

CSE 12 Final Review Discussion

...

Max Jiao & Pooja Bhat

Review Quiz

What is the output of the following code?

```
#include <stdio.h>

void funcOne(int n)
{
    n = 120;
}

void funcTwo(int *n)
{
    *n = 240;
}

int main()
{
    int n = 250;
    funcOne(n);
    funcTwo(&n);
    printf("n= %d", n);
    return 0;
}
```

- A. n= 120
- B. n= 240
- C. n= 250
- D. None of these
- E. Seg fault

Review Quiz

What is the output of the following code?

```
#include <stdio.h>

void funcOne(int n)
{
    n = 120;
}

void funcTwo(int *n)
{
    *n = 240;
}

int main()
{
    int n = 250;
    funcOne(n);
    funcTwo(&n);
    printf("n= %d", n);
    return 0;
}
```

- A. n= 120
- B. n= 240
- C. n= 250
- D. None of these
- E. Seg fault

Review Quiz

What is the output of the following code?

```
#include <stdio.h>
#include <stdlib.h>

int main () {
char * str1 = (char*)malloc(sizeof(char) *
100);

char str2[ ] = str1;
char str3[ ] = "gogogo";

printf("%s\n", str2);

return 0;
}
```

- A. Garbage values
- B. gogogo
- C. Address of str1
- D. Compile error

Review Quiz

What is the output of the following code?

```
#include <stdio.h>
#include <stdlib.h>

int main () {
char * str1 = (char*)malloc(sizeof(char) *
100);

char str2[ ] = str1;
char str3[ ] = "gogogo";

printf("%s\n", str2);

return 0;
}
```

- A. Garbage values
- B. gogogo
- C. Address of str1
- D. Compile error

Review Quiz

What is the output of the following code?

```
int main () {  
    char buffer[10];  
  
    strcpy(buffer, "CSE12");  
    buffer[2] = (int)'3';  
  
    printf("%s\n", buffer);  
  
    return 0;  
}
```

- A. CS312
- B. CS?12 (? is some non-letter or unknown character)
- C. Error because putting int in char array
- D. CSE12
- E. None of these

Review Quiz

What is the output of the following code?

```
int main () {  
    char buffer[10];  
  
    strcpy(buffer, "CSE12");  
    buffer[2] = (int)'3';  
  
    printf("%s\n", buffer);  
  
    return 0;  
}
```

- A. CS312
- B. CS?12 (? is some non-letter or unknown character)
- C. Error because putting int in char array
- D. CSE12
- E. None of these

Review Quiz

What goes in the call to foo to print out 10?

```
#include<stdio.h>
#include <stdlib.h>
#include "main.h"

void foo(int * x) {
    printf("%d", *x);
}
```

```
int main()
{
    int *p = (int
    *)malloc(sizeof(int));
    int x = 10;
    *p = x;

    foo(____); //what goes here?
}
```

A. &p

B. p

C. *p

Review Quiz

What goes in the call to foo to print out 10?

```
#include<stdio.h>
#include <stdlib.h>
#include "main.h"

void foo(int * x) {
    printf("%d", *x);
}
```

```
int main()
{
    int *p = (int
    *)malloc(sizeof(int));
    int x = 10;
    *p = x;

    foo(____); //what goes here?
}
```

A. &p

B. p

C. *p

Review Quiz

How do you de-allocate memory for p?

```
#include<stdio.h>
#include <stdlib.h>
#include "main.h"
```

```
void foo(int * x) {
    printf("%d", *x);
}
```

```
int main()
{
    int *p = (int
    *)malloc(sizeof(int));
    int x = 10;
    *p = x;

    foo(____); //what goes here?
}
```

A. free(p)

B. free(&p)

C. free(*p)

Review Quiz

How do you de-allocate memory for p?

```
#include<stdio.h>
#include <stdlib.h>
#include "main.h"
```

```
void foo(int * x) {
    printf("%d", *x);
}
```

```
int main()
{
    int *p = (int
    *)malloc(sizeof(int));
    int x = 10;
    *p = x;

    foo(____); //what goes here?
}
```

