giá trị biểu thức dạng trung tố có dấu ngoặc đầy đủ |
| D | Đảo ngược xâu kí tự |

|  |  |
| --- | --- |
| Kết quả của biểu thức dạng hậu tố E= 5 3 + 2 / 4 2 - \* khi ứng dụng ngăn xếp để tính: E = bao nhiêu? | |
| A | 5 |
| B | 7 |
| C | 8 |
| D | 10 |

|  |  |
| --- | --- |
| Biểu thức toán học (((6 +2) \* 3) / (7 - 4)) viết dưới dạng hậu tố ký pháp Balan có dạng nào? | |
| A | 6 2 3 7 4 + \* / - |
| B | 6 2 + 3 \* 7 4 / - |
| C | 6 2 3 \* + 7 4 - / |
| D | 6 2 + 3 \* 7 4 - / |

|  |  |  |
| --- | --- | --- |
| Biểu thức dạng hậu tố: 3 5 2 - 1 7 + \* 6 - \* tương ứng với biểu thức toán học nào dưới đây? | | |
| A | (3\*(((5 - 2) \* (1+ 7) -6))) |
| B | ((((3\*5) - 2) \* (1 + 7) -6)) |
| C | (3\*(((5 - 2) \* (1 + (7-6))) |
| D | ((((3\*5) - 2) \* (1 + (7-6)) |

|  |  |
| --- | --- |
| Khi tính giá trị biểu thức dạng hậu tố: E=3 5 2 - 1 7 + \* 6 - \* kết quả của E bằng bao nhiêu? | |
| A | 16 |
| B | 18 |
| C | 24 E 54 |
| D | 26 |

|  |  |
| --- | --- |
| Biểu thức dạng hậu tố E=A B + U V - C \* / tương ứng với biểu thức toán học nào dưới đây? | |
| A | (((A+B)/(U - V))\* C) |
| B | ((A+B)/((U - (V\* C))) |
| C | ((A+B)/((U - V)\* C)) |
| D | ((A+(B/(U - V))\* C)) |

|  |  |  |
| --- | --- | --- |
| Cho dãy thao tác: EAS\*Y\*\* Với một chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào hàng đợi, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong hàng đợi in lên màn hình. Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình? | | |
| A | EAS |
| B | YSA |
| C | SYA |
| D | ASY |

|  |  |
| --- | --- |
| Cho dãy thao tác: EAS\*Y\*\* Với một chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào ngăn xếp, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong ngăn xếp in lên màn hình. Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình? | |
| A | SAE |
| B | EAS |
| C | YSA |
| D | SYA |

|  |  |
| --- | --- |
| Khi tính giá trị biểu thức dạng hậu tố: E=5 3 + 7 5 - 2 \* / kết quả của E bằng bao nhiêu? | |
| A | 4 |
| B | 2 |
| C | 8 |
| D | 6 |

|  |  |
| --- | --- |
| Cho một danh sách liên kết đơn lưu trữ các số nguyên, hãy xóa bỏ các phần tử chia hết cho 3 nhưng không chia hết cho 5 trong danh sách. Đoạn chương trình sau đây mô tả thuật toán trên:  while (r !=NULL)  { …………………………….. (1)  { q = r 🡪 next; (2)  r 🡪 next = q 🡪 next; (3)  q🡪 next = NULL; (4)  free(q); (5)  }  r = r 🡪 next; (6)  }  Hãy chọn câu lệnh đúng điền vào vị trí (1)? | |
| A | if (q 🡪 item %3 !=0 && q🡪item %5 !=0) |
| B | if (q 🡪 item %3 ==0 && q🡪item %5 !=0) |
| C | if (q 🡪 item %3 ==0 || q🡪item %5 !=0) |
| D | if (q 🡪 item %3 ==0 && q🡪item %5 ==0) |

|  |  |
| --- | --- |
| Quy tắc cộng được áp dụng tính độ phức tạp cho 2 đoạn chương trình P1, P2 trong trường hợp nào dưới đây? | |
| A | P1 và P2 lồng nhau |
| B | Thực hiện tuần tự P1 rồi đến P2 |
| C | Thực hiện đồng thời P1, P2 |
| D | Trong đoạn P1 có một phần của P2 |

|  |  |
| --- | --- |
| Quy tắc nhân được áp dụng tính độ phức tạp cho 2 đoạn chương trình P1, P2 trong trường hợp nào dưới đây? | |
| A | P1 và P2 lồng nhau |
| B | Thực hiện tuần tự P1 rồi đến P2 |
| C | Thực hiện đồng thời P1, P2 |
| D | Trong đoạn P1 có một phần của P2 |

|  |  |
| --- | --- |
| Câu lệnh sau có độ phức tạp tính toán là bao nhiêu?  for(i=0; i<=n; i++)  printf(“\n So thu %d:” ,i); | |
| A | O(1) |
| B | O(i) |
| C | O(n) |
| D | O(n2) |

|  |  |
| --- | --- |
| Đoạn câu lệnh sau có độ phức tạp tính toán là bao nhiêu?  for(i=0; i<n; i++)  for(j=0; j<n; j++)  S++; | |
| A | O(n2) |
| B | O(N) |
| C | O(NlogN) |
| D | O(1) |

|  |  |
| --- | --- |
| Đoạn câu lệnh sau có độ phức tạp tính toán là bao nhiêu?  1. sum = 0;  2. for(int i = 0; i<=n; i++)  2.1. for(k=1; k<10; k++)  2.1.1. sum++; | |
| A | O(10n) |
| B | O(n2) |
| C | O(1) |
| D | O(n3) |

|  |  |
| --- | --- |
| Đoạn câu lệnh sau có độ phức tạp tính toán là bao nhiêu?  1. i = 0;  2. while(i<n && x!= A[i])  2.1. i++; | |
| A | O(n) |
| B | O(NlogN) |
| C | O(n2) |
| D | O(2n) |

|  |  |
| --- | --- |
| Ý tưởng của phương pháp quy hoạch động là gì? | |
| A | Tìm nghiệm của một bài toán bằng cách xem xét tất cả các phương án có thể |
| B | Phương pháp xây dựng các thuật toán giải các bài toán tối ưu dựa trên tư tưởng tối ưu cục bộ theo một chiến lược tư duy kiểu con người nhằm nhanh chóng đạt đến một lời giải “tốt” |
| C | Để có lời giải của bài toán tối ưu có kích thước n, ta giải bài toán tương tự có kích thước nhỏ hơn và phối hợp lời giải của chúng để được lời giải của bài toán ban đầu |
| D | Chọn cái tốt nhất trước, sau đó cải tiến chúng thành những cái tốt hơn |

|  |  |
| --- | --- |
| Ưu điểm lớn nhất của phương pháp vét cạn là gì? | |
| A | Luôn đảm bảo tìm ra nghiệm chính xác hoặc thông báo vô nghiệm khi không tìm ra nghiệm |
| B | Thường cho kết quả gần đúng |
| C | Dễ thiết kế giải thuật |
| D | Không đòi hỏi máy tính phải có cấu hình cao |

|  |  |
| --- | --- |
| Bước phác thảo giải thuật tìm số nhỏ nhất như sau:  “Từ dãy các số nguyên, chọn ra số nhỏ nhất”  Bước tiếp theo để thực hiện giải thuật theo phương pháp tinh chình từng bước nào dưới đây là đúng? | |
| A | min = a[1];  for(i=2; i<=n; i+=)  if(a[i] < min) min = a[i]; |
| B | for(i=1; i<=n; i++)  Xét từ a[1] đến a[n] để tìm số nhỏ nhất; |
| C | for(i=1; i<=n; i++)  Tìm số nhỏ nhất |
| D | Xét từ a[1] đến a[n] để tìm số nhỏ nhất; |

|  |  |
| --- | --- |
| Bước phác thảo đầu tiên của bài toán tính tổng các số lẻ âm trong một dãy các số nguyên như sau:  - Xét từ a[1] đến a[n]  - Tìm các số lẻ, âm  - Tính tổng  Bước tiếp theo của bài toán theo phương pháp tinh chỉnh từng bước nào dưới đây là đúng? | |
| A | Duyệt các phần tử  if (a[i] %2!=0 && a[i]<0)  s+ = a[i]; |
| B | Duyệt các phần tử  if (a[i] %2!=0 && a[i]<0)  Tính tổng |
| C | for (i=1; i<=n; i++)  if (a[i] %2!=0 && a[i]<0)  s+ = a[i]; |
| D | for (i=1; i<=n; i++)  if (a[i] %2!=0 || a[i]<0)  s+ = a[i]; |

|  |  |
| --- | --- |
| Bước phác thảo của đoạn chương trình đếm các số chính phương trong một dãy các số nguyên như sau:  1. Nhập n, các phần tử của mảng  2. Đếm các số chính phương  Bước tiếp theo để thực hiện lệnh 2 theo phương pháp tinh chỉnh từng bước nào dưới đây là đúng? | |
| A | 2.1 Xây dựng hàm tìm số chính phương  2.2 for (i=1; i<=n; i++)  if (cphuong(a[i])) d++; |
| B | 2.1 Xây dựng hàm tìm số chính phương  2.2 Kiểm tra và đếm số chính phương trong mảng |
| C | 2.1 cphuong=True;  2.2 for (i = 1; i<=n/2; i++)  if (i\*i !=n)  cphuong=False; |
| D | 2.1 for (i=1; i<=n; i++)  2.2 if (a[i] % i != 0) d++; |

|  |  |
| --- | --- |
| Bài toán: Cho một dãy số nguyên, đếm xem có bao nhiêu số lẻ  Bước phác thảo đầu tiên theo phương pháp tinh chỉnh từng bước nào dưới đây là đúng? | |
| A | - Nhập n, nhập các phần tử của mảng  - Đếm các số lẻ trong mảng |
| B | - Duyệt các phần tử trong mảng  - Đếm các số lẻ |
| C | for(i=1; i<=n; i++)  Đếm các số lẻ trong mảng |
| D | Duyệt các phần tử trong mảng  if(a[i]%2!=0) dem++; |

