

Praktikum 10

Teksti pakkimine

Ülesanded

- KKP_6. Huffman
- KKP_7. Prefiiskood
- KKP_8. Pakkimine
- KKP_9. Lahtipakkimine
- KKP_10. Prefiiskodeerimine

Iseseisev töö nr 6

Ülesanded

KKP_6. Huffman [Kiho 2003, p.5.2]

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: tekst s (sõnena).

Tulemus: Huffmani algoritmiga leitud koodipuu teksti s jaoks.

Soovitus. Kuna s võib sisaldada suvalisi sümboleid, sh ka erisümboleid, nagu tühemikud ning $[] ; < jmt$, siis koodipuu lehttipudes tuleks hoida mitte sümbolit vaid selle arvkoodi (sõnena).

Kui $sü$ on ühesümboliline sõne (nt $"["$), siis koodipuu lehttipu t märgendiks seada sümboli $"["$ arvkoode sõne kujul $"91"$):

```
t.seadaMärgend( "" + (int)(sü.charAt(0)) );
```

KKP_7. Prefiiskood

Kirjutada ja testida rekursiivne meetod järgmise ülesande lahendamiseks.

Antud: koodipuu kdp , selle tipp t ja 0-1-rada kdp juurtipust tipuni t ,

prefiiskoodide (täiendatav) paisktabel d

Tulemus: iga rippuva tipu u korral prinditud 0-1-rada kdp juurtipust tipuni u ja

paisktabelisse d lisatud paar (sümbol tipus u , leitud 0-1-rada)

Vihje. Tipu u märgendiks oleva (sõnekujul) arvkoodi teisendamine vastavaks ühesümboliliseks sõneks:

```
"" + (char) Integer.parseInt(u.märgend00())
```

KKP_8. Pakkimine

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: tekst s (sõnena).

Tulemus: Huffmani algoritmiga leitud koodipuu teksti s jaoks ja selle põhjal prefiks-kodeeritud s .

KKP_9. Lahtipakkimine

Kirjutada ja testida meetod järgmise ülesande lahendamiseks.

Antud: prefiks-kodeeritud s ja kodeerimiseks kasutatud koodipuu.

Tulemus: lahtipakitud s .

Soovitus. Lahtipakkimiseks teha pöörd-paiskabel (prefiiskood