

Bir soru da benden size hediye:

bir min heap i max heap e dönüştüren bir fonksiyon yazın

```
queue *convert(queue *q){
    queue *maxheap;
    initialize(&maxheap);
    while (q->counter!=0){
        int data=delete(&q);
        insert2(&maxheap,data);
    }
    return maxheap;
}.
```

BU kadar işte sadece insert2 fonksiyonunu bizim yazdığımız insert te eşitsizliğin yönü değiştirerek yazacaksınız. Ya da o fonksiyonu while içine gömebilirsiniz. Ama modülerlik her zaman iyidir. delete fonksiyonu zaten sınıftakinin aynısı.

Şimdi mantık şu. delete ile elimizdeki heaptan siliyoruz. tabiiki ilk eleman siliniyor yani minimum olan. Bunu da yeni kuyruğa yani heap e ekliyoruz. insert2 ise max-heap özelliği ile ekleyecek. yani parent cocuklardan büyük olacak. Şimdi süper oldu hocam :)

"C:\Users\Halil Dilaver\Desktop\binary-min-heap\bin\Debug\binary-min-heap.exe"

Kuyruk(Min)= 10 20 25 30 35 40 60 90 85
Cevrilmis Kuyruk(Max)= 90 85 40 60 25 20 35 10 30

```

void insert2(queue *q,int key){
    q->counter++;

    if(q->counter<=QUEUE_SIZE){
        int index=q->counter;
        q->A[index]=key;
        while(index!=1 && q->A[index/2] < q->A[index])
        {
            int temp=q->A[index];
            q->A[index]=q->A[index/2];
            q->A[index/2]=temp;
            index=index/2;
        }
    }

    queue *convert(queue *q){
        queue *maxheap;
        initialize(maxheap);
        while (q->counter!=0){
            int data=delete(&q);
            insert2(maxheap,data);
        }
        return maxheap;
    }
}

```

Soru:(Klasik) Bir ikili arama ağacındaki verilerden tek olanları diğer bir BST ağacına kopyalayan copyOdd isimli fonksiyonu yazınız.

Yanıt:

```

struct node *copyOdd(struct node *root, struct node *root2){
    if(root != NULL){
        if(root -> data % 2 == 1)
            root2 = insert(root2, root -> data);
        root2 = copyOdd(root -> left, root2);
        root2 = copyOdd(root -> right, root2);
    }
    return root2;
}

```

Soru:(Test)Avl ağaçlarında neden döndürme işlemi yaparız?

Yanıt:

İterasyon sayısını azaltmak ve Performansı artırmak için döndürme yaparız. Bu aslında sonuçtur. Ağacı dengeli hale getirmek için döndürme/döndürmeler yaparız. Dengeli bir ağaçta search/min/max/add/delete/successor/predecessor gibi işlemler daha hızlı gerçekleşir. (HK)

Soru:(Test)N düğüme sahip bir heap'in yüksekliği nedir?

Yanıt:Ign

Soru:(klasik)Tanımlanan bir ikili arama ağacında sadece sol çocuğu olan düğümleri ekrana yazdıran fonksiyonu yazınız?

Yanıt:

```
void leftchild(BTREE *root)
```

```
{
    if(root != NULL)
    {
        if(root->left != NULL && root->right == NULL)
            printf("%d",root->data);
        leftchild(root->left);
        leftchild(root->right);
    }
    else
        return;
}
```

1

Soru:(Test)Binary heap te minimum eleman nerede bulunur?

Yanıt:

Root

Soru:(Test)Neden Binary heap kullanırız?

Yanıt:

Amaç öncelikli kuyrukları gerçekleştirmek için kullanılır.

Soru:(Test)Aşağıdakilerden hangisi stack fonksiyonlarından biri değildir?

Yanıt:D(data)

a) Push b) Reset c)Top d)Data e)Pop

Soru:Aşağıdaki açıklamalardan hangisi yanlıştır ?

Yanıt:E

a) Düğüm (node) : Ağacın her bir elemanına düğüm adı verilir.

b)Kök (root) : En üst seviyedeki tek düğümdür.

c)Çocuk (child) : Bir düğümün sol ve sağ bağı aracılığı ile bağlandığı düğümler o düğümün çocuklarıdır.

d)Aile (parent) : Bir düğüm, sağ ve sol bağları ile bağlandığı düğümlerin ailesidir.

e)Yaprak (leaf) : Bir düğümün kök düğümünden olan uzaklığıdır

Soru:(Klasik)Düğüm sayısını bulan fonksiyonu yazınız?