|  |  |
| --- | --- |
| Xây dựng hàm kiểm tra n có phải là số chính phương hay không được phác thảo như sau:  - Duyệt i chạy từ 1 đến n/2  - Nếu i \* i = n thì n là số chính phương  Bước tiếp theo để thực hiện đoạn chương trình theo phương pháp tinh chỉnh từng bước nào dưới đây là đúng? | |
| A | Cp = 0;  for (i = 1; i<=n/2; i++)  if (i \* i ==n) d++; |
| B | Cp = 0;  for (i = 1; i<=n/2; i++)  Kiểm tra số n có phải là số chính phương hay không  In kết quả |
| C | Cp = 0;  for (i = 1; i<=n/2; i++)  if (i \* i ==n) Cp = 1; |
| D | Cp = 0;  for (i = 1; i<=n/2; i++)  if ( i \* i == a[n]) Cp= 1; |

|  |  |
| --- | --- |
| Bước phác thảo của đoạn chương trình đếm các số nguyên tố trong một dãy các số nguyên như sau:  1. Nhập n, các phần tử của mảng  2. Đếm các số nguyên tố  Bước tiếp theo để thực hiện lệnh 2 theo phương pháp tinh chỉnh từng bước nào dưới đây là đúng? | |
| A | 2.1 Xây dựng hàm tìm số nguyên tố  2.2 Kiểm tra và đếm số nguyên tố trong mảng |
| B | 2.1 nto=True;  2.2 for (i = 2; i<=n/2; i++)  if (n%i ==0)  nto=False; |
| C | 2.1 Xây dựng hàm tìm số nguyên tố  2.2 for (i=1; i<=n; i++)  if (nto(a[i])) d++; |
| D | 2.1 for (i=1; i<=n; i++)  2.2 if (a[i] % i != 0) d++; |

|  |  |
| --- | --- |
| Áp dụng phương pháp tinh chỉnh từng bước xây dựng thuật toán nhập một dãy các số nguyên, đếm xem trong dãy có bao nhiêu số lẻ dương, chia hết cho 3 nhưng không chia hết cho 5 và tính tích các số ấy. Bước phác thảo đầu tiên nào dưới đây là đúng? | |
| A | 1. Nhập n, các phần tử của mảng  2. Đếm các số lẻ, dương, chia hết cho 3 nhưng không chia hết cho 5  3. Tính tích |
| B | 1. Nhập n, các phần tử trong mảng  2. for (i=1; i<=n; i++)  2.1. Tính tích các số lẻ, dương chia hết cho 3 không chia hết cho 5 |
| C | 1. Nhập n, các phần tử trong mảng  2. for (i=1; i<=n; i++)  2.1. Tính tích và đếm các số lẻ, dương, chia hết cho 3 không chia hết cho 5  3. Tính tích |
| D | 1. Nhập n, các phần tử trong mảng  2. Tính tích các số lẻ, dương, chia hết cho 3 không chia hết cho 5 |

|  |  |
| --- | --- |
| Áp dụng phương pháp tinh chỉnh từng bước xây dựng thuật toán: nhập một dãy các số nguyên, đếm các số hoàn hảo có trong mảng. Bước phác thảo đầu tiên nào dưới đây là đúng? | |
| A | 1. Nhập n, các phần tử trong mảng  2. Đếm số hoàn hảo |
| B | 1. Nhập n, các phần tử trong mảng  2. for (i = 1; i<n; i++)  if (n % i ==0) hoanhao ++; |
| C | 1. Nhập n, các phần tử trong mảng  2. for (i = 1; i<n; i++)  if (n % i ==0) hoanhao ++; |
| D | 1. Nhập n, các phần tử trong mảng  2. for (i = 1; i<n; i++)  if (n % i ==0) hoanhao = hoanhao +i; |

|  |  |
| --- | --- |
| Phác thảo sau mô tả thuật toán gì?  1. Gán S = 0;  2. Cho i chạy từ 1 đến n/2;  3. Kiểm tra nếu bình phương của i bằng n thì gán S = 1; | |
| A | Kiểm tra số nguyên tố |
| B | Kiểm tra số chính phương |
| C | Kiểm tra số hoàn hảo |
| D | Kiểm tra số chẵn, dương |

|  |  |
| --- | --- |
| Đoạn chương trình sau mô tả thuật toán gì?  S = 1;  for (i = 1; i<=n/2; i++)  if ( n%i ==0)  S = 0; | |
| A | Kiểm tra số nguyên tố |
| B | Kiểm tra số chính phương |
| C | Kiểm tra số hoàn hảo |
| D | Kiểm tra số chẵn, dương |

|  |  |
| --- | --- |
| Thời gian thực hiện một chương trình được định nghĩa là gì? | |
| A | Một hàm của kích thước dữ liệu vào, ký hiệu T(n) trong đó n là kích thước (độ lớn) của dữ liệu vào. |
| B | Một hàm của độ dài dữ liệu vào, ký hiệu N(x) trong đó x là độ dài của dữ liệu vào. |
|  | Thời gian ngắn nhất để thực hiện chương trình đối với mọi dữ liệu vào có cùng kích thước n. |
| D | Thời gian thực hiện chương trình trong trường hợp nhanh nhất trên dữ liệu vào có kích thước n |

|  |  |
| --- | --- |
| Thời gian thực hiện của một giải thuật T(n) được xác định bằng gì? | |
| A | Đơn vị đo thời gian bình thường giờ, phút, giây... |
| B | Không phải là đơn vị đo thời gian bình thường như giờ, phút, giây.... |
| C | Được xác định bởi số các lệnh được thực hiện trong một máy tính lý tưởng. |
| D | Được xác định bởi độ lớn của dữ liệu đầu vào |

|  |  |
| --- | --- |
| Thời gian thực hiện một giải thuật là T(n) = Cn (với C là hằng số) thì ta nói độ phức tạp tính toán của giải thuật có cấp là gì? | |
| A | C |
| B | Cn |
| C | n |
| D | Cn2 |

|  |  |
| --- | --- |
| Đoạn chương trình sau có độ phức tạp tính toán là bao nhiêu?  1.for(i=0; i<n; i++)  1.1. for(j=0; j<n; j++)  1.1.1. a[j][j] = 0;  2. for(i=0; i<n; i++)  2.1. a[i][j] = 1; | |
| A | O(n3) |
| B | O(n2) |
| C | O(n) |
| D | O(2n) |

|  |  |
| --- | --- |
| Đoạn chương trình sau có độ phức tạp tính toán là bao nhiêu?  int Fact(int n)  {  if(n = = 1) return 1;  else return n\*Fact(n-1);  } | |
| A | O(n) |
| B | O(n2) |
| C | O(n3) |
| D | O(1) |

|  |  |
| --- | --- |
| Trong các nguyên tắc dưới đây, nguyên tắc nào KHÔNG phải là nguyên tắc của các phương pháp thiết kế thuật toán? | |
| A | Tìm nghiệm của một bài toán bằng cách xem xét tất cả các phương án có thể |
| B | Chọn cái tốt nhất trước, sau đó cải tiến chúng thành những cái tốt hơn |
| C | Thiết kế bài toán theo cấu trúc hình cây. |
| D | Để có lời giải của bài toán tối ưu kích thước n, ta giải các bài toán tương tự có kích thước nhỏ hơn và phối hợp lời giải của chúng để được lời giải của bài toán ban đầu |

|  |  |
| --- | --- |
| Nhược điểm chính của phương pháp vét cạn là gì? | |
| A | Độ phức tạp tính toán của bài toán rất lớn (lên đến hàm mũ) |
| B | Đòi hỏi máy tính thực hiện phải có cấu hình cao |
| C | Bài toán luôn tìm được nghiệm |
| D | Thường cho kết quả gần đúng |

|  |  |
| --- | --- |
| Trong trường hợp xấu nhất đoạn lệnh sau có độ phức tạp tính toán là bao nhiêu?  j = n;  while(j>=i)  {  for(i=1; i<=j; j++) x=x+1;  j = j div 2;  } | |
| A | O(n2) |
| B | O(1) |
| C | O(n3) |
| D | O(n) |

|  |  |
| --- | --- |
| Đoạn chương trình sau có độ phức tạp là bao nhiêu?  int Linear\_Search(int s, int n, int key)  { i=0;  while(i<=n)||(key= si) i=i+1;  Linear\_Search=i;  } | |
| A | O(n) |
| B | O(n2) |
| C | O(nlogn) |
| D | O(1) |

|  |  |
| --- | --- |
| Đoạn chương trình sau có độ phức tạp là bao nhiêu?  int Find\_Large(int a, int n, int large)  {  large = a1;  for(i=2; i<=n; i++)  {  if (ai>large) large = ai;  }  } | |
| A | O(n) |
| B | O(n2) |
| C | O(n3) |
| D | O(1) |

|  |  |
| --- | --- |
| Bước phác thảo của đoạn chương trình in đường chéo ứng với hàng i song song với đường chéo chính như sau:  1. for (i=2; i<=n; i++)  1.1. In đường chéo ứng với hàng i;  Bước tiếp theo để thực hiện câu lệnh 1.1 theo phương pháp tinh chỉnh từng bước nào dưới đây là đúng? | |
| A | for (j=1; j<=n-i+1; j++)  printf(”%6d”, a[i + j - 1, j]);  printf(”\n”); |
| B | for (i=1; i<=n-j+1; i++)  printf(”%6d”, a[i + j - 1, j]);  printf(”\n”); |
| C | Duyệt các phần tử theo cột  printf(”%6d”, a[i + j - 1, j]); |
| D | Duyệt các phần tử theo dòng  printf(”%6d”, a[i + j - 1, j]); |

|  |  |
| --- | --- |
| Bước phác thảo của đoạn chương trình in đường chéo ứng với cột j song song với đường chéo chính như sau:  1. for (i=2; i<=n; i++)  1.1. In đường chéo ứng với cột j;  Bước tiếp theo để thực hiện câu lệnh 1.1 theo phương pháp tinh chỉnh từng bước nào dưới đây là đúng? | |
| A | for (j=1; j<=n-i+1; j++)  printf(”%6d”, a[i, j+i - 1]);  printf(”\n”); |
| B | for (i=1; i<=n-j+1; i++)  printf(”%6d”, a[i, j+i - 1]);  printf(”\n”); |
| C | Duyệt các phần tử theo dòng  printf(”%6d”, a[i, j+i - 1]); |
| D | Duyệt các phần tử theo cột  printf(”%6d”, a[i, j+i - 1]); |

