**PAMUKKALE ÜNİVERSİTESİ**

Name : 09.11.2013

Student Number :

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**VERİ YAPILARI ARASINAV SORULARI**

1. Write the fibonacci method that recursively calculates Nth Fibonacci number.(N. Fibonacci sayısını özyinelemeli hesaplayan aşağıdaki fibonacci metodunu yazın)**(10P)**

int fibonacci(int N)

{

}

1. Postfix evaluation algorithm is given below.Write the PostfixEvaluate method. Numbers in postfix expression is between 0 and 9.(Every character in string is a token)(Verilen postfix bir ifadenin değerini hesaplama algoritması aşağıda verilmiştir. Algoritmaya göre hesaplamayı yapacak PostfixEvaluate metodunu yazınız. Postfix ifade içindeki sayılar 0-9 arasında olacaktır.(İfadedeki her bir karakter bir token))**(20P)**

**Algorithm: Postfix Evaluation**

**1. Create an operand stack**

**2. Convert the input string into tokens**

**3. Scan the token from left to right**

**If the token is an operand push it to the stack**

**If the token is an operator pop 2 operands from stack and do the operation(\*+-/), then push the result to the stack**

**4. Pop the stack and return the result**

static double PostfixEvaluate(string exp)

{

}

1. Add the numbers below to an empty binary search tree respectively and write Preorder, Inorder and Postorder traversals.(Aşağıda sayıları verilen sırada boş bir ikili arama ağacına ekleyin ve ağacın Preorder, Inorder ve Postorder dolaşmalarını yazın.)**(20P)**

13, 3, 4, 12, 14, 10, 5, 1, 8, 2, 7, 9, 11, 6, 18

1. Write the Linked List methods below(Count method is provided)(Aşağıdaki bağlı liste metodlarını yazın(Count metodu yazılmış varsayılacak))

class Node<T>

{

public T element;

public Node<T> next;

}

class LinkedList<T>

{

public Node<T> head;

public int Count();//Implemented

}

* 1. Swap method to swap values of two items at given indexes(Verilen indis değerlerindeki elemanları takas eden(değiştiren))**(10P)**

public void Swap(int index1, int index2)

{

}

* 1. Reverse method to reverse the order of items(Swap method can be used)(Elemanların sırasını tersine çeviren(swap metodu kullanılabilir))(A,B,C->B,C,A)**(10P)**

public void Reverse()

{

}

1. Below is a **function** method written for binary search trees. What is the output of program when it is run?(Insert method should be assumed as written)(Aşağıda ikili arama ağaçları için **function** metodu yazılmıştır. Uygulama çalıştırıldığında programın çıktısı ne olur?(Insert metodu yazılmış varsayılacaktır))**(20P)**

Output(Program Çıktısı)

class BSTNode<T>

{

public BSTNode<T> LeftTree;

public BSTNode<T> RightTree;

public T Value;

}

class BST<T>

{

public BSTNode<T> root;

public void Insert(T val) { }//Implemented

public void function()

{

Queue<BSTNode<T>> q = new Queue<BSTNode<T>>();

q.Enqueue(root);

while (q.Count > 0)

{

BSTNode<T> current = q.Dequeue();

if (current != null)

{

q.Enqueue(current.LeftTree);

q.Enqueue(current.RightTree);

Console.WriteLine(current.Value);

}

}

}

}

class Program

{

public static void Main(String[] args)

{

BST<int> tree = new BST<int>();

int[] array = new int[] { 8, 4, 12, 2, 6, 14, 10, 5, 7, 1, 3, 11, 13, 9, 15 };

foreach (int val in array)

{

tree.Insert(val);

}

tree.function();

}

}

1. Write the smash method for stacks as described below. If the top value of stack is greater than the new value then new value is pushed to the stack. Otherwise top element is replaced with new value and comparision is made with the item below. Until a greater value than new value appears in stack replacing is continued.(Push, Pop, isEmpty, Peek methods are already implemented)(Aşağıda anlatıldığı şekilde smash metodunu yazınız. smash metoduna parametre olarak verilen değer yığının üst elemanından(top) küçükse yığına ekleme işlemi yapılır(push). Aksi takdirde yeni eleman en üstteki elemanla değiştirilir(ezme) ve bu ezme işlemi yeni elemandan büyük bir değer görene kadar devam eder.(Push, Pop, isEmpty, Peek metodları yazılmış varsayılacak))**(10P)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| top→ | 2 |  |  |  |  |
|  | 4 |  |  |  |  |
|  | 5 |  |  | top→ | 9 |
|  | 8 | top→ | 10 |  | 10 |
| *smash of 8, 5, 4 and 2* | | *smash of 10* | | *smash of 9* | |

public smash(int value)

{

}

Süre:120 dk Başarılar

Öğr.Gör. Şevket Umut ÇAKIR