

## DATA STRUCTURES

### FINAL EXAM

#### Question 1(35p)

Symbol	A	B	C	D	E
frequency	0.4	0.1	0.2	0.15	0.15

- a) Construct the Huffman coding tree for the following data(yukarıdaki verileri kullanarak Huffman kodlama ağacını oluşturun.)(Kod yok)(5p)
- b) Encode ABACABAD using the coding tree you constructed in (a)(a şıkkında oluşturduğunuz ağacı kullanarak ABACABAD dizgesini kodlayınız)(Kod yok) (5p)
- c) Decode 100010111001010 using the coding tree you constructed in (a) (a şıkkında oluşturduğunuz ağacı kullanarak 100010111001010 dizgesini çözünüz)(Kod yok) (5p)
- d) Write a method which encodes a given text using an already constructed Huffman coding tree. (verilen bir texti Huffman ağacı kullanarak kodlayan metodu yazınız. Huffman ağacı sizin için daha önceden oluşturulmuştur.) (20p)  
The fields in the Node class are(Node sınıfındaki Field'lar); String value, Float frequency, Node left, Node right

## Question 2(30p)

- a) What are the properties that a heap tree should carry.(Bir heap ağacının taşıması gereken özellikleri yazınız) **(5p)**
- b) Considering that you already have a heap tree, write the sort method which will sort the values on this tree. You can save the values that you delete from the tree in each iteration, in an ArrayList. The following methods are already written for you to use. (Elinizde bir heap ağacı olduğunu düşünerek, sıralama metodunu yazınız. Her seferinde heap ağacından sildiğiniz değeri, ArrayList yapısında saklayabilirsiniz. Aşağıdaki metodlar sizin kullanabilmeniz için daha önceden yazılmıştır.) **(25p)**

**Public Node<T> findParent( Node<T> current) //parent bulur**

**Public Node<T> findTheLast() // heap ağacındaki son elemanı(Node) bulur.(En sağdaki yaprağı)**

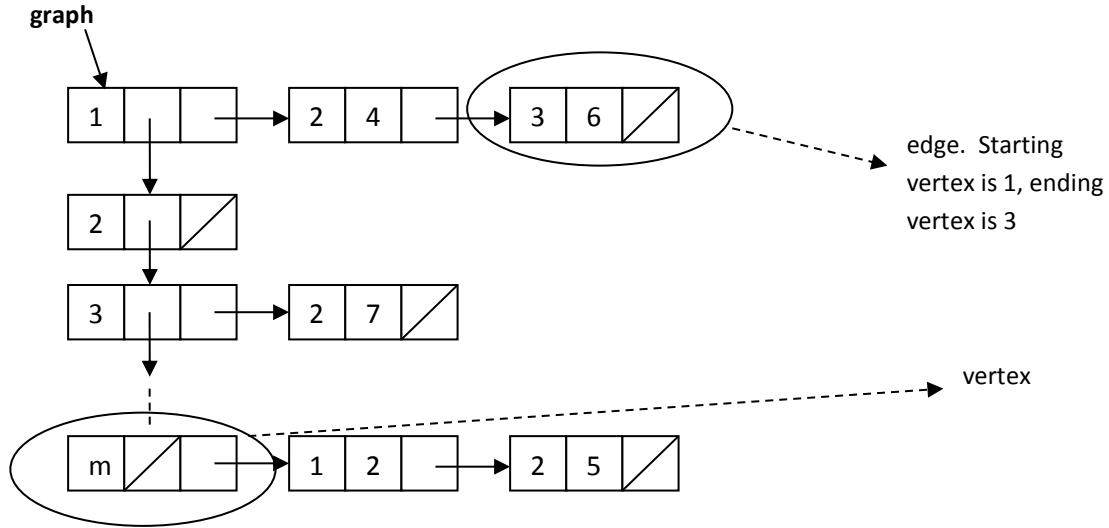
**Question 3(25p)** The figure below shows the structure of a directed graph. Write a method that finds the total weight of the graph. (Aşağıda yönlü bir graf figürü bulunmaktadır. Bu graftaki ağırlıkların toplamını bulan metodu yazınız)

```
class Vertex
```

```
{
    int vertexId;
    Vertex nextVertex;
    Edge edgeLink;
};
```

```
class Edge
```

```
{
    int vertexId;
    int weight;
    Edge nextEdge;
};
```



**Question 4(15p)** Insert the following values into an AVL tree in a given order.(verilen değerleri sırasıyla bir AVL ağacına ekleyiniz.)

**9, 20, 30, 25, 23, 24, 5, 15,17**