|  |  |
| --- | --- |
| Áp dụng phương pháp tinh chỉnh từng bước xây dựng thuật toán: nhập một dãy các số nguyên, tính trung bình nhân các số lẻ, dương, chia hết cho 5. Bước phác thảo đầu tiên nào dưới đây là đúng? | |
| A | 1. Nhập n, các phần tử trong mảng  2. for (i=1; i<=n; i++)  2.1. Tính trung bình nhân các số lẻ, dương, chia hết cho 5 |
| B | 1. Nhập n, các phần tử trong mảng  2. Tính tích và đếm các số lẻ , dương, chia hết cho 5  3. Tính trung bình nhân |
| C | 1. Nhập n, các phần tử trong mảng  2. Tính trung bình nhân các số lẻ, dương, chia hết cho 5 |
| D | 1. Nhập n, các phần tử trong mảng  2. for (i=1; i<=n; i++)  2.1. Tính tích và đếm các số lẻ, dương, chia hết cho 5  3. Tính trung bình tích |

|  |  |
| --- | --- |
| Trong các nguyên tắc dưới đây, nguyên tắc nào thực hiện phương pháp thiết kế bằng phương pháp tham lam? | |
| A | Đi từ những vấn đề cụ thể trước rồi ghép chúng lại thành những vấn đề lớn hơn |
| B | Chọn cái tốt nhất trước, sau đó cải tiến chúng thành những cái tốt hơn. |
| C | Tìm nghiệm của một bài toán bằng cách xem xét tất cả các phương án có thể |
| D | Đi từ các vấn đề tổng quát đến chi tiết. |

|  |  |
| --- | --- |
| Trong các nguyên tắc dưới đây, nguyên tắc nào thực hiện phương pháp thiết kế kiểu Bottom - Up? | |
| A | Tìm nghiệm của một bài toán bằng cách xem xét tất cả các phương án có thể |
| B | Đi từ các vấn đề tổng quát đến chi tiết. |
| C | Đi từ những vấn đề cụ thể trước rồi ghép chúng lại thành những vấn đề lớn hơn |
| D | Chọn cái tốt nhất trước, sau đó cải tiến chúng thành những cái tốt hơn. |

|  |  |
| --- | --- |
| Cho thuật toán sau:  int LinearSearch(float M[], int N, float X)  { int k = 0;  while ((k<N)&&(M[k] != X )) // N+1 lần  k++;  if (k<N) return (k);  return (-1);  }  Trong trường hợp xấu nhất khi không tìm thấy phần tử nào có giá trị bằng X, thì số phép gán và số phép so sánh bằng bao nhiêu? | |
| A | Số phép gán = 2  Số phép so sánh = 2N + 2 |
| B | Số phép gán = 2  Số phép so sánh = N + 2 |
| C | Số phép gán = 1  Số phép so sánh = N + 1 |
| D | Số phép gán = 1  Số phép so sánh = 2N + 1 |

|  |  |
| --- | --- |
| Trong các nguyên tắc dưới đây, nguyên tắc nào thực hiện phương pháp thiết kế kiểu Top - Down? | |
| A | Thiết kế bài toán theo cấu trúc hình cây. |
| B | Đi từ các vấn đề tổng quát đến chi tiết. |
| C | Đi từ những vấn đề cụ thể trước rồi ghép chúng lại thành những vấn đề lớn hơn |
| D | Chọn cái tốt nhất trước, sau đó cải tiến chúng thành những cái tốt hơn. |

|  |  |
| --- | --- |
| Giả sử trong bài toán sắp xếp dãy số nguyên thành dãy tăng dần đã có bước phác thảo: Từ dãy các số nguyên chưa được sắp, chọn ra số nhỏ nhất, đặt nó vào đầu dãy đã được sắp xếp, bước tinh chỉnh tiếp theo nào dưới đây cho bài toán này là đúng? | |
| A | for (i=1; i<=n; i++)  - Xét từ ai đến an để tìm ra số nhỏ nhất |
| B | - Xét từ ai đến an để tìm ra số nhỏ nhất  - Đổi chỗ giữa ai và aj |
| C | for (i=1; i<=n; i++)  - Xét từ ai đến an để tìm ra số nhỏ nhất  - Đổi chỗ giữa ai và aj |
| D | for (i=1; i<n; i++)  for (j=i+1; j<=n; j++)  if (a[i] > a[j])  đổi chỗ (a[i], a[j]); |

|  |  |
| --- | --- |
| Phác thảo nào dưới đây để xây dựng hàm kiểm tra xem một số n có phải là số hoàn hảo hay không? Biết rằng số hoàn hảo là số có tổng các ước bằng chính số đó (Ví dụ: 6 = 1 + 2 + 3) | |
| A | 1. Nhập n  2. Cho i chạy từ 1 đến n  2.1. Nếu n % i ==0 thì cộng dồn các i lại  3. Kiểm tra nếu s = n 🡪 đó là số hoàn hảo |
| B | 1. Nhập n  2. Cho i chạy từ 1 đến n/2  2.1. Nếu n % i ==0 thì cộng dồn các i lại  3. Kiểm tra nếu s = n 🡪 đó là số hoàn hảo |
| C | 1. Nhập n  2. Cho i chạy từ 1 đến n/2  2.1. Nếu n % i ==0 thì cộng dồn các i lại |
| D | 1. Nhập n  2. Cho i chạy từ 1 đến n/2  2.1. Nếu n % i ==0 thì cộng dồn các i lại |

|  |  |
| --- | --- |
| Đoạn chương trình sau mô tả thuật toán gì?  S = 0;  for (i = 1; i<=n/2; i++)  if ( i \* i ==n)  S = 1; | |
| A | Kiểm tra số nguyên tố |
| B | Kiểm tra số chính phương |
| C | Kiểm tra số hoàn hảo |
| D | Kiểm tra số chẵn, dương |

|  |  |
| --- | --- |
| Phác thảo sau mô tả thuật toán gì?  1. Gán s=1;  2. Cho i chạy từ 1 đến n/2  3. Kiểm tra nếu n chia hết cho i thì gán s=0; | |
| A | Kiểm tra số nguyên tố |
| B | Kiểm tra số chính phương |
| C | Kiểm tra số hoàn hảo |
| D | Kiểm tra số chẵn |

|  |  |
| --- | --- |
| Khi ta nói thời gian thực hiện của một chương trình là T(n) = Cn thì có nghĩa là chương trình ấy cần bao nhiêu chỉ thị thực thi? | |
| A | Cần C chỉ thị thực thi |
| B | Cần T(n) chỉ thị thực thi |
| C | Cần n chỉ thị thực thi |
| D | Cần Cn chỉ thị thực thi |

|  |  |
| --- | --- |
| Thường ta coi T(n) là thời gian thực hiện chương trình trong trường hợp xấu nhất trên dữ liệu vào có kích thước n, tức T(n) là gì? | |
| A | Thời gian nhỏ nhất để thực hiện chương trình đối với mọi dữ liệu vào có cùng kích thước T |
| B | Thời gian nhỏ nhất để thực hiện chương trình đối với mọi dữ liệu vào có cùng kích thước n |
| C | Thời gian lớn nhất để thực hiện chương trình đối với mọi dữ liệu vào có cùng kích thước n. |
| D | Thời gian lớn nhất để thực hiện chương trình đối với mọi dữ liệu vào có cùng kích thước T |

|  |  |
| --- | --- |
| Bước phác thảo của đoạn chương trình tính tích các số chẵn âm trong một dãy các số nguyên như sau:  - Xét từ a[1] đến a[n]  - Tìm các số chẵn, âm  - Tính tổng  Bước tiếp theo để thực hiện đoạn chương trình theo phương pháp tinh chỉnh từng bước nào dưới đây là đúng? | |
| A | for (i=1; i<=n; i++)  if (a[i] %2==0 && a[i]<0)  s =s\*a[i]; |
| B | for (i=1; i<=n; i++)  if (a[i] %2==0 || a[i]<0)  s=s\*a[i]; |
| C | Duyệt các phần tử  if (a[i] %2==0 && a[i]<0)  s=s\*a[i]; |
| D | Duyệt các phần tử  if (a[i] %2==0 && a[i]<0)  Tính tích |

|  |  |
| --- | --- |
| Cho một dãy số nguyên, đếm xem có bao nhiêu số chẵn. Bước phác thảo đầu tiên theo phương pháp tinh chỉnh từng bước nào dưới đây là đúng? | |
| A | - Duyệt các phần tử trong mảng  - Đếm các số chẵn |
| B | for(i=1; i<=n; i++)  Đếm các số chẵn trong mảng |
| C | - Nhập n và các phần tử của mảng  - Đếm các số chẵn trong mảng |
| D | Duyệt các phần tử trong mảng  if(a[i]%2 == 0) dem++; |

|  |  |
| --- | --- |
| Khi nói đến độ phức tạp của giải thuật là ta muốn nói đến cái gì? | |
| A | Kết quả thu được sau khi thực hiện của chương trình |
| B | Hiệu quả của thời gian thực hiện của chương trình |
| C | Các bước tính toán trong quá trình thực hiện chương trình |
| D | Tổng số câu lệnh trong giải thuật |

|  |  |
| --- | --- |
| Khi đánh giá độ phức tạp của giải thuật cho các đoạn chương trình, thời gian thực hiện tuần tự của hai đoạn chương trình P1, P2 được thực hiện theo qui tắc nào? | |
| A | Qui tắc cộng |
| B | Qui tắc nhân |
| C | Qui tắc trừ |
| D | Qui tắc chia |

|  |  |
| --- | --- |
| Khi đánh giá độ phức tạp của giải thuật cho các đoạn chương trình, thời gian thực hiện lồng nhau của hai đoạn chương trình P1, P2 được thực hiện theo qui tắc nào? | |
| A | Qui tắc cộng |
| B | Qui tắc nhân |
| C | Qui tắc trừ |
| D | Qui tắc chia |

