



Ad Soyad: \_\_\_\_\_ Öğrenci No: \_\_\_\_\_ İmza: \_\_\_\_\_

**SINAV BİLGİLERİ:** Toplam süre: 75 dakika.**Soru 1 - Bubble Sort Algoritması**

10 puan

**Örnek:** Aşağıda **[8, 3, 5, 1]** başlangıç listesinin Bubble Sort algoritması ile küçükten büyüğe sıralanma adımları gösterilmiştir.

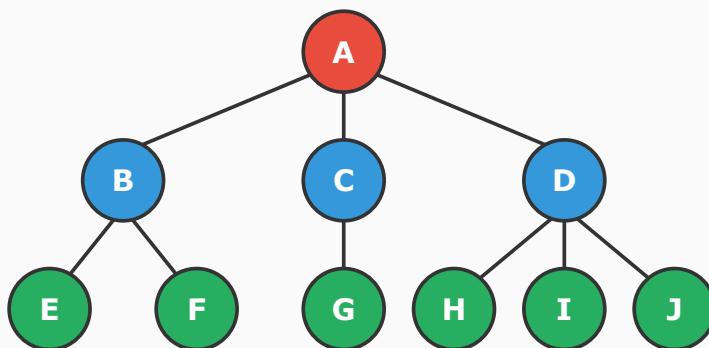
Geçiş	Adım	Liste Durumu
1	1	[3, 8, 5, 1]
	2	[3, 5, 8, 1]
	3	[3, 5, 1, 8]
2	4	[3, 5, 1, 8]
	5	[3, 1, 5, 8]
	6	[3, 1, 5, 8]

Geçiş	Adım	Liste Durumu
3	7	[1, 3, 5, 8]
	8	[1, 3, 5, 8]
	9	[1, 3, 5, 8]
4	10	[1, 3, 5, 8]
	11	[1, 3, 5, 8]
	12	[1, 3, 5, 8]

**[5, 1, 4, 2, 0]** listesini Bubble Sort algoritması ile yukarıdaki örneğe benzer şekilde adım adım sıralayınız.

Soru 2 - Ağaç Veri Yapısı

10 puan



Yukarıdaki ağaç yapısını inceleyerek aşağıdaki soruları cevaplayınız:

- **a)** Kök (root) düğüm hangisidir?
- **b)** Yaprak (leaf) düğümleri yazınız.
- **c)** B düşümünün çocuklarını yazınız.
- **d)** D düşümünün çocuklarını yazınız.
- **e)** F düşümünün ebeveyni (parent) hangisidir?
- **f)** H düşümünün ebeveyni (parent) hangisidir?

Soru 3 - Dijkstra'nın İki-Stack Algoritması

10 puan

**Dijkstra'nın İki-Stack Algoritması** hakkında aşağıdaki soruları cevaplayınız:

- **a)** Bu algoritma hangi problemin çözümünde kullanılır?
- **b)** Algoritma neden "iki stack" kullanır? Her bir stack ne amaçla kullanılır?
- **c)** Algoritmanın temel çalışma mantığını adım adım sözel olarak açıklayınız. (İfade taranırken sayı, operatör, açık parantez ve kapalı parantez görüldüğünde ne yapılır?)

Ad Soyad: \_\_\_\_\_ Öğrenci No: \_\_\_\_\_ İmza: \_\_\_\_\_

**Soru 4 - Stack Veri Yapısı**

15 puan

```

1 import numpy as np
2
3 class NumpyStack:
4     def __init__(self, capacity):
5         self.capacity = capacity
6         self.data = np.zeros(capacity, dtype=int)
7         self.top = 0
8
9     def a(self, value):
10        if self.top == self.capacity:
11            print("Hata a")
12            return False
13        self.data[self.top] = value
14        self.top += 1
15        return True
16
17     def b(self):
18        if self.top == 0:
19            print("Hata b")
20            return None
21
22     def c(self):
23         self.top -= 1
24         val = self.data[self.top]
25         self.data[self.top] = 0
26         return val
27
28     def is_empty(self):
29         return self.top == 0
30
31     def size(self):
32         return self.top
33
34
35
36

```

Yukarıdaki kodda **a**, **b** ve **c** metot isimleri ile **"Hata a"**, **"Hata b"** ve **"Hata c"** hata mesajları anlamsız şekilde verilmiştir.

- a) **a**, **b** ve **c** metot isimlerini, metodların yaptıkları işleri yansıtacak şekilde nasıl değiştireceğiniz? (push, pop, peek)
- b) **"Hata a"**, **"Hata b"** ve **"Hata c"** hata mesajlarını, sınıf kullanıcısının anlamlı hata mesajları görmesini sağlayacak şekilde nasıl değiştireceğiniz?