A. free(p)

B. free(&p)

C. free(*p)

Review Quiz

What is the output of the following code?

```
#include <stdio.h>

int foo(int x, int y) {
    return x+y;
}

int main() {
    printf("%d", foo(1,3));

    return 0;
}
```

- A. Seg fault
- B. Memory leak
- C. Some garbage value
- D. 4
- E. Compile error

Review Quiz

What is the output of the following code?

```
#include <stdio.h>

int foo(int x, int y) {
    return x+y;
}

int main() {
    printf("%d", foo(1,3));

    return 0;
}
```

- A. Seg fault
- B. Memory leak
- C. Some garbage value
- D. 4**
- E. Compile error

Tips

1. For the non-coding part, through all the resources you have:
 - HW1 (True/False questions), HW2 (LinkedLists and Iterators), HW3, HW5 (All sorting algorithms), HW8-Part1
 - Zybooks, Midterm/Final Practice Problems, Midterm, Midterm Review Discussion Section, In-class Quiz questions, Reading and review quiz
2. For the coding part, you may refer to all the homeworks, the extra credit problems in homeworks, midterm review discussion and the midterm/final practice problems.
3. Manage time well
4. One cheat sheet allowed - use it well
5. Make no assumptions. Read the question carefully. Your answers must be clear and unambiguous

What is the output of the following code?

```
void main()
{
    int a[]={ 1, 2, 3, 4, 5 }, *p;
    p=a;
    ++*p;
    printf("%d ", *p);
    p += 2;
    printf("%d", *p);
}
```

a) 2, 4

b) 3, 4

c) 2, 2

d) 2, 3

e) 3, 3

What is the output of the following code?

```
void main()
```

```
{
```

```
    int a[]={ 1, 2, 3, 4, 5 }, *p;
```

```
    p=a;
```

```
    ++*p;
```

```
    printf("%d ", *p);
```

```
    p += 2;
```

```
    printf("%d", *p);
```

```
}
```

a) 2, 4

b) 3, 4

c) 2, 2

d) 2, 3

e) 3, 3

What is the output of the following?

```
int main()
{
    char str[20] = "Hello";
    char *p=str;
    *p='M';
    char* q = p;
    q+=2;
    printf("%s\n", p);
    printf("%s\n", q);
    return 0;
}
```

- a) Hello, llo
- b) Mello, l
- c) M, l
- d) H, l
- e) Mello, llo

What is the output of the following?

```
int main()
{
    char str[20] = "Hello";
    char *p=str;
    *p='M';
    char* q = p;
    q+=2;
    printf("%s\n", p);
    printf("%s\n", q);
    return 0;
}
```

- a) Hello, llo
- b) Mello, l
- c) M, l
- d) H, l
- e) Mello, llo

4. Suppose we are sorting an array of eight integers using Quick Sort, and we have just finished the first partitioning with the array looking like this:

2 5 1 7 9 12 11 10

Which statement is correct?

- a. The pivot could be either 7 or 9.
- b. The pivot could be 7, but it is not 9.
- c. The pivot is not 7, but it could be 9.
- d. Neither 7 nor 9 is the pivot.

4. Suppose we are sorting an array of eight integers using Quick Sort, and we have just finished the first partitioning with the array looking like this:

2 5 1 7 9 12 11 10

Which statement is correct?

- a. The pivot could be either 7 or 9.
- b. The pivot could be 7, but it is not 9.
- c. The pivot is not 7, but it could be 9.
- d. Neither 7 nor 9 is the pivot.

Which of the following would be best suited to store data with the following requirement?

A list of employee records with distinct IDs need to be stored in a manner that it is easy to add a new employee record or delete an old employee record.