|  |  |
| --- | --- |
| Cho đoạn chương trình  1. S=1;  2. for (i=0; i<n; i++)  { p=1;  for (j=1; j<=n; j++) p= p\*x/j;  S= S+p;  }  có độ phức tạp tính toán là bao nhiêu? | |
| A | O(log2n) |
| B | O(nlog2n) |
| C | O(3n) |
| D | O(n2) |

|  |  |
| --- | --- |
| Đoạn chương trình sau có độ phức tạp tính toán là bao nhiêu?  1. Sum = 0;  2. for (i = 1; i<=n; i++)  { printf(“\nNhập số thứ %d”,i);  scanf(“%d”, &x);  Sum = Sum + x;  } | |
| A | O(n) |
| B | O(log2n) |
| C | O(nlog2n) |
| D | O(n2) |

|  |  |
| --- | --- |
| Xác định thời gian thực hiện của chương trình là xác định cái gì? | |
| A | Xác định các bước thực hiện của giải thuật |
| B | Xác định sai sót trong kết quả của giải thuật |
| C | Xác định độ phức tạp của giải thuật |
| D | Xác định hiệu quả của chương trình |

|  |  |
| --- | --- |
| Thời gian thực hiện lệnh rẽ nhánh ***if*** được xác định như thế nào? | |
| A | Là thời gian thực hiện lệnh điều kiện |
| B | Là thời gian thực hiện lệnh điều kiện cộng với thời gian kiểm tra điều kiện |
| C | Là thời gian thực hiện việc kiểm tra điều kiện |
| D | Được tính là O(1) |

|  |  |
| --- | --- |
| Thời gian thực hiện lệnh rẽ nhánh ***if ... else*** được xác định như thế nào? | |
| A | Là thời gian thực hiện lệnh điều kiện cộng với thời gian kiểm tra điều kiện |
| B | Là thời gian thực hiện lệnh điều kiện |
| C | Là thời gian kiểm tra điều kiện cộng với thời gian lớn nhất của 1 trong 2 lệnh rẽ nhánh |
| D | Là thời gian kiểm tra điều kiện cộng với thời gian nhỏ nhất nhất của 1 trong 2 lệnh rẽ nhánh |

|  |  |
| --- | --- |
| Thời gian thực hiện các lệnh đọc, gán, ghi là bao nhiêu? | |
| A | O(n2) |
| B | O(log2n) |
| C | O(1) |
| D | O(n) |

|  |  |
| --- | --- |
| Thời gian thực hiện của nhóm lệnh tuần tự được xác định như thế nào? | |
| A | Bằng tổng thời gian thực hiện của các lệnh |
| B | Bằng tích thời gian thực hiện của các lệnh |
| C | Bằng thời gian thực hiện của lệnh bé nhất |
| D | Bằng thời gian thực hiện của lệnh lớn nhất |

|  |  |
| --- | --- |
| Trong các đoạn chương trình dưới đây, đoạn chương trình nào có độ phức tạp là O(1)? | |
| A | for(i=1; i<=n; i++)  if(a[i]<min) min = a[i]; |
| B | if(a<b) min = a; |
| C | printf(“Nhap n=”); |
| D | for(i=1; i<=n; i++)  dem++; |

|  |  |
| --- | --- |
| Trong các đoạn chương trình sau, đoạn chương trình nào có độ phức tạp là O(n2)? | |
| A | for(i=1; i<=n; i++)  for(j=i+1; j<=n; j++)  if(a[i]<a[j]) doicho(a[i], a[j]); |
| B | if(a<b) min = a; |
| C | printf(“Nhap n=”); |
| D | for(i=1; i<=n; i++)  dem++; |

|  |  |
| --- | --- |
| Trong các đoạn chương trình sau, đoạn chương trình nào có độ phức tạp là O(n)? | |
| A | for(i=1; i<=n; i++)  for(j=i+1; j<=n; j++)  if(a[i]<a[j]) doicho(a[i], a[j]); |
| B | if(a<b) min = a; |
| C | printf(“Nhap n=”); |
| D | for(i=1; i<=n; i++)  dem++; |

|  |  |
| --- | --- |
| Đánh giá độ phức tạp của đoạn chương trình sau theo kí pháp O cho kết quả bằng bao nhiêu?  1. for (i = 0; i<p;i++) c[i] = a[i] + b[i];  2. if (p < m)  2.1. for (i = p + 1; i<n; i++) c[i] = a[i]  3. else  3.1. for (i = p + 1; i<n; i++) c[i] = b[i];  Trong đó m = n; p>=m,n | |
| A | O(n-(p+1)) |
| B | O(p) |
| C | O(p) + O(n-(p+1)) |
| D | O(2n+p) |

|  |  |
| --- | --- |
| Đánh giá độ phức tạp của đoạn chương trình sau theo kí pháp O cho kết quả bằng bao nhiêu?  1. p = m + n;  2. for (i = 0; i<p;i++) c[i] = 0;  3. for (i = 0; i<m;i++)  3.1. for (j= 0; j<n;j++)  3.1.1. c[i + j] = c[i + j] + a[i] \* b[j]; | |
| A | O(m.n+ p + 1) |
| B | O(p) |
| C | O(m.n) |
| D | O(m.n+p) |

|  |  |
| --- | --- |
| Cho đoạn chương trình:  if (a+b>c && b+c>a && c+a>b && a>0 && b>0 && c>0)  {  P = (a+b+c)/2;  S = sqrt(P\*(P-a)\*(P-b)\*(P-c));  }  else  printf(“\nDay khong phai la ba canh của tam giac”);  Đoạn chương trình trên có độ phức tạp tính toán là bao nhiêu? | |
| A | T(n) = 8 |
| B | T(n) = 9 |
| C | T(n) = 1 |
| D | T(n) = 6 |

|  |  |
| --- | --- |
| Cho đoạn chương trình:  for (i=1; i<n; i++)  for (j = i; j<=n; j++)  if (a[i]< a[j])  { tg = a[i];  a[i] = a[j];  a[j] = tg;  }  Trong trường hợp xấu nhất, đoạn chương trình trên có độ phức tạp tính toán là bao nhiêu? | |
| A | O(n) |
| B | O(n2) |
| C | O(4n) |
| D | O(4n2) |

|  |  |
| --- | --- |
| Cho đoạn chương trình:  1. for (j = i+1; j<N, j++)  1.1. if (x[j] < x[k]) k= j;  2. if (k!= i)  {  2.1. temp = x[i];  2.2. x[i] = x[k] ;  2.3. x[k] = temp;  }  Trong trường hợp xấu nhất, đoạn chương trình trên có độ phức tạp tính toán là bao nhiêu? | |
| A | O(n+4) |
| B | O(max(2n, 4)) = O(n) |
| C | O(4) |
| D | O(2n) |

|  |  |
| --- | --- |
| Bước đầu tiên khi thực hiện phương pháp tinh chỉnh từng bước để thiết kế giải thuật là gì? | |
| A | Thể hiện giải thuật bằng ngôn ngữ lập trình, phản ánh ý chính của công việc cần làm |
| B | Thể hiện giải thuật bằng sơ đồ khối, phản ánh ý chính của công việc cần làm |
| C | Thể hiện giải thuật bằng ngôn ngữ tự nhiên, phản ánh ý chính của công việc cần làm |
| D | Thể hiện giải thuật bằng ngôn ngữ giả mã, phản ánh ý chính của công việc cần làm |

|  |  |
| --- | --- |
| Trong qui tắc cộng, nếu T1(n) và T2(n) là thời gian thực hiện của hai đoạn chương trình P1 và P2; và T1(n)=O(f(n)), T2(n)=O(g(n)) thì thời gian thực hiện của đoạn hai chương trình đó nối tiếp nhau là bao nhiêu? | |
| A | T(n)=O(min(f(n),g(n))) |
| B | T(n)=O(max(f(n),g(n))) |
| C | T(n) = O(f(n).g(n)) |
| D | T=O(max(f(n),g(n))) |

|  |  |
| --- | --- |
| Lệnh gán x:=1 tốn một hằng thời gian hay O(1), Lệnh đọc dữ liệu scanf(“%d”,&x) tốn một hằng thời gian hay O(1).Vậy thời gian thực hiện cả hai lệnh trên nối tiếp nhau là bao nhiêu? | |
| A | O(max(1,1))=O(1) |
| B | O(1+1))=O(2) |
| C | O(min(1,1))=O(1) |
| D | O(min(0,0))=O(0) |

|  |  |
| --- | --- |
| Trong qui tắc nhân, Nếu T1(n) và T2(n) là thời gian thực hiện của hai đoạn chương trình P1và P2 và T1(n) = O(f(n)), T2(n) = O(g(n)) thì thời gian thực hiện của đoạn hai đoạn chương trình đó lồng nhau là bao nhiêu? | |
| A | T(n)=O(min(f(n),g(n))) |
| B | T(n)=O(max(f(n),g(n))) |
| C | T(n) = O(f(n).g(n)) |
| D | T(n2) = O(f(n).g(n)) |

|  |  |
| --- | --- |
| Thời gian thực hiện T(n) của thủ tục sắp xếp “nổi bọt” là bao nhiêu?  void Bubble\_sort()  { int i=0, j, k, temp;  do { k = 1;//k biến không\_chuyển vị trí  for (j=n-1; j>=i; j--)  { if (a[j-1] > a[j])  { temp = a[j-1]; a[j-1] = a[j];  a[j] = temp; k = 0;  }  }  i++  } while (k || (i==n));  } | |
| A | T(n) = [n(n -1 )] / 2 = O(n) |
| B | T(n) = [n(n -1 )] / 2 = O(n2) |
| C | T(n) = [n(n +1 )] / 2 = O(n) |
| D | Top of Form  T(n) = [n(n +1 )] / 2 = O(n2) |

|  |  |
| --- | --- |
| Khái niệm về “cấp độ lớn của của thời gian thực hiện giải thuật” còn được gọi là gì? | |
| A | Chương trình thể hiện giải thuật đó |
| B | Độ phức tạp về thời gian thực hiện giải thuật |
| C | Độ phức tạp của giải thuật |
| D | Độ phức tạp về thời gian của giải thuật |

|  |  |
| --- | --- |
| Giả sử thời gian thực hiện một giải thuật là T(n) = Cn2 (với C là hằng số), thì ta nói độ phức tạp tính toán của giải thuật này có cấp là bao nhiêu? | |
| A | O(C) |
| B | O(Cn2) |
| C | O(n2) |
| D | O(Cn) |

|  |  |
| --- | --- |
| Khi đánh giá độ phức tạp của giải thuật, những bài toán thuộc lớp nào có độ phức tạp là O(log2n)? | |
| A | Khi giải một bài toán bằng cách tách nó thành các bài toán con nhỏ hơn |
| B | Thời gian chạy chương trình tiến chậm khi N lớn dần |
| C | Các chỉ thị của các chương trình đều được thực hiện một lần hay nhiều nhất chỉ một vài lần |
| D | Khi thời gian chạy của một thuật toán là bậc hai |

|  |  |
| --- | --- |
| Khi đánh giá độ phức tạp của giải thuật, những bài toán thuộc lớp nào có độ phức tạp là hằng số? | |
| A | Các chỉ thị của các chương trình đều được thực hiện một lần hay nhiều nhất chỉ một vài lần |
| B | Thời gian chạy chương trình tiến chậm khi N lớn dần |
| C | Khi thời gian chạy của một chương trình là tuyến tính |
| D | Khi thời gian chạy của một thuật toán là bậc hai |

s

Cho dãy khóa 42,23,74,11,65,58,94,36. Dùng phương pháp sắp xếp nhanh và lấy phần tử đầu

