

Manisa Celal Bayar Üniversitesi Yazılım Mühendisliği Bölümü
YZM 2116 – Veri Yapıları Dersi

Quiz#1
Bahar 2018

| | | | |
|---------------|-----------------------|------------------|------|
| Adı ve Soyadı | YANIT ANAHTARI | Öğrenci Numarası | |
| Grubu | | İmza | |
| Tarih | | Not | /100 |

Soru#1 (25 puan): Aşağıda bir döngüsel (dairesel) kuyruk yapısı sunulmuştur. Buna göre, kuyrukta gerçekleştirilen işlemlere göre “front” ve “rear” değerlerini uygun şekilde Tabloda doldurunuz:

| | | | | | |
|---------------------|-----|-----|-----|-----|-----|
| front = 1, rear = 2 | [0] | [1] | [2] | [3] | [4] |
| | | 51 | 13 | | |

| İşlem | Front | Rear |
|------------------|-------|------|
| Queue.Remove() | 2 | 2 |
| Queue.Insert(5) | 2 | 3 |
| Queue.Insert(12) | 2 | 4 |
| Queue.Insert(9) | 2 | 0 |
| Queue.Remove() | 3 | 0 |

Soru#2 (25 puan): Aşağıda verilen kod parçasının Big-O analizini (çalışma süresi bakımından) yapınız. İlgili kod parçası için, en iyi durum ve en kötü durum senaryoları söz konusu ise her ikisi için Big-O karmaşıklıklarını yazınız. Ara işlemlerin yazılmasına gerek yoktur.

```
public static boolean method2(int[] array, int n, int key)
{
    int indexOfKey = -1;
    for (int index = 1; index <= n - 1; index++)
    {
        if (array[index] == key)
            indexOfKey = index;
    }
    if (indexOfKey > 0)
        return true;
    else
        return false;
}
```

En kötü durum: O(n)
En iyi durum: O(1)

Soru#3 (50 puan): Sondan başa ya da baştan sona okunduğunda aynı içeriği veren sözcük, kalıp ya da tümceler, **“palindrom”** olarak tanımlanmaktadır. Örneğin, boşluk ve noktalama işaretlerini göz ardı ettiğimizde, *“Race car”* bir palindromdur. Benzer şekilde, *“A man, a plan, a canal: Panama”* yine bir palindromdur. **Buna göre, girilen bir stringin palindrom olup olmadığını yığın(stack) kullanarak belirleyiniz.** [Gerçekleştirmede, C#, Java, C vb. diller kullanılabilir. Temel yığın operasyonları kullanılarak çözüm, sözde kod ile algoritmik olarak da ifade edilebilir.]

- Boşluk ve noktalama işaretleri dışında kalan “incelenmesi gereken karakter” sayısını N olarak belirle.
- Girdiler içerisinde ilk $N/2$ karakteri yığına “push” operasyonu ile ekle.
- Eğer N değeri tek sayı ise (N ikiye tam bölünemiyorsa, ilgili karakteri) yığına dâhil etme.
- Sırasıyla yığından “pop” operasyonu ile çıkartılan her bir eleman ile girdiler arasındaki $N/2$ karakteri karşılaştır.
- Tek olması nedeniyle yığına dâhil edilmeyen karakter dışında, eğer girdi seti ile yığından kaldırılan karakterler birbirleriyle eşleşmiyorsa, ilgili ifade “palindrom” değildir. Aksi takdirde, ilgili ifade palindromdur.

```
import java.util.Stack;

public class PalindromeTest {

    public static void main(String[] args) {

        String input = "test";
        Stack<Character> stack = new Stack<Character>();

        for (int i = 0; i < input.length(); i++) {
            stack.push(input.charAt(i));
        }

        String reverseInput = "";

        while (!stack.isEmpty()) {
            reverseInput += stack.pop();
        }

        if (input.equals(reverseInput))
            System.out.println("Yes! that is a palindrome.");
        else
            System.out.println("No! that isn't a palindrome.");

    }
}
```