Yanıt:

```
int size(btree *root)
{
    if(root==NULL)
        return 0;
    return 1+size(root->left)+size(root->right);
}
```

Soru:(Klasik)Verilen bir q1 kuyruğundaki tek sayı içeren elemanları yeni bir q2 kuyruğuna,çift sayı içeren elemanları yeni bir q3 kuyruğuna ekleyen bir fonksiyon yazınız.

Yanıt:

```
void ekle(queue *q1)
{
    queue q2,q3;
    initialize(&q2);
    initialize(&q3);
    while(!isEmpty(q1))
    {
        int x=dequeue(q1);
        if(x %2==1)
```

```
        enqueue(&q2,x);
    else
        enqueue(&q3,x);
}
}
```

Soru:(Klasik)Fonksiyona parametre olarak gelen sayının tam bölenlerini tanımlanan “q” kuyruğuna sırasıyla ekleyen fonksiyonu yazınız

Yanıt:

```
void bölen(int x,queue *q)
{
    int i;

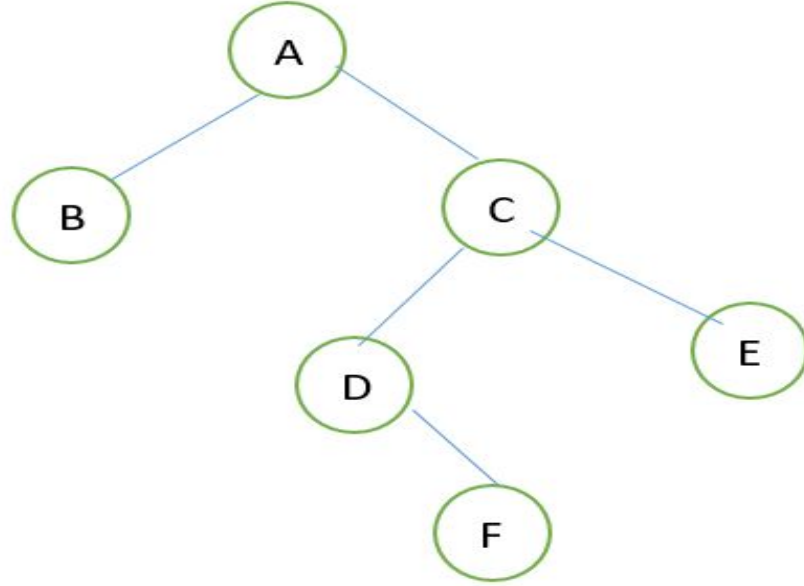
    for(i=1;i<=x;i++)
    {
        if (x % i ==0)
        {
            enqueue(q,i);
        }
    }
}
```

Soru:(Klasik)Yaprakları kırılmış bir ağaçtım. datalarını toplayan fonksiyon yazınız.

Yanıt:

```
int sumLeaves(BTREE *root)
{
    if(root==NULL)
        return 0;
    if(root->left==root->right)
        return root->data;

    return sumLeaves(root->left) +sumLeaves(root->right);
}
```



Soru:(Test)Yukarıdaki AVL ağacının hangi düğünde dengeyi bozan düğüm hangisidir?

Yanıt: Cevap A

Soru:(Test) 1 milyon düğümü olan bir AVL ağacında yükseklik kaçtır?

A)10 B)20 C)100 D)1000 e)999999

Yanıt:B (20) **Bu yükseklik değeri tam olarak söylenemez ama bu civardadır denilebilir. 20 den az olamaz. 100 de olamaz (HK)**

Soru:(Test)

i)Printers

ii)CPU

iii)Recursion

Yukarıdakilerden hangisi veya hangileri priority queue lerde kullanılır?

A)Yalnız i B.)Yalnız iii C)i ve ii D)i,ii,iii

Cevap: C

Soru:(Test) n düğümlü bir BST ağacında maximum yükseklik kaçtır?

A)n B)2n C)n² D)n-1

Yanıt: D (n-1)

Soru:

I.Bir ikili ağaçta, k seviyesindeki maksimum düğüm sayısı 2^k 'dır.

II.Bağlı listelerden farkı nonlinear(dogrusal olmayan) olmasıdır.

III.İki düğüm arasında birden fazla kenar olamaz.

İkili ağaçlarla ilgili olarak hangisi yada hangileri doğrudur?