1. làm chốt. Sau bước 1, sau chốt(42) có bao nhiêu phầntử? 4 phần tử

Về nguyên tắc vun đống, trong các điều sau đây điều nào sai?

Nếu đống có 10 nút thì nút a5 không cần điều chỉnh

Cho 2 dãy khóa đã sắp xếp:

A = 17, 42, 50, 70, 81

B = 8, 19, 30, 41, 53, 84

Xếp thứ tự 2 dãy trên thành dãy C theo kiểu trộn (merge sort), phần tử C[7] là khóa nào?

50

Cho 2 dãy khóa đã sắp xếp:

1. 3, 7, 15, 19, 21, 30
2. 2, 4, 9, 14, 17, 45

Xếp thứ tự 2 dãy trên thành dãy C theo kiểu trộn (merge sort). Phần tử C[9] là khóa nào?

19

Cho dãy khóa: 42, 23, 74, 11, 65, 58, 94, 36, 99, 87. Để sắp xếp theo kiểu hòa nhập, ta có thể tạo thành ít nhất bao nhiêu đường chạy?

6

Cho dãy khóa 42, 23, 74, 11, 65, 58, 94, 36. Lần lượt đưa dãy khóa trên vào cây nhị phân tìm kiếm. bây giờ ta muốn tìm kiếm xem trong dãy khóa trên có khóa 60 không thi phải làm bao nhiêu phép so sánh?

4

Cho dãy khóa 42, 23, 74, 11, 65, 58

Dùng phương pháp sắp xếp kiểu chọn (selection sort), sau 3 bước dãy có dạng nào?

11, 23, 42, 74, 65, 58

Cho dãy khóa 42, 23, 74, 11, 65, 58

Dùng phương pháp sắp xếp kiểu chèn (insert sort), sau 3 bước dãy có dạng nào?

11, 23, 42, 74, 65, 58

Cho dãy khóa 42, 23, 74, 11, 65, 58, 94, 36

Sắp xếp dãy khóa theo kiểu nổi bọt (buble sort), sau mấy bước phần tử 36 giữ vị trí ổn định (không thay đổi vị trí nữa) của nó ?

3

Cho dãy khóa 42, 23, 74, 11, 65, 58, 94, 36

Sắp xếp dãy khóa theo kiểu nổi bọt (buble sort), sau mấy bước phần tử 11 giữ vị trị ổn định (không thay đổi vị trí nữa) của nó ?

1

Cho dãy khóa 42, 23, 74, 11, 65, 58, 94, 36, 99, 87

Sắp xếp dãy khóa theo kiểu nổi bọt (buble sort), sau bước thứ 1, khóa 87 về vị trí thứ mấy?

9

Cho dãy khóa 42, 23, 74, 11, 65, 58, 94, 36, 99, 87

Sắp xếp dãy khóa theo kiểu nổi bọt ( buble sort) theo thứ tự tăng dần, sau bước thứ 1, ba vị trí cuối cùng là các khóa nào?

94,87,99

Cho dãy khóa 42, 23, 74, 11, 65, 58, 94, 36, 99, 87

Sắp xếp dãy khóa theo kiểu nhanh ( quick sort), sau bước thứ 1, ba vị trí đầu tiên là các khóa nào?

11,23,36

Cho dãy khóa 42, 23, 74, 11, 65, 58, 94, 36, 99, 87

Sắp xếp dãy khóa theo kiểu chèn (insert sort), sau 3 bước ba vị trí cuối cùng là các khóa nào?

36,99,87

Cho 2 dãy đã sắp xếp:

1. 15, 17, 19, 21, 30, 48
2. 7, 9, 16, 22, 47

Sắp xếp 2 dãy này theo kiểu hòa nhập và dãy C. Phần tử C[7] của dãy C là khóa nào?

21

Void sort(int a[], int n)

{

Int i,j,t;

For(i=n;j>0;i++)

For(j=1;j<1;j++)

If(a[j-1] >a[j])

{

T=a[j-1];

A[j-1]=a[j];

A[j]=t;

}

}

Sắp xếp nổi bọt

Giả sử dãy cần sắp xếp có *n* phần tử. Khi tiến hành từ trên xuống, ta so sánh hai phần tử đầu, nếu phần tử đứng trước lớn hơn phần tử đứng sau thì đổi chỗ chúng cho nhau. Tiếp tục làm như vậy với cặp phần tử thứ hai và thứ ba và tiếp tục cho đến cuối tập hợp dữ liệu, nghĩa là so sánh (và đổi chỗ nếu cần) phần tử thứ *n-1* với phần tử thứ *n*. Sau bước này phần tử cuối cùng chính là phần tử lớn nhất của dãy. Đây là thuật toán sắp xếp nào?

Sắp xếp nổi bọt

Chọn phần tử nhỏ nhất trong n phần tử ban đầu, đưa phần tử này về vị trí đúng là đầu tiên của dãy hiện hành. Sau đó không quan tâm đến nó nữa, xem dãy hiện hành chỉ còn n-1 phần tử của dãy ban đầu, bắt đầu từ vị trí thứ 2. Lặp lại quá trình trên cho dãy hiện hành đến khi dãy hiện hành chỉ còn 1 phần tử. Dãy ban đầu có n phần tử, vậy tóm tắt ý tưởng thuật toán là thực hiện n-1 lượt việc đưa phần tử nhỏ nhất trong dãy hiện hành về đúng vị trí đúng ở đầu dãy. Đây là thuật toán sắp xếp nào?

Sắp xếp chọn

Phân chia danh sách được sắp thành hai danh sách con. Chia danh sách cần sắp xếp a[1…n] thành hai danh sách con có kích thước tương đối bằng nhau nhờ chỉ số đứng giữa danh sách, sắp xếp nhanh chia nó thành hai danh sách bằng cách so sánh từng phần tử của danh sách với một phần tử được chọn được gọi là phần tử chốt. Những phần tử nhỏ hơn hoặc bằng phần tử chốt được đưa về phía trước và nằm trong danh sách con thứ nhất, các phần tử lớn hơn chốt được đưa về phía sau và thuộc danh sách đứng sau. Cứ tiếp tục chia như vậy cho tới khi các danh sách con đều có độ dài bằng 1. Đây là thuật toán sắp xếp nào?

Sắp xếp nhanh

Giả sử có hai danh sách đã được sắp xếp a[1…m] và b[1…n]. Ta có thể trộn chúng lại thành một danh sách mới c[1…m+n] được sắp xếp theo cách sau:

* So sánh hai phần tử đứng đầu của hai danh sách, lấy phần tử nhỏ hơn cho vào danh sách mới. Tiếp tục như vậy cho tới khi một trong hai danh sách là rỗng.
* Khi một trong hai danh sách là rỗng ta lấy phần còn lại của danh sách kia cho vào cuối danh sách mới.

Đây là thuật toán sắp xếp nào?

Sắp xếp trộn

Độ phức tạp của thuật toán sắp xếp nổi bọt là bằng bao nhiêu?

O(n2)

Độ phức tạp của thuật toán sắp xếp chọn là bằng bao nhiêu?

O(n2)

Độ phức tạp của thuật toán sắp xếp nhanh trong trường hợp xấu nhất là bằng bao nhiêu?

O(n2)

Trong trường hợp xấu nhất, thuật toán sắp xếp chèn sử dụng bao nhiêu phép so sánh?

n-1 phép so sánh

Trong trường hợp xấu nhất, thuật toán sắp xếp chèn sử dụng bao nhiêu phép hoán vị?

N2/2 phép hoán vị

Trong trường hợp xấu nhất, thuật toán sắp xếp chọn sử dụng bao nhiêu phép so sánh?

N2/2 phép so sánh

Trong trường hợp xấu nhất, thuật toán sắp xếp chọn sử dụng bao nhiêu phép hoán vị?

0 phép hoán vị

Trong trường hợp tốt nhất, thuật toán sắp xếp chèn sử dụng bao nhiêu phép so sánh?

N2/8 phép so sánh

Trong trường hợp tốt nhất, thuật toán sắp xếp chèn sử dụng bao nhiêu phép hoán vị?

0 phép hoán vị

* Bước 1: i=1
* Bước 2: Tìm phần tử a[min] nhỏ nhất trong dãy hiện hành từ a[i] đến a[n]
* Bước 3: Hoán vị a[min] và a[i]
* Bước 4: Nếu i<=n-1 thì i=i+1; lặp lại bước 2
* Ngược lại: Dừng. n-1 phần tử đã nằm đúng vị trí.

Đây các bước của thuật toán sắp xếp nào?

