



Trắc nghiệm thi giữa kỳ

Cấu trúc dữ liệu và giải thuật (Trường Đại Học Nông Lâm Thành Phố Hồ Chí Minh)



Scan to open on Studeersnel

CTDL_HK3_TinChi

21130408@st.hcmuaf.edu.vn [Chuyển đổi tài khoản](#)



Đã lưu bản nháp

Tên, email và ảnh liên kết với Tài khoản Google của bạn sẽ được ghi lại khi bạn tải tệp lên và gửi biểu mẫu này

Mọi tệp đã tải lên sẽ được chia sẻ bên ngoài tổ chức sở hữu các tệp đó.

*** Biểu thị câu hỏi bắt buộc**

Nội dung bài làm

Để tìm kiếm 1 phần tử trong mảng, nếu sử dụng Linear search (tìm kiếm tuyến tính) thì trường hợp tệ nhất độ phức tạp của giải thuật là? *

- ☐ $O(n/2)$
- ☒ $O(n)$
- ☐ $O(n \log(n))$
- ☐ $O(1)$

Phát biểu nào sau đây không đúng về java.util.ListIterator: *

- ☐ ListIterator dùng để duyệt tuần tự các phần tử của cấu trúc dữ liệu ArrayList, LinkedHashSet, TreeSet
- ☒ ListIterator là một interface
- ☐ ListIterator có chứa các hàm hasNext(), next(), add(), remove(), previous()
- ☐ ListIterator chỉ cho phép duyệt các phần tử theo hướng forward

Kết quả thực hiện đoạn chương trình dưới đây là gì? *

```
public static void f1() {  
    List<Integer> list = Arrays.asList(1, 2, 3);  
    ListIterator<Integer> iter = list.listIterator();  
    while(iter.hasNext()) {  
        int next = iter.next();  
        if(!iter.hasNext()) {  
            iter.add(next+1);  
        }  
    }  
    System.out.println(list);  
}
```

- ☐ [1, 2, 3]
- ☒ [1, 2, 3, 4]
- ☐ Ngoại lệ
- ☐ Lỗi biên dịch

Khai báo mảng 2 chiều nào dưới đây không bị lỗi biên dịch? *

- ☐ String [3] array[3] = new String[3][3];
- ☐ String array[5][];
- ☐ String [] array [] = [3][3];
- ☒ Object[][] array = new String[3][3];

**Đối với tổng đài chăm sóc khách hàng của một doanh nghiệp thì cấu trúc dữ liệu *
nào sau đây phù hợp nhất để quản lý các cuộc gọi đến của khách hàng**

- ☒ HashSet
- ☐ LinkedList
- ☐ ArrayList
- ☐ Stack

**Đối với mảng [7,7,7,7,7,4,4,4] , hãy so sánh tốc độ thực thi của của giải thuật *
Quick Sort chọn pivot là phần tử đầu và giải thuật Merge sort**

- ☐ Quick sort nhanh hơn Merge sort
- ☐ Quick sort tìm ra kết quả, Merge sort không tìm ra kết quả.
- ☐ Quick sort và Merge sort như nhau.
- ☒ Quick sort chậm hơn Merge sort.

**Làm cách nào để chuyển một mảng thành một iterator (VD: int[] a = {1,2,3,4,5,6}; *
)**

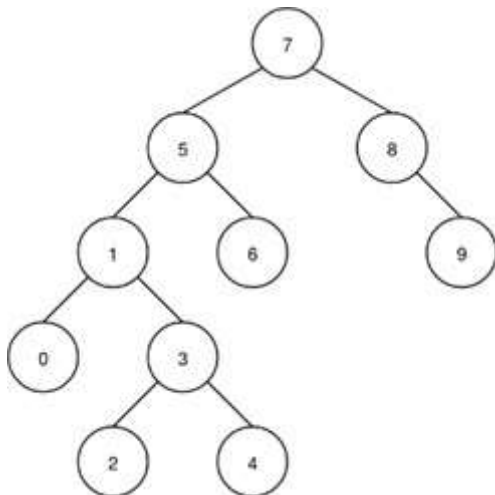
- ☐ Array.asList(a).iterator();
- ☐ a.tolterator();
- ☒ Arrays.asList(a).iterator();
- ☐ a.iterator();

Kết quả sau khi thực thi đoạn chương trình dưới đây là gì? *

```
public class Point {  
    private int x;  
    private int y;  
    public Point(int x, int y) {  
        this.x = x; this.y = y;  
    }  
    public static void main(String[] args) {  
        Set<Point> pos = new HashSet<Point>();  
        pos.add(new Point(1, 2)); pos.add(new Point(3, 4));  
        pos.add(new Point(3, 4));  
  
        System.out.println(pos.size());  
    }  
}
```

- ☒ 3
- ☐ Lỗi biên dịch
- ☐ 2
- ☐ Ngoại lệ

Chiều cao của cây nhị phân sau đây là bao nhiêu? *



- ☐ 2
- ☒ 4
- ☐ 3
- ☐ 10

Cho mảng của các số nguyên như sau: {29, 10, 14, 37, 13}. Ở lần lặp thứ 2 (vòng lặp ngoài) của thuật toán Insertion sort thì thứ tự các phần tử trong mảng là gì? *

- ☐ Lựa chọn khác
- ☐ {10, 29, 14, 37, 13}
- ☒ {10, 13, 14, 37, 29}
- ☐ {10, 14, 29, 37, 13}

