

Veri Yapıları ve Algoritmalar Dersi 2. Ödevi

Konu : Huffman Ağacı Oluşturma

Bu ödevde kayıpsız veri sıkıştırma yöntemlerinden Huffman Kodlamanın, Huffman Ağacı oluşturma aşamasına ait algoritmayı tasarlamamız ve gerçekleştirmemiz istenmektedir.

İşlem Adımları :

Aşağıdaki her aşamayı yapmak için gerekli fonksiyonları yazarak Huffman ağacını oluşturunuz.

1. Sıkıştırılması istenen yazıdaki her harfin kullanım sıklığını hesaplayınız. Harfleri ve kullanım sıklıklarını linkli liste yapısında saklayınız.
2. Harflerin kullanım sıklıklarına göre bu düğümleri insertion sort ile küçükten büyüğe sıralayınız.
3. Sıralı linkli listeyi kullanarak aşağıda verilen örnekteki gibi Huffman ağacını oluşturunuz.
4. Huffman ağacını, kökten başlayarak ağacın her seviyesi bir satırda olacak şekilde satır satır ekrana yazdırınız.

Raporunuzda :

- a. Çözümünüze yönelik raporlama kurallarına uygun olarak hazırlanmış açıklama/diyagram/şekil/tablolara
- b. Ödevinize ait C dilinde yazılmış program kodu
- c. Aşağıdaki örnek için programınızın ekran çıktısı olmalıdır.

Girdi	Output
huffman coding is a data compression algorithm	?

Teslim tarihi : Ödev teslim tarihi 6 Nisan 2020 saat 23.59' dur. Teslim işlemleri için aşağıdaki linki kullanınız. **Dosya isimlendirme kuralları linkte belirtildiği gibi olmalıdır.**

<https://forms.gle/EH5CZfkanas13Hm57>

ÖRNEK:

Örnek: Sıkıştırılacak yazı, a, b, c, d ve e harflerinden oluşsun.

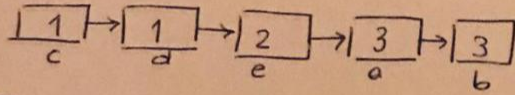
Harflerin yazıda geçme sıklıkları $a=3, b=3, c=1, d=1$ ve $e=2$ olsun.

Huffman ağacını aşağıdaki işlem adımlarını takip ederek oluşturunuz.

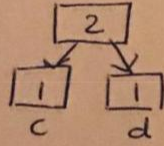
1. Yazıdaki her harfin kullanım sayısını bulunuz. Kolaylık açısından, yazının sadece küçük harflerden oluştuğunu varsayınız.

2. Yazıda geçen harfleri, frekanslarına göre küçükten büyüğe sıralı olacak şekilde linkli liste yapısında saklayınız.

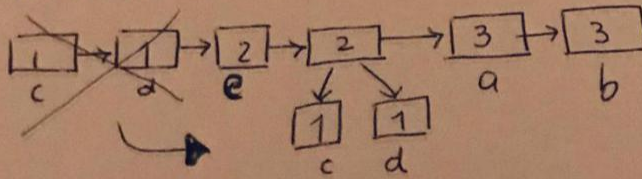
Sıralama için insertion sort kullanınız.



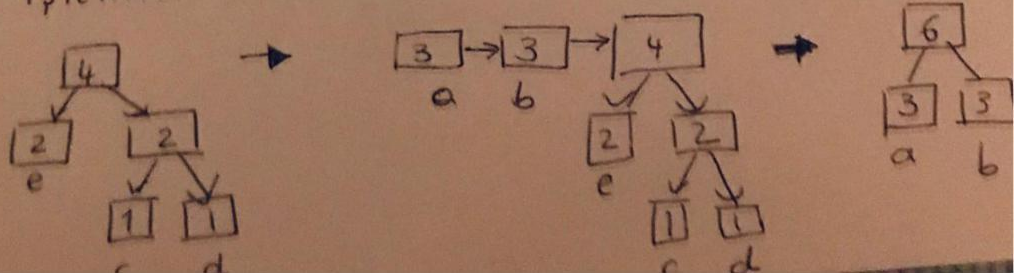
3. En küçük iki sayıyı toplayınız. 1. sayı SOL düğüm, 2. sayı SAĞ düğüm ve toplam değer kök olacak şekilde bir ikili ağaç elde ediniz:

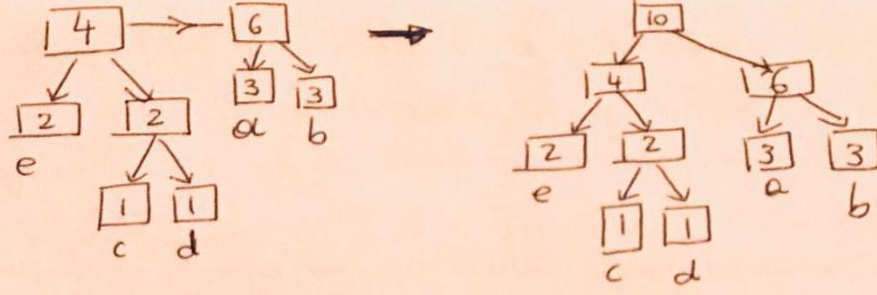


4. Bu ağacı, sırayı bozmayacak şekilde linkli listeye insert ediniz.



5. Aynı işlemleri tekrar ediniz =





Bu ağaçtaki her düğümün harf, kullanım sıklığı, sağ düğüm pointeri, sol düğüm pointer'i ve sonraki düğüm pointeri bileşenleri olmalıdır =

```

typedef struct node
{
    char harf;
    int frequency;
    struct node *left, *right, *next;
} NODE;

```

Ara düğümlerin harf değeri NULL, yaprak düğümlerin ise harfin kendisidir.

6. Ekran çıktısı =

```

10
4 6
2 e 2 3a 3b

```

```

NULL NULL 1c 1d NULL NULL NULL NULL

```

→
Düğüm görünüm için bu NULL değerler yerine boşluk bastırabilirsiniz.