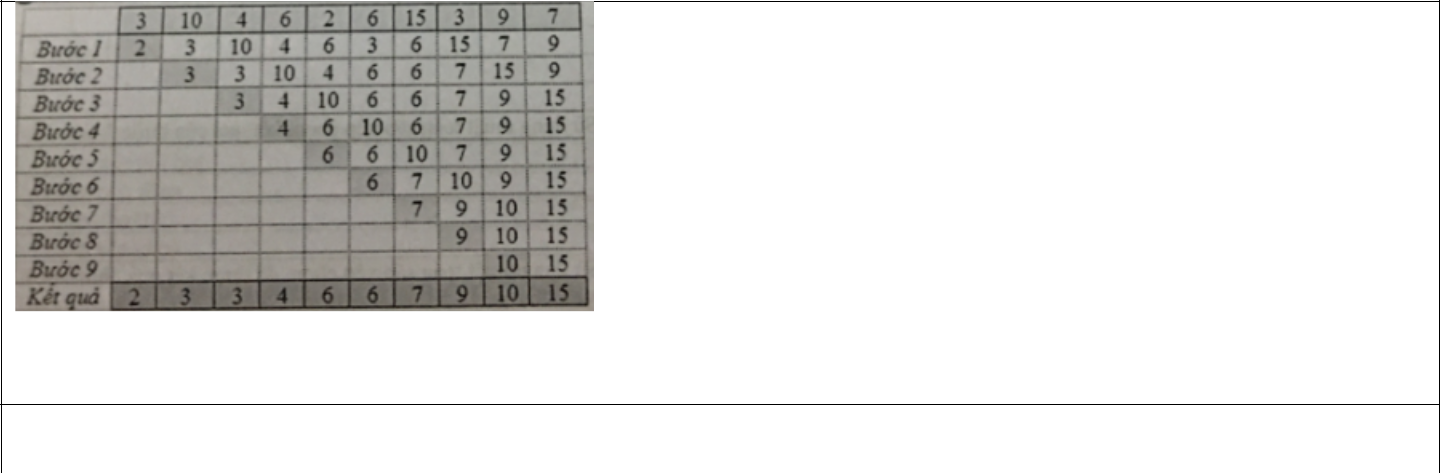
Sắp xếp chọn

Thuật toán sắp xếp nào tốn thời gian gần như bằng nhau đối với mảng đã được sắp xếp cũng như mảng chưa được sắp xếp?

Sắp xếp chọn

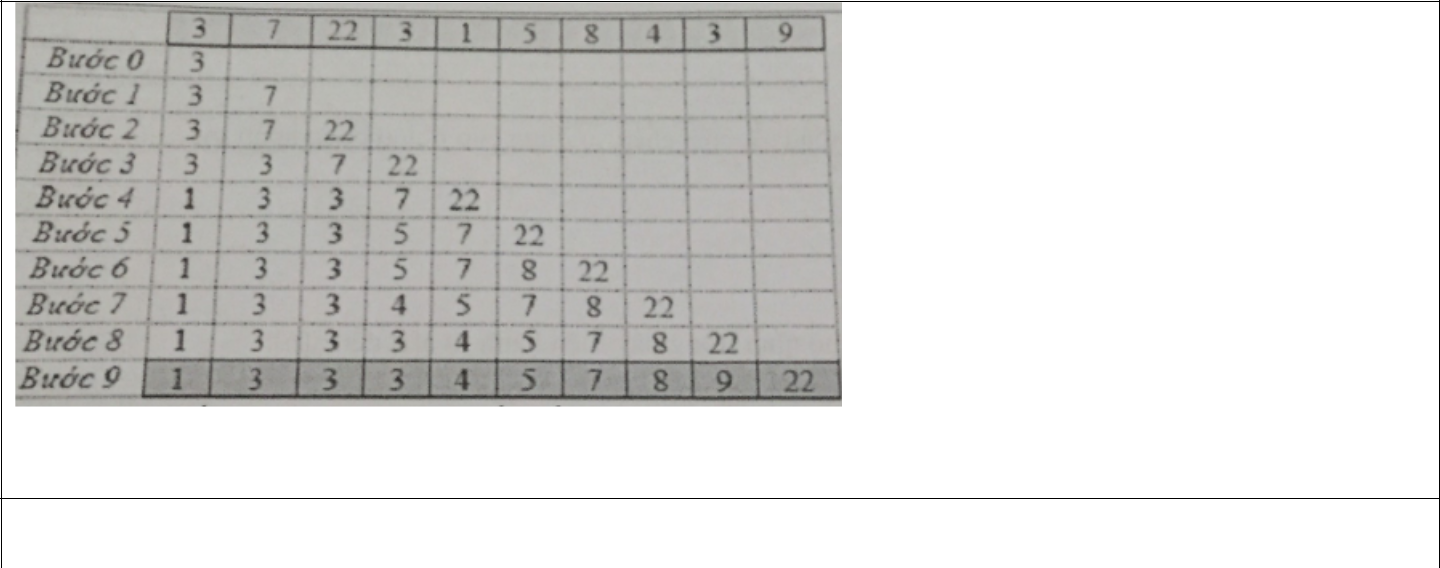
Phần tử chốt được sử dụng trong thuật toán sắp xếp nào?

Sắp xếp nhanh



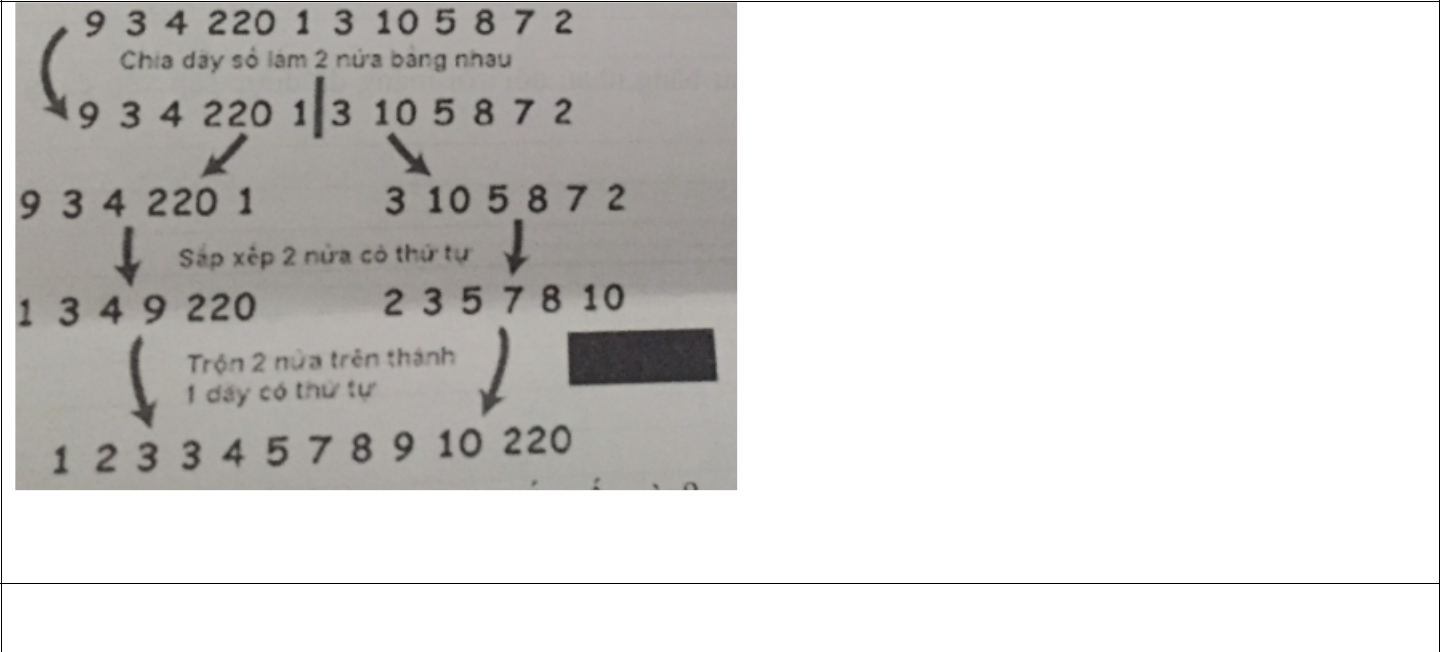
Trong hình là kết quả của thuật toán sắp xếp nào?

Sắp xếp nổi bọt



Trong hình là kết quả của thuật toán sắp xếp nào?

Sắp xếp chèn



Trong hình là kết quả của thuật toán sắp xếp nào?

Sắp xếp trộn

Cho dãy a= (12, 2, 8, 5, 1, 6, 4, 15). Sử dụng thuật toán sắp xếp chọn, để được kết quả như sau: 1 2 4 5 6 12 8 15. Thuật toán đã chạy đến bước thứ mấy?

5

Cho dãy a= (12, 2, 8, 5, 1, 6, 4, 15). Sử dụng thuật toán sắp xếp chọn. Dãy được sắp xếp sau bao nhiêu bước

7

Void sort(int c[], int(n){

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

{

int m = i;

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

if(c[j] < c[m])

m = j;

int t = c[i];

c[i] = c[m];

c[m] = t;

}

Đoạn code trên thuộc thuật toán sắp xếp nào?

Sắp xếp chọn

Void sort(int a[],int n)

{

int x,j;

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

{

x=a[i];

j = j-1;

while((j>=0) && (a[j] > x))

{

a[j+1] = a[j];

j--;

}

a[j+1] = x;

}

}

Đoạn code trên thuộc thuật toán sắp xếp nào?

Sắp xếp chèn

* Bước 1: i=1
* Bước 2: Lần lượt so sánh và đổi chỗ (nếu cần) từ phải sang trái đối với các phần tử từ X[n] đến X[i]
* Bước 3: i=i+1
* Bước 4:
  + Nếu i<n, quay lại bước 2.
  + Ngược lại, dừng, dãy đã cho đã sắp xếp đúng vị trí.

Đây là các bước của thuật toán sắp xếp nào?