## Soru 5 - Bağlı Liste

15 puan

```

1  class Node:
2      def __init__(self, data):
3          self.data = data
4          self.next = None
5
6  class SinglyLinkedList:
7      def __init__(self):
8          self.head = None
9          self.size = 0
10
11     def prepend(self, data):
12         new_node = Node(data)
13         new_node.next = self.head
14         self.head = new_node
15         self.size += 1
16
17     def append(self, data):
18         new_node = Node(data)
19         if not self.head:
20             self.head = new_node
21         else:
22             current = self.head
23             while current.next:
24                 current = current.next
25             current.next = new_node
26         self.size += 1
27
28     def delete(self, data):
29         if not self.head:
30             return
31         if self.head.data == data:
32             self.head = self.head.next
33             self.size -= 1
34             return
35         current = self.head
36         while current.next:
37             if current.next.data == data:
38                 current.next = current.next.next
39                 self.size -= 1
40                 return
41             current = current.next
42
43     def display(self):
44         elements = []
45         current = self.head
46         while current:
47             elements.append(str(current.data))
48             current = current.next
49         print(f'{elements} -> {elements}')
50
51 liste = SinglyLinkedList()
52 liste.append(10)
53 liste.append(20)
54 liste.prepend(5)
55 liste.display()
56 liste.delete(20)
57 liste.display()
58 liste.prepend(1)
59 liste.display()

```

Yukarıdaki tek yönlü bağlı liste kodunun çıktısını yazınız.

Ad Soyad: \_\_\_\_\_ Öğrenci No: \_\_\_\_\_ İmza: \_\_\_\_\_

**Soru 6 - Arama Algoritmaları**

20 puan

```
def ara(arr, target):
    low = 0
    high = len(arr) - 1
    while low <= high:
        mid = (low + high) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            low = mid + 1
        else:
            high = mid - 1
    return -1

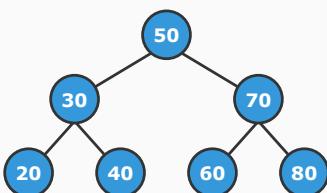
dizi = [3, 7, 12, 18, 24, 31, 38, 45, 52, 67, 73, 89]
print(ara(dizi, 7))
print(ara(dizi, 31))
print(ara(dizi, 50))
```

- a) `ara(dizi, 7)` çağrılarında, aranan değer (7) dizinin hangi elemanları ile karşılaştırılır? Karşılaştırma sırasını yazınız.
- b) `ara(dizi, 31)` çağrılarında, aranan değer (31) dizinin hangi elemanları ile karşılaştırılır? Karşılaştırma sırasını yazınız.
- c) `ara(dizi, 50)` çağrılarında, aranan değer (50) dizinin hangi elemanları ile karşılaştırılır? Karşılaştırma sırasını yazınız.
- d) Bu fonksiyonda kullanılan algoritma **Doğrusal (Lineer) Arama** mı yoksa **İkili (Binary) Arama** mı algoritmasıdır?
- e) Doğrusal arama ve ikili arama algoritmalarından hangisi daha hızlıdır? Neden?
- f) İkili arama algoritmasının doğru sonuç verebilmesi için verilen dizinin hangi özellikleri olması gereklidir?
- g) Kodun ekran çıktısını yazınız.

**Soru 7 - İkili Arama Ağacı (BST)**

10 puan

Aşağıda bir **İkili Arama Ağacı (BST)** verilmiştir:



Bu ağaca sırasıyla aşağıdaki değerler eklenmektedir:

**25, 35, 65, 90**

Tüm değerler eklendikten sonra oluşan BST'nin son halini çiziniz.

*Not: BST kuralı: Her düğüm için, sol alt ağactaki tüm değerler düğümden küçük, sağ alt ağactaki tüm değerler düğümden büyuktur.*

**Soru 8 - İkili Ağaç Oluşturma**

10 puan

```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.left = None  
        self.right = None  
  
# Ağaç 1  
root1 = Node(10)  
root1.left = Node(5)  
root1.right = Node(15)  
root1.left.left = Node(3)  
root1.left.right = Node(8)  
  
# Ağaç 2  
root2 = Node(1)  
root2.left = Node(2)  
root2.right = Node(3)  
root2.left.left = Node(4)  
root2.right.right = Node(5)
```

- a) Yukarıdaki kodda oluşturulan **Ağaç 1** ve **Ağaç 2**'yi çiziniz.
- b) Ağaç 1 bir **İkili Arama Ağacı (BST)** mıdır? Ağaç 2 bir **İkili Arama Ağacı (BST)** mıdır? Nedenini açıklayınız.