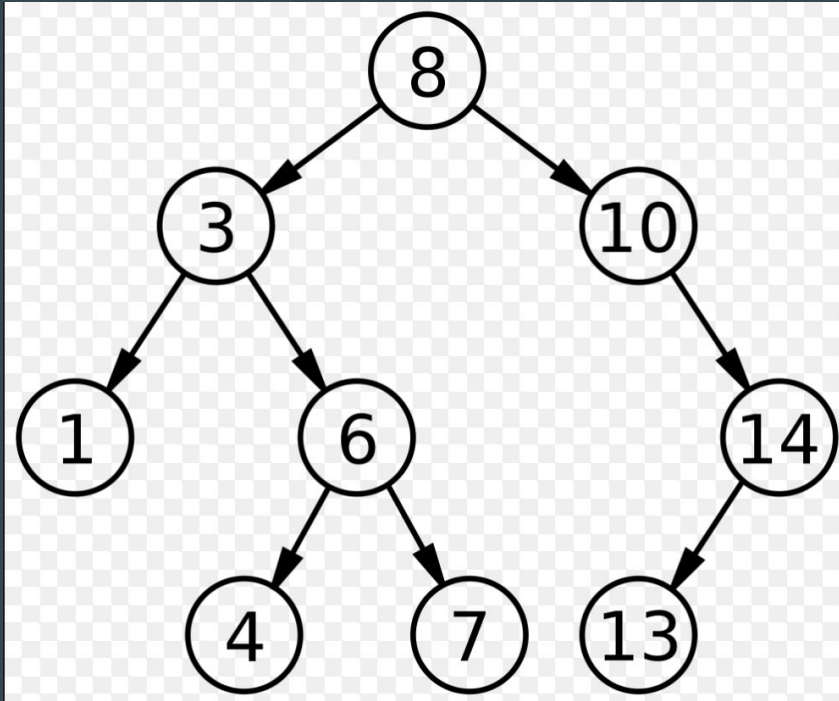
- A. Array
- B. LinkedList
- C. HashTable
- D. Binary Search Tree
- E. Heap

Which of the following would be best suited to store data with the following requirement?

A list of employee records with **distinct IDs** need to be stored in a manner that it is easy to **add** a new employee record or **delete** an old employee record.

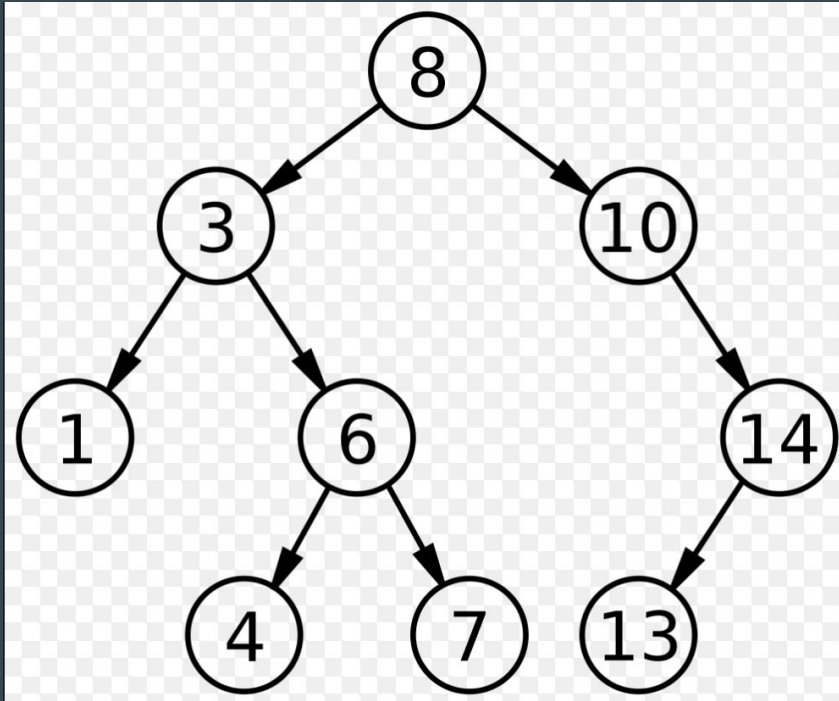
- A. Array
- B. LinkedList
- C. HashTable
- D. Binary Search Tree
- E. Heap