Sắp xếp nổi bọt

Cho dãy khóa {7, 1, 5, 13, 2, 9, 1, 5, 7 } dùng phương pháp sắp xếp nội bọt, sau 2 bước dãy có dạng nào

D. 1, 1, 5, 7, 13, 7, 5, 9, 2

Cho dãy khóa {7, 1, 5, 13, 2, 9, 1, 5, 7 } dùng phương pháp sắp xếp chèn, sau 3 bước dãy có dạng nào

D. 1, 5, 7, 13, 2, 9, 1, 5, 7

Cho dãy khóa {7, 1, 5, 13, 2, 9, 1, 5, 7 } dùng phương pháp sắp xếp chọn, sau 3 bước dãy có dạng nào

B. 1, 1, 2, 13, 5, 9, 7, 5, 7

Cho dãy khóa {7, 1, 5, 13, 2, 9, 1, 5, 7 } dùng phương pháp sắp xếp nhanh, sau 1 bước dãy có dạng {[1,1,5,7,2,5], [7], [9,13]}, sau 2 bước dãy có dạng nào?

B. {[1],[1,5,7,2,5], [7], [9], [13]}

Cho dãy khóa {3, 7, 9, 5, 1, 8, 6 } dùng phương pháp sắp xếp chèn, sau 2 bước dãy có dạng nào

A.3,7,9,5,1,8,6

Cho dãy khóa {3, 7, 9, 5, 1, 8, 6 } dùng phương pháp sắp xếp nổi bọt, sau 3 bước dãy có dạng nào

A.1,3,5,6,7,9,8

Cho dãy khóa {3, 7, 9, 5, 1, 8, 6 } dùng phương pháp sắp xếp chọn, sau 2 bước dãy có dạng nào

D.1,3,9,5,7,8,6

Cho dãy khóa X= {3, 7, 9, 5, 1, 8, 6}, dùng phương pháp sắp xếp chèn, sau 2 bước dãy có dạng, phần tử thứ 2 ở vị trí nào ?

B. Thứ 2

Đối với thuật toán sắp xếp chèn cho dãy các phần tử sau:{11, 16, 12, 75, 51, 54, 5, 73, 36, 52, 98}. Cần thực hiện bao nhiêu lần chèn các phần tử vào dãy con đã có thứ tự tang dần đứng đầu dãy M để sắp xếp mảng M có thứ tự tang dần ?

B. 10 lần

Đối với thuật toán sắp xếp chèn cho dãy các phần tử sau:{16, 60, 2, 25, 15, 45, 5, 30, 33, 20}. Cần thực hiện bao nhiêu lần chọn lựa các phần tử nhỏ nhất để sắp xếp mảng M có thứ tự tang dần ?

C. 9 lần

Ý tưởng phương pháp sắp xếp tăng dần là ?

A. Chọn phần tử bé nhất xếp vào vị trí thứ nhất bằng cách đổi chỗ phần tử bé nhất với phần tử thứ nhất; tương tự đối với phần tử nhỏ thứ hai, ba,….

Ý tưởng phương pháp sắp xếp nổi bọt là ?

D. Bắt đầu từ cuối dãy đến đầu dãy, ta lần lượt so sánh hai phần tử kế tiếp nhau, nếu phần tử nhỏ hơn đươc đứng vị trí trên.

Ý tưởng phương pháp sắp xếp chèn là ?

A. Lần lượt lấy phần tử của danh sách chèn vị trí thích hợp của nó trong dãy bằng cách đẩy phần tử lớn hơn xuống.

Ý tưởng phương pháp sắp xếp nhanh (Quick sort) là ?

A. Lần lượt chia dãy phần tử thành hay dãy con bởi một phần từ khóa ( dãy con trước khóa gồm các phần tử nhỏ hơn khóa và dãy còn lại gồm các phần tử lớn hơn khóa )

Ý tưởng phương pháp sắp xếp trộn (Merge sort) là ?

C. Phân đoạn dãy thành nhiều dãy con cho đến khi các dãy con chỉ có 1 phần tử thì dừng và lẫn lượt trộn hay dãy con thành dãy lớn hơn, cho đến khi thu được dãy ban đầu đã được sắp xếp

Ý tưởng phương pháp sắp xếp vun đống là ?

B. Lần lượt tạo đống cho cây nhị phân (phần tử gốc có giá trị lớn nhất), đổi chỗ phần tử gốc và phần tử cuối cùng, loại phần tử gốc cũ ra khỏi cây đưa vào dãy sắp xếp, sau đó vun đống lại dãy n-1 phần tử còn lại

Phương pháp sắp xếp nhanh (Quick sort) chính jlaf phương pháp

D. Phân đoạn

Cơ chế heap trong sắp xếp vun đống là ?

A. Cây nhị phân đầy đủ với tính chất giá trị của nút cha luôn lớn hơn giá trị hai nút con.

Trong giải thuật sắp xếp vun đống, ta có 4 thủ tục con (Insert - thêm 1 phần tử vào cây: Downheap- vun đống lại sau khi loại 1 phần tử khỏi heap, Upheap

* vun đống sau khi thêm một phần tử vào cây; Remove – loại 1 phần tử khỏi cây nhị phân). Để sắp xếp các phần tử trong dãy theo phương pháp vun đống, ta thực hiện 4 thủ tục trên thứ tự như thế nào ?

C. Insert – Upheap – Remove - Insert

Cho 1 mảng gồm 10 phần tử có Khóa là các số nguyên: 5,6,2,2,10,12,9,10,9 và 3. Dùng phương pháp sắp xếp đơn giản nào là nhanh chóng nhất

C. Sắp xếp chọn (Selection Sort)

Vấn đề chọn chốt trong thiết kế giải thuật sắp xếp phân đoạn, trong 2 phần tử có khóa khác nhau đầu tiên kể từ trái qua ta chọn khóa nào ?

C. Khóa lớn nhất

Trong thiết kế giải thuật sắp xếp phân đoạn, nếu mảng chỉ gồm một phần tử hay gồm nhiều phần tử có khóa bằng nhau thì chọn khóa nào làm chốt

A. Giải thuật không có chốt

Cho mảng gồm các phần tử có khóa là 6,6,6,6,6,6. Khi dung phương pháp sắp xếp phân đoạn, chốt của mảng là bao nhiêu?

D. Mảng này không có chốt

Cho mảng gồm có một phần tử có khóa là 6. Khi dung phương pháp sắp xếp phân đoạn, chốt của mảng là bao nhiêu ?

D. Mảng này không có chốt

Trong giải thuật sắp xếp phân đoạn (QuickSort), các bước để tiến hành sắp xếp mảng a[i]…a[j] là gì ?

A. Xác định chốt – Phân hoạch mảng đã cho thành hai mảng con a[i]…a[k-1] và a[k]…a[j] – sắp xếp mảng a[i]…a[k-1] ( Đệ quy ) – sắp xếp mảng a[k]…a[j] (Đệ quy)

Cho mảng gồm các phần tử có khóa là 6,6,7,5,7,4. Khi dung phương pháp sắp xếp phân đoạn, chốt của mảng là bao nhiêu ?

D. 7

Trong vấn đề phân hoạch, để phân hoạch mảng ta dung 2 ‘con nháy’ là gì ?

A. L và R trong đó L từ bên trái và R từ bên phải

Cho các dãy số: 1,2,3,4,5,6 (n=6); khi thực hiện giải thuật sắp xếp chèn để sắp dãy trên thành dãy không giảm, cần thực hiện bao nhiêu lần so sánh ?

C. 5

Đối với thuật toán sắp xếp chọn dãy các phần tử sau ( 10 phần tử )

16 50 3 24 15 45 5 33 30 20

Cần thực hiện mấy lần chọn lựa phần tử nhỏ nhất đêqr sắp xếp mảng M có thứ tự tang dần ?

C. 9 lần

Giả sử cần sắp xếp mảng M gồm 11 phần tử sau theo phương pháp sắp xếp chèn: 10 15 11 70 49 54 3 70 36 52 100

Cần thực hiện mấy lần chèn các phần tử vào dãy con đã có thứ tự tang dần đứng đầu dãy M đêqr sắp xếp mảng M theo thứ tự không giảm ?

A. 10 lần

Cho dãy khóa 42, 23, 74, 11, 65, 58

Dung phương pháp sắp xếp kiểu chọn (Selection sort), sau 2 bước dãy có dạng nào ?

C. 11,23,74,42,65,58

Cho dãy khóa 42, 23, 74, 11, 65, 58

Dung phương pháp sắp xếp kiểu chèn (insertion sort), sau 3 bước dãy có dạng nào ?

C. 11,23,42,74,65,58

Cho dãy khóa 42,23,74,11,65,58,94,36

Dùng phương pháp sắp xếp kiểu nổi bọt (buble sort) sắp xếp dãy trên theo thứ tự tăng dần, sau bước 2, dãy có dạng nào?

C. 11,23,42,36,74,58,65,,94

Cho dãy khóa 42,23,74,11,65,58,94,36

Dùng phương pháp sắp xếp kiểu phân đoạn với 42 làm chốt, sau bước 1, chốt 42 nằm ở vị trí thứ mấy trong mảng (mảng được đánh số từ 1,2,....,8)

C. thứ 4

Trong các giải thuật sắp xếp, giải thuật nào áp dụng phương pháp “Chia để trị”

B. Quick sort, Merge sort

Cho dãy số {6 1 3 0 5 7 9 2 8 4}. Áp dụng phương pháp sắp xếp lựa chọn (Select sort) sau lần lập đầu tiên của giải thuật ta có kết quả: {0 1 3 6

* 7 9 2 8 4}. Dãy số thu được sau lần lập thứ hai là gì ? B.{0136579284}

Cho dãy số {6 1 3 0 5 7 9 2 8 4}. Áp dụng phương pháp sắp xếp lựa chọn (Select sort) sau lần lập đầu tiên của giải thuật ta có kết quả: {0 1 3 6

* 7 9 2 8 4}. Dãy số thu được sau lần lập thứ ba là gì ? D.{0126579384}

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cho dãy số: 12 | | | 2 | | 8 | 5 | 1 | 6 | 4 15 và các bước sắp xếp sau: | | | | | |
| Bước | 1: | 1 |  | 2 | | 8 |  | 5 | 12 | 6 | 4 | 15 |  |  |
| Bước | 2: | 1 |  | 2 | | 8 |  | 5 | 12 | 6 | 4 | 15 |  |  |
| Bước | 3: | 1 |  | 2 | | 4 |  | 5 | 12 | 6 | 8 | 15 |  |  |
| Bước | 4: | 1 |  | 2 | | 4 |  | 5 | 12 | 6 | 8 | 15 |  |  |
| Bước | 5: | 1 |  | 2 | | 4 |  | 5 | 6 | 12 | 8 | 15 |  |  |
| Bước 6: 1 | 2 | 4 | | 5 | 6 | 8 |  | 12 | 15 |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D. Select sort | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | |  |  |  |  | |  |  |  | | | | |
|  | | |  |  |  |  | |  |  |  | | | | |
| Cho dãy số: 4 | | | 7 | 0 | 9 | 2 | | 5 | 3 1 8 | 6 và các bước sắp xếp sau: | | | | |
| Bước | 1: | 0 |  | 4 | | 7 |  | 1 | 9 | 2 | 5 | 3 | 6 | 8 |
| Bước | 2: | 0 |  | 1 | | 4 |  | 7 | 2 | 9 | 4 | 5 | 6 | 8 |
| Bước | 3: | 0 |  | 1 | | 2 |  | 4 | 7 | 3 | 9 | 5 | 6 | 8 |
| Bước | 4: | 0 |  | 1 | | 2 |  | 3 | 4 | 7 | 5 | 9 | 6 | 8 |
| Bước | 5: | 0 |  | 1 | | 2 |  | 3 | 4 | 5 | 7 | 6 | 9 | 8 |
| Bước 6: 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |  |  |  |  |  |  |

A. Bubble sort

Các bước trên dựa theo giải thuật sắp xếp nào?