A)II ve III B) Yalnız III C)I,II ve III D)I ve II E)Yalnız II

Yanıt: C

Soru(Test):

60-70-65-80-72-100-90-82-91-75

Bir Binary Heap olsun. Aşağıdakilerden hangisi 100'ün parent ıdır?

A)60 B)91 C)65 D)80

Yanıt: C

SORU:

A 4-ary max heap is like a binary max heap, but instead of 2 children, nodes have 4 children. Find mapping.And insert function for max heap

Cevap:

Parent(i) = $(i+2)/4$ (aşağıya yuvarlama ile),

1.child = $4i-2$,

2.child = $4i-1$,

3.child = $4i$,

4.child = $4i+1$

```
İnt insertMax(int A[], int key){  
    counter++;
```

```

    if(counter<=QUEUE_SIZE){
        int index = counter;
        A[index] = key;
        while(index != 1 && A[(index+2)/4]<A[index]){
            swap(A[(index+2)/4],A[index]);
            index = (index+2)/4;
        }
    }
}

```

Soru :

What is the smallest and largest number of nodes in a heap of height 8? What is the height of a heap with 600 nodes ?

Cevap:(A)

A-) min: 256 max:511 height=9

B-) min: 255 max:511 height=8

C-) min: 256 max:512 height=9

D-) min: 255 max:512 height=9

E-) min: 256 max:511 height=8

Soru.

Ağaçlarda bağlı bir liste olduğuna göre klasik bağlı listelerden farkı nedir?

Cevap.

Ağaçlar, klasik listeler gibi doğrusal yapıda değildir.

Soru(Klasik):Tüm düğümlerin toplamını bulan bir fonksiyon yazınız?

Yanıt:

```

int sumAllNodes(BTREE *root,int sum)
{

```



```

        if(root!=NULL)
        {
            sum+=root->data;
            sumAllNodes(root->left,sum);
            sumAllNodes(root->right,sum);
        }
        else
            return sum;
    }

```

Fonksiyon çağırma: sumAllNodes(root,0);

Soru(Klasik):Tüm sağ çocuklar ile sol çocukların yerlerini birbirleri ile değiştiren fonksiyonu yazınız

(inorder sıralandığında büyükten küçüğe doğru sıralanacak şekilde değiştiren fonksiyonu yazınız)

Yanıt:

```

void mirror (BTREE root){
    if(root==NULL)
        return;
    mirror(root->left);
    mirror(root->right);
    BTREE temp=root->left;
    root->left=root->right;
    root->right=temp;
}

```

Q boş olmayan bir kuyruk ve S bir yığın. Sadece S ve Q'yu kullanarak kuyruğu ters çeviren fonksiyonu yazınız.

Yanıt:

```

void cevir( )
{
    while(!isEmpty(Q)) {
        push( S, dequeue(Q));
        while(!isEmpty(S)){
            enqueue(Q,pop(S));
        }
    }
}

```

```
}
```

Q) Write a function Addmax that adds the largest node value to all nodes (including itself) in a given linked list

```
struct node *addmax(struct node *head){
```

```
    struct node * temp=head;
```

```
    int max=head->data;
```

```
    while(temp!=NULL){
```

```
        if(temp->data>max)
```

```
        {
```

```
            max=temp->data;
```

```
        }
```

```
        temp=temp->next;
```

```
    }
```

```
    temp=head;
```

```
    while(temp!=NULL)
```

```
    {
```

```
        temp->data+=max;
```

```
        temp=temp->next;
```

```
    }
```

```
    return head;
```

```
}
```

Soru:(Test)60-70-65-80-72-100-90-82-91-75

Bir binary Heap'imiz olsun.Min heap'e göre " 71" eklersek yeni heap'imiz nasıl olur?

A)60-70-65-80-72-100-90-82-91-75-71

B)60-70-71-65-80-72-100-90-82-91-75

C)71-60-70-65-80-72-100-90-82-91-75

D)60-70-65-80-71-100-90-82-91-75-72

Cevap D

Soru:(Test)

- i.)Recursive function calls use queue
- ii)A tree is a graph that may have cycle
- iii)The length of a path is the number of edges on that path
- iv)Every node in a tree has at most 2 children

Which ones right?