9. Which of the following sequence of insertion cannot result in the given BST?



- a) 8, 3, 10, 14, 6, 1, 4, 13, 7
- b) 8, 3, 1, 6, 10, 4, 14, 7, 13
- c) 8, 10, 3, 14, 13, 6, 7, 4, 1
- d) 8, 10, 3, 14, 1, 4, 6, 7, 13
- e) 8, 3, 10, 1, 6, 14, 13, 7, 4

9. Which of the following sequence of insertion cannot result in the given BST?



- a) 8, 3, 10, 14, 6, 1, 4, 13, 7
- b) 8, 3, 1, 6, 10, 4, 14, 7, 13
- c) 8, 10, 3, 14, 13, 6, 7, 4, 1
- d) 8, 10, 3, 14, 1, 4, 6, 7, 13
- e) 8, 3, 10, 1, 6, 14, 13, 7, 4

HEAPS

15. Which of the following represents a valid sequence of an array representing a 3-ary max heap?

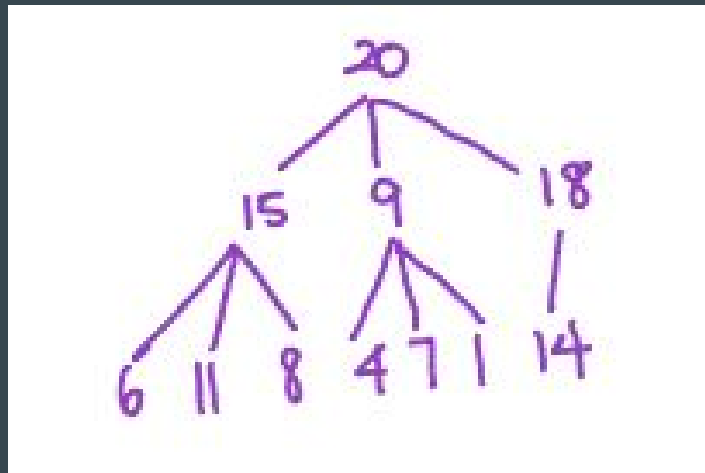
- a) 20, 15, 18, 11, 9, 6, 4, 8, 7, 1, 14
- b) 20, 9, 18, 7, 15, 6, 1, 14, 11, 8, 4
- c) 20, 15, 9, 18, 6, 11, 8, 4, 7, 1, 14
- d) 20, 18, 9, 15, 14, 8, 1, 6, 4, 11, 7

HEAPS

15. Which of the following represents a **valid** sequence of an array representing a **3-ary** max heap?

- a) 20, 15, 18, 11, 9, 6, 4, 8, 7, 1, 14
- b) 20, 9, 18, 7, 15, 6, 1, 14, 11, 8, 4
- c) 20, 15, 9, 18, 6, 11, 8, 4, 7, 1, 14
- d) 20, 18, 9, 15, 14, 8, 1, 6, 4, 11, 7

Remember the heap property



Which of the following would be best suited to store data with the following requirement?

The data set contains many records with duplicate keys. Only requirement is to always store the list in a sorted order and search a key quickly.

- A. LinkedList
- B. Stack
- C. HashTable
- D. Binary Search Tree
- E. Array
- F. Priority Queue

Which of the following would be best suited to store data with the following requirement?

The data set contains many records with duplicate keys. Only requirement is to always store the list in a sorted order and search a key quickly.

- A. LinkedList → can't search quickly
- B. Stack → can't keep it sorted easily
- C. HashTable → can't keep it sorted
- D. Binary Search Tree → Maybe a modified version (careful about sorted def)
- E. Array → have to do a sort every time you insert, but if we don't need to add, then searching is binary search

Which of the following would be best suited to store data with the following requirement?