C.Insert sort

Cho dãy số “ 3 1 6 0 5 4 8 2 9 7” và các bước sắp xếp sau:

Bước 1:(0 1 2) 3 (5 4 8 6 9 7)

Bước 2:0(1 2) 3 ( 5 4 8 6 9 7)

Bước 3:0 1 (2) 3 (5 4 8 6 9 7)

Bước 4:0 1 2 3 (5 4 8 6 9 7)

Bước 5:0 1 2 3 (4) 5 (8 6 9 7 )

Bước 6:0 1 2 3 4 5 (8 6 9 7)

Bước 7:0 1 2 3 4 5 ( 7 6) 8 (9)

Bước 8:0 1 2 3 4 5 (6) 7 8 (9)

Bước 9:0 1 2 3 4 5 6 7 8 (9)

Bước 10:0 1 2 3 4 5 6 7 8 9

Các bước trên dựa theo giải thuật sắp xếp nào?

B. Quick sort

Cho dãy số “ 3 1 6 0 5 4 8 2 9 7” và các bước sắp xếp sau:

Bước 1: 1 3 6 0 5 4 8 2 9 7

Bước 2: 1 3 6 0 5 4 8 2 9 7

Bước 3: 1 3 6 0 5 4 8 2 9 7

Bước 4: 0 1 3 5 6 4 8 2 9 7

Bước 5: 0 1 3 5 6 4 8 2 9 7

Bước 6: 0 1 3 5 6 2 4 8 9 7

Bước 7: 0 1 3 5 6 2 4 8 7 9

Bước 8: 0 1 3 5 6 2 4 7 8 9

Bước 9: 0 1 2 3 4 5 6 7 8 9

Các bước trên dựa theo giải thuật sắp xếp nào?

C. Merge sort

Cho dãy số “ 3 1 6 0 5 4 8 2 9 7” và các bước sắp xếp sau:

Bước 1: 1 3 0 6 4 5 2 8 7 9

Bước 2: 0 1 3 6 2 4 5 8 7 9

Bước 3: 0 1 2 3 4 5 6 8 7 9

Bước 4: 0 1 2 3 4 5 6 7 8 9

Các bước trên dựa theo giải thuật sắp xếp nào?

A. Merge sort hai đường trực tiếp

Cho dãy khóa 42, 23,74, 11, 65, 58, 94, 36. Lần lượt đưa dãy khóa trên vào cây nhị phân tìm kiếm . Những khóa nào được đưa vào cây con phải?

A.74,65,58,94

Cho dãy khóa 42,23,74,11,65,58,94,36. Lần lượt đưa dãy khóa trên vào cây nhị phân tìm kiếm. Bây giờ ta muốn tìm kiếm xem trong dãy khóa trên có khóa 105 không thì phải làm bao nhiêu phép so sánh?

A.3

Cho dãy khóa đã sắp thứ tự

K(k1,k2,k3,…,k11)

Nếu ta muốn tìm kiếm trong K xem có khóa X không và dùng phương pháp nhị phân thì đầu tiên ta so sánh X với khóa nào?

C.6

Phương pháp tìm kiếm một phần tử cho trước trong một danh sách bằng cách duyệt lần lượt từng phần tử của danh sách đó cho đến lúc tìm thấy giá trị mong muốn hay đã duyệt qua toàn bộ danh sách. Đây là phương pháp của thuật toán tìm kiếm nào?

A. Tuần tự

\*Bước 1: Khởi gán i=0;

\*Bước 2: So sánh a[i] với giá trị x cần tìm, có 2 khả năng

a[i]= =x tìm thấy x, Dừng;

a[i]!=x sang bước 3;

\*Bước 3:i=i+1 xét tiếp phần tử kế tiếp trong mảng Nếu i==N: hết mảng.Dừng; Ngược lại: lặp lại bước 2;

Đây là các bước thuật toán tìm kiếm nào?

A. Tuần tự

Độ phức tạp của thuật toán tìm kiếm tuần tự trong trường hợp xấu nhất là bằng bao nhiêu?

C.O(n)

Độ phức tạp của thuật toán tìm kiếm nhị phân trong trường hợp tốt nhất là bằng bao nhiêu?

A.1

Độ phức tạp của thuật toán tìm kiếm nhị phân trong trường hợp xấu nhất là bằng bao nhiêu?

B.O(log$ )

Cho cây nhị phân tìm kiếm gồm các khoa tuần tự như sau: 50,25,63,11,32,59,78.Để tìm phần tử có giá trị=50, cần bao nhiêu phép so sánh?

A.1

Giải thuật tìm kiếm tuần tự được mô tả như sau:

Int sequential\_search(int\*a,int x, int n)

[ int i;

for (i=0;i<n;i++)

if (a[i] == x) return(i);

return (-1);

]

Cho dãy số: -2,6,1,5,10 áp dụng giải thuật trên tìm x có giá trị = -5, giải thuật cho kết quả bằng bao nhiêu?

A.-1

Trong thuật toán tìm kiếm nhị phân, sau mỗi bước, số bản ghi thực hiện sẽ như thế nào?

C. Giảm đi 1 nửa

Cho thuật toán tìm kiếm trên cây nhị phân như sau:

treenode tree\_search(int x, treenode root)

{ treenode temp=root; while ( temp!=NULL)

{ if( x<temp.item) temp=temp.left; Else

If (x>temp.item) temp=temp.right;

Else break;

}

Return temp;

}

Cho cây nhị phân tìm kiếm gồm các khóa tuần tự như sau: 25,19,36,5,22,27,40,-9. Áp dụng thuật toán tìm kiếm trên cây nhị phân (ở trên), tìm vị trí của phần tử có giá trị bằng 20 trên dây, kết quả nhận được là bao nhiêu?

B.-1

Thuật toán vị trí phần tử có giá trị bằng x được mô tả như sau:

Int search(int\*a,int x, int n)

{ int i=1;

While ((a[i]!=x) && (i<=n)) i++;

If(i<=n);

Return(i);

Else Return (-1);

}

Cho dãy số -3,5,8,-2,8,8,6. Áp dụng thuật toán trên tìm vị trí phần tử x=8?

D.3

Đoạn chương trình sau dựa vào thuật toán tìm kiếm tuần tự mô tả thuật toán đếm xem có bao nhiêu phần tử có giá trị bằng 5 trong mảng:

Int search(int\*a,int n)

{ int d=0,i;

…………..(1)

…………..(2) d++;

return (d);

}

Hãy chọn các câu lệnh đúng cho vị trí (1) và (2)? A. (1) for(i=0;i<n;i++)

* 1. if (a[i]==5)

Cho dãy số:48,30,75,28,37,53,84.Lần lượt đưa dãy khóa trên vào cây nhị phân tìm kiếm. Nếu ta chèn thêm phần tử có giá trị=13 vào cây nhị phân này thì phần tử đưa vào nằm ở vị trí nào trong cây?

B. Cây con trái

Khi tìm kiếm trên cây nhị phân để tìm phần tử x, khóa nào trên cây được so sánh với x trước tiên?

C. Gốc

Khi áp dụng thuật toán tìm kiếm trên cây nhị phân, nhưng xảy ra trường hợp cây không có gốc(cây rỗng), khi đó phép tìm kiếm sẽ cho kết quả là bao nhiêu?

D. Phép tìm kiếm không thỏa mãn

Tư tưởng của giải thuật tìm kiếm nhị phân là gì?

B. Tại mỗi bước tiến hành so sánh X với phần tử ở giữa của dãy. Dựa vào bước so sánh này quyết định giới hạn dãy tìm kiếm nằm ở nửa trên, hay nửa dưới của dãy hiện hành

Tư tưởng của giải thuật tìm kiếm tuần tự là gì?

B. So sánh X lần lượt với các thành phần tử thứ nhất, thứ hai,… của dãy cho đến khi gặp phần tử có khóa cần tìm

Tư tưởng của giải thuật tìm kiếm trên cây nhị phân tìm kiếm là gì?

C. Tìm kiếm dựa vào cây nhị tìm kiếm: Nếu giá trị cần tìm nhỏ hơn gốc thì thực hiện tìm kiếm trên cây con trái, ngược lại ta việc tìm kiếm được thực hiện trên cây con phải

Cây nhị phân tìm kiếm là cây có đặc điểm gì?

A. Cây nhị phân mà mỗi nút trong cây đều thỏa tính chất, giá trị của nút cha nhỏ hơn mọi nút trên cây con trái và lớn hơn mọi nút trên cây con phải của nó

Đô phức tạp của thuật toán tìm kiếm tuần tự trong trường hợp tốt nhất là bằng bao nhiêu?

A.1

Cho dãy khóa 42,23,74,11,65,58,94,36. Lần lượt đưa dãy khóa trên vào cây nhị phân tìm kiếm. Những khóa nào được đưa vào cây con trái?

B. 23,11,36