A)i and ii B) only ii C)only iii D)i,ii,iii

Answer: C

Write a code that find the avarage of all even numbers in BST and add this avarage to all odd numbers in same BST

(Answer will be in image)

```
int findeven(BTNode root) {
```

```
    if (root != NULL) {
```

```
        if (root->data % 2 == 0)
```

```
            return root->data + findeven(root->left) + findeven(root->right);
```

```
        else
```

```
            return findeven(root->left) + findeven(root->right);
```

```
    }
```

```
    else return 0;
```

```
int size(BTNode root) {
```

```
    if (root == NULL)
```

```
        return 0;
```

```
    else return 1 + size(root->left) + size(root->right);
```

```
int evenaverage, int n;  
double average = evenaverage / n;
```

GLOBAL TANIMLAMA ☺

```
void add(BTNode root, double average) {
```

```
    if (root != NULL)
```

```
        if (root->data % 2 != 0)
```

```
            root->data = root->data + average;
```

```
        add(root->left);
```

```
        add(root->right);
```

```
    else
```

```
        add(root->left);
```

```
        add(root->right);
```

```
}
```

The number of edges from the root to the node is called _____ of the tree.

- a) Height
- b) Depth
- c) Length
- d) None of the mentioned

Cevap: B

What is the time complexity for finding the height of the binary tree?

- a) $h = O(\log \log n)$
- b) $h = O(n \log n)$
- c) $h = O(n)$
- d) $h = O(\log n)$

Cevap: C

Which of the following is not an advantage of trees?

- a) Hierarchical structure
- b) Faster search
- c) Router algorithms
- d) Undo/Redo operations in a notepad

Cevap: D

In a full binary tree if there are L leaves, then total number of nodes N are?

- a) $N = 2L$
- b) $N = L + 1$
- c) $N = L - 1$
- d) $N = 2L - 1$

Cevap: D

A Priority-Queue is implemented as a Max-Heap. Initially, it has 5 elements.

The level-order traversal of the heap is given below: 10, 8, 5, 3, 2 Two new elements "1" and "7" are inserted in the heap in that order.

The level-order traversal of the heap after the insertion of the elements is

- A) 10, 8, 7, 5, 3, 2, 1
- B) 10, 8, 7, 2, 3, 1, 5

C)10,8,7,1,2,3,5

D)10,8,7,3,2,1,5

Cevap:D

SORU: Which of the following is not a type of data structure?

- A) Primitive data structure
- B) Simple data structure
- C) Linear - nonlinear data structure
- D) Complicated data structure

CEVAP: D

What is an AVL tree?

- a) a tree which is balanced and is a height balanced tree
- b) a tree which is unbalanced and is a height balanced tree
- c) a tree with three children
- d) a tree with at most 3 children

Cevap: A

Why we need to a binary tree which is height balanced?

- a) to avoid formation of skew trees
- b) to save memory
- c) to attain faster memory access
- d) to simplify storing

Cevap: A

SORU: Which of the following is the basic data structures?

- A) Union
- B) Stack
- C) Tree
- D) Integer

CEVAP: D

Percolate up and down are used for

- A) AVL trees
- B) B-trees
- C) Circular queue
- D) Binary heaps

Cevap: D

SORU: Which of the following data structure is used by recursive functions?

- A) Queue
- B) Graph
- C) Stack
- D) Boolean

CEVAP: C

SORU: Which of the following can not be done with linked lists?

- A) Binary heaps
- B) Stacks
- C) Queues
- D) Trees

CEVAP: A

A node with key 8 has a left child with key 10. Which of the following objects could this node be found in?

- A) Binary Search Tree
- B) Max Heap
- C) Min Heap
- D) Two of the above

Cevap: C

SORU: Where is the smallest element a binary search tree?

- A) Root
- B) Leftmost
- C) Rightmost
- D) It can be anywhere

CEVAP: B

SORU: Which one of the following is not in the queue data structure?

- A) Front
- B) Top
- C) Rear

D) Counter

CEVAP: B

SORU: Why keep the first element index in the queues?

- A) Because the elements are removed from the first index
- B) For fun
- C) To use memory more efficiently
- D) Because the first element is so important

CEVAP: A

SORU: Which one of the following do not have right and left child concept?

(Aşağıdakilerden hangisinin sağ ve sol çocuk kavramı yoktur?)

- A) BST
- B) Double Linked List
- C) Binary Heap
- D) AVL Tree

CEVAP: B

SORU: Which of the following data structure is linear type?

- A) Strings
- B) Lists
- C) Queues
- D) All of above

Yanıt: D

Soru:

Yanıt:

Soru: The condition $Top = -1$ indicates that

- a) Stack is empty
- b) Stack is full
- c) Stack has only one element
- d) None of these

Yanıt: A