You want to build a meeting reminder for a PDA that keeps track of events you schedule and periodically checks the next event to sound an alarm to remind you of the next thing you need to do.

- A. LinkedList
- B. Stack
- C. HashTable
- D. Binary Search Tree
- E. Array
- F. Priority Queue

Which of the following would be best suited to store data with the following requirement?

A library needs to maintain books by their ISBN number. The requirement is finding them as soon as possible, adding new books to the system and deleting an existing book entry from the system.

- A. LinkedList
- B. Stack
- C. HashTable
- D. Binary Search Tree
- E. Array

Which of the following would be best suited to store data with the following requirement?

A library needs to maintain books by their ISBN number. The requirement is finding them as soon as possible, adding new books to the system and deleting an existing book entry from the system.

- A. LinkedList
- B. Stack
- C. HashTable
- D. Binary Search Tree
- E. Array

Which of the following would be best suited to store data with the following requirement?

A system needs to maintain Student Registrations for enrolling into a Course. The requirement is to store and return the Registrations in the order they were added to the system.

- A. LinkedList
- B. Stack
- C. HashTable
- D. Binary Search Tree
- E. Queue

Which of the following would be best suited to store data with the following requirement?

A system needs to maintain Student Registrations for enrolling into a Course. The requirement is to store and return the Registrations **in the order they were added to the system.**

- A. LinkedList → similar to queue, but it has more functionality than needed
- B. Stack
- C. HashTable
- D. Binary Search Tree
- E. Queue → does what we need

A hash table of length 10 uses open addressing with hash function $h(k)=k \bmod 10$, and linear probing.

0	1	2	3	4	5	6	7	8	9
		42	23	34	52	46	33		

After inserting 6 values, the resulting table is shown above. Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

- a) 46, 42, 34, 52, 23, 33
- b) 34, 42, 23, 52, 33, 46
- c) 46, 34, 42, 23, 52, 33
- d) 42, 46, 33, 23, 34, 52

A hash table of length 10 uses open addressing with hash function $h(k)=k \bmod 10$, and linear probing.

0	1	2	3	4	5	6	7	8	9
		42	23	34	52	46	33		

After inserting 6 values, the resulting table is shown above. Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

- a) 46, 42, 34, 52, 23, 33
- b) 34, 42, 23, 52, 33, 46
- c) 46, 34, 42, 23, 52, 33
- d) 42, 46, 33, 23, 34, 52

Given a Sorted Doubly Linked List with a head:

1. What is the asymptotic complexity of finding the smallest element in the List?
2. What is the asymptotic complexity of finding the largest element in the list?
3. What is the asymptotic complexity of determining whether a given element e appears in the list?
4. What is the asymptotic complexity of finding the median element in the list?
5. What is the asymptotic complexity of deleting a given element e in the list?

7. Which of the following is true about a binary and d-heap with same number of elements

- A. Insert is faster in a binary heap while remove is faster in a d-ary heap
- B. Insert is faster in a d-ary heap while remove is faster in a binary heap
- C. Both insert and delete are equally efficient in binary and d-heap of the same nelems
- D. Insert is equally efficient while remove is faster in a binary heap.

7. Which of the following is true about a binary and d-heap with same number of elements

- A. Insert is faster in a binary heap while remove is faster in a d-ary heap
- B. Insert is faster in a d-ary heap while remove is faster in a binary heap
- C. Both insert and delete are equally efficient in binary and d-heap of the same nelems
- D. Insert is equally efficient while remove is faster in a binary heap.

Explanation:

1. Inserting a value into a heap only depends on the number of levels of the heap, since bubbleUp does not require comparison between same-level elements. Thus the time it takes to insert is $O(\log_d(n))$. With bigger 'd' results in faster insertion.
2. Faster in binary heap because comparison is only among the two children

Which of the following statements are true about a binary tree of height 3?