Kết quả thực hiện đoạn chương trình sau đây là gì? *

```
HashMap<Integer, String> hash = new HashMap<>();  
    hash.put(2, "Hai");  
    hash.put(3, "Ba");  
    hash.put(1, "Mot");  
    hash.put(1, "One");  
    System.out.println(hash);
```

- ☐ {1=One, 2=Hai, 3=Ba}
- ☐ Thứ tự ngẫu nhiên của {1=Mot, 2=Hai, 3=Ba}
- ☒ Báo lỗi
- ☐ {1=Mot, 2=Hai, 3=Ba}

Kết quả thực hiện đoạn code dưới đây với $n=7$ sẽ in ra màn hình nội dung gì? *

```
public static void inc(int n) {  
    if (n == 0)  
        return;  
    else {  
        inc(n/2);  
        System.out.print(n + " ");  
    }  
}
```

- ☒ 7 4 2 1
- ☐ 1 2 4 7
- ☐ 1 3 7
- ☐ 7 3 1

Hãy nhận diện phát biểu sai trong các phát biểu sau? *

- ☒ Mảng là cấu trúc tuần tự, có kích thước không cố định.
- ☐ Mảng có nhiều loại mảng một chiều, hai chiều, đa chiều.
- ☐ Mảng có tính chỉ mục (index).
- ☐ Mảng cho phép chứa các phần tử có cùng kiểu dữ liệu.

Kết quả của danh sách liên kết sau là gì? *

```
LinkedList<Integer> list = new LinkedList<>();  
list.add(9);  
list.addFirst(10);  
list.add(0,7);  
list.add(2,5);  
list.addLast(11);  
list.remove();
```

```
System.out.println(list);
```

- ☐ [10, 5, 9, 11]
- ☐ [7, 9, 5, 11]
- ☐ [10, 7, 9, 5]
- ☒ [9,0,7,2,5,11]

Cho mảng của các số nguyên như sau: {29, 10, 14, 37, 13}. Ở lần lặp thứ 2 (vòng lặp ngoài) của thuật toán Insertion sort thì thứ tự các phần tử trong mảng là gì? *

- ☐ {10, 14, 29, 37, 13}
- ☐ Lựa chọn khác
- ☒ {10, 13, 14, 37, 29}
- ☐ {10, 29, 14, 37, 13}

Kết quả thực hiện đoạn chương trình sau đây là gì? *

```
Stack<String> st = new Stack<>();  
    st.push("Apple");  
    st.push("Pen");  
    st.peek();  
    st.push("Apple Pen");  
    st.pop();  
    st.push("Pine");  
    st.push("Pine Apple Pen");  
    st.peek();  
    System.out.println(st);
```

- ☐ [Apple, Pen]
- ☐ [Pine Apple Pen, Pine, Pen, Apple]
- ☒ [Apple, Pen, Pine, Pine Apple Pen]
- ☐ [Pine, Pine Apple Pen]

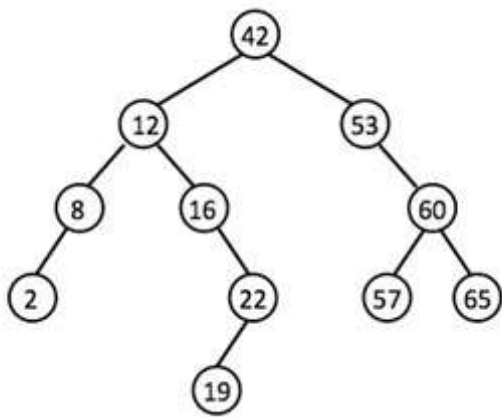
Cho mảng[1,6,3,4,5,2,7,8]so sánh tốc độ chạy của giải thuật linear search và giải thuật binary search khi tìm kiếm số 6 *

- ☐ Linear search tìm ra kết quả và binary search không tìm ra kết quả.
- ☐ Cả hai tìm ra kết quả, Linear search và binary search nhanh như nhau.
- ☒ Cả hai tìm ra kết quả, Binary search tìm kiếm nhanh hơn linear search.
- ☐ Cả hai tìm ra kết quả, Linear search tìm kiếm nhanh hơn binary search.

Sắp xếp tốc độ chạy của các giải thuật Selection Sort, Merge Sort Bubble Sort theo thứ tự tăng dần (tốc độ chậm nhất đến tốc độ nhanh nhất). *

- ☐ Merge Sort, Bubble Sort, Selection Sort.
- ☐ Selection Sort, Merge Sort, Bubble Sort .
- ☐ Selection Sort, Bubble Sort, Merge Sort.
- ☒ Bubble Sort, Merge Sort, Selection Sort.

Kết quả duyệt cây nhị phân sau đây theo preorder là gì? *



- ☐ 2 8 12 16 19 22 42 53 57 60 65
- ☐ 42 12 8 2 16 22 19 53 60 57 65
- ☐ Lựa chọn khác
- ☒ 2 8 19 22 16 12 57 65 60 53 42

Cây nhị phân có level=4 thì cây loại nào sau đây có số node nhiều nhất? *

- ☒ Vừa là cây nhị phân đầy đủ, vừa là cây nhị phân hoàn chỉnh
- ☐ Cây nhị phân hoàn chỉnh (complete binary tree)
- ☐ Không phải là cây nhị phân đầy đủ, cũng không phải là cây nhị phân hoàn chỉnh
- ☐ Cây nhị phân đầy đủ (full binary tree)

Quay lại

Tiếp

Xóa hết câu trả lời

Không bao giờ gửi mật khẩu thông qua Google Biểu mẫu.

Biểu mẫu này đã được tạo ra bên trong Nong Lam University. [Báo cáo Lạm dụng](#)

Google Biểu mẫu