- a) It should have a minimum of 3 nodes
 - b) It can have a maximum of 7 nodes.
 - c) It should have a minimum of 4 nodes.
 - d) It can have a maximum of 15 nodes
 - e) It can have at most 8 leaf nodes.
 - f) It can have at most 4 leaf nodes
- A. a) b)
 - B. c) d)
 - C. a) b) e)
 - D. c) d) e)
 - E. a) b) f)

Which of the following statements are true about a binary tree of height 3?

- a) It should have a minimum of 3 nodes
 - b) It can have a maximum of 7 nodes.
 - c) It should have a minimum of 4 nodes.
 - d) It can have a maximum of 15 nodes
 - e) It can have at most 8 leaf nodes.
 - f) It can have at most 4 leaf nodes
- A. a) b)
 - B. c) d)
 - C. a) b) e)
 - D. c) d) e)
 - E. a) b) f)

Remember that a binary heap has max 2 children
Height 0 vs height 1!

Suppose that items A, B, C, D and E are pushed, in that order, onto an initially empty stack S. S is then popped four times; as each item is popped off, it is inserted into an initially empty queue. If two items are then removed from the queue, what is the next item that will be removed from the queue?

1. Item A
2. Item B
3. Item C
4. Item D
5. Item E

What is the output?

```
void main()  
{  
    int i=10; /* assume address of i is 1000 */  
    int *ip=&i;  
    int **ipp=&&i;  
    printf("%d, %d, %d", &i, ip, *ipp);  
}
```

- a) 1000, 10, 10
- b) 1000, 1000, 1000
- c) 1000, 1000, 10
- d) 1000, 10, 10
- e) Syntax error

What is the output?

```
void main()
{
    int i=10; /* assume address of i is 1000 */
    int *ip=&i;
    int **ipp=&&i;
    printf("%d, %d, %d", &i, ip, *ipp);
}
```

- a) 1000, 10, 10
- b) 1000, 1000, 1000
- c) 1000, 1000, 10
- d) 1000, 10, 10
- e) Syntax error

&&i → address of address of i?

&& is logical AND operator in C

Consider a set of numbers 40, 5, 18, 29, 35, 7 and a hash table of size 11. Insert the numbers into the tables using:

- a) Separate chaining
- b) Linear Probing
- c) Quadratic Probing

TRUE or FALSE

11. In a binary search tree, a node can be inserted only as a leaf node

Answer: True

12. A queue can be implemented efficiently using a linear array.

Answer: False

13. In a max-heap the smallest element resides always at the leaves assuming all elements are distinct.

Answer: True

TRUE OR FALSE

1. If class A implements interface I, class B extends A, and class C extends B, then C implements I.
2. If class A implements interface I, class B extends A, class C extends B, and interface J extends interface I, then C implements J.
3. 3^n is $O(2^n)$
4. The finally block in a try-catch-finally statement is executed even if no exception is thrown and there is a return statement in the try block.
5. A stack is a FIFO structure and a queue is a LIFO structure.
6. A collision occurs when two identical keys result in two identical hash values

What is a hash function?

What are the properties of a good hash function? What is the load factor of a hash table?

What is the problem with Linear Probing and Quadratic Probing respectively and how does Double Hashing solve this issue?

17. Write a method to recursively count the number of nodes in a BST. Assume that you have access to the getters `getLChild()` and `getRChild()`

```
public int countNodes(BSTNode root) {  
  
}
```


Answer:

```
public int countNodes(BSTNode root) {  
    if (root == NULL)  
        return 0;  
    return (1 + countNodes(root.getLChild()) + countNodes(root.getRChild()))  
}
```

For each of the following storage and operations, give the expected time required for the operation. Your answers should assume that the data structure is performing well, i.e., binary search trees are reasonably well balanced and hash tables have a low load factor (i.e., your answer should be the expected time, not the worst case time if those two are different).

STORAGE	CONTAINS : determine whether a particular object is a member of a collection.	INSERT : add a new item to the collection	DELETE: remove a given item from the collection
SORTED ARRAY			
SORTED SINGLY LINKED LIST WITH HEAD AND TAIL			
HASH TABLE			
BINARY SEARCH TREE			

- 1) Draw the BST that results from inserting 45, 78, 25, 19, 38, 99, 72, 44, 85, 5 in that order
- 2) What is the height of the tree? What about leaf count?
- 3) Draw the tree after deleting the value 25

Remember: **Depth** of a node is the number of edges between the node and the root.
Height of a tree is the maximum depth of any node in the tree. By convention, the root sits at height 0.

- Draw the 2-max-heap that results from inserting 12, 7, 25, 19, 8, 15, 36, 45 in that order.
- What is the inorder, preorder and postorder sequence of the above heap structure?
- Draw the tree that results after a remove operation on the heap. Show the intermediate steps.

Suppose that we have numbers between 1 and 100 in a binary search tree and want to search for the number 55. Which of the following sequences CANNOT be the sequence of nodes examined?

- a) {10, 75, 64, 43, 60, 57, 55}
- b) {90, 12, 68, 34, 62, 45, 55}
- c) {9, 85, 47, 68, 43, 57, 55}
- d) {79, 14, 72, 56, 16, 53, 55}

Answer: c)

A Collection of Coding Questions

- Write a RECURSIVE method to reverse a Singly Linked List with a head.
- Write a method to count the number of nodes in a BST with Integer values, whose value is greater than k
 - a. `public int countNodes(BSTree tree, int k)`
- Given a list, print all the pairs whose difference is k. Your solution must be as time-efficient as possible. (What data structure do you think is useful here?)
 - a. Example: List is { 1, 7, 5, 9, 2, 12, 3} and k =2
 - b. Output Pairs: (1,3), (3,5), (5,7), (7,9)
- Given two DoublyLinkedLists with head, write a method that returns an intersection of the lists: `public intersection(DoublyLinkedList1, DoublyLinkedList2)`

- Write a method to determine whether the contents of a Singly Linked List with a head forms a palindrome. Assume that each node contains a char data. Your solution must be time-efficient and you may use any additional data structures you wish
 - `public isPalindrome(SinglyLinkedList myList)`
- Given two strings A and B, find if one is a permutation of the other.
 - abb and bba - true
 - abcd and acdb - true
 - acc and ccb - false
- You are given a list of $n-1$ integers and these integers are in the range of 1 to n . There are no duplicates in list. One of the integers is missing in the list. Find the missing number.
 - Hint: Do you even need a hash table?

- Find if a string contains all unique characters. Your solution must be as efficient as possible.
 - What if you cannot use additional $O(n)$ space?
- Write a method to find the element in a BST that is one larger than the root.
`public E nextLargestFromRoot(BSTree tree)`
- Write a method to print the levelOrderTraversal of a Binary Tree.
 - `public void levelOrderTraversal(BSTree tree)`
- Write a method to find the sum of all leaf nodes in a BST with Integer data.
- Using the iterators of two BSTs, find the union of the two BSTs
 - `public ArrayList<T> findUnion(Iterator it1, Iterator it2)`
- Count the number of nodes in a BST with one child
- Write a method to determine if a BST is a complete tree

Compiles or not?

1. `LinkedList<List> myL = new LinkedList<List>();`

//Good. same type

2. `Collection<List> c = new LinkedList<List>();`

//Good because LinkedList implements Collection

3. `LinkedList<List> myL;`

`myL = new LinkedList<ArrayList>();`

//Error! Type mismatch: cannot convert from LinkedList<ArrayList> to LinkedList<List>

4. `LinkedList<? extends List> myL = new LinkedList<ArrayList>();`

//Good: Holds any subtype of list

5. `LinkedList<? super List> myL = new LinkedList<List>();`
//Good. List or any superclass of List.

6. `LinkedList<? super List> myL = new LinkedList<Collection>();`
Good. List extends Collection.

7. `LinkedList<Collection> myL = new LinkedList<Collection>();`
Good.

8. `myL.add(new ArrayList());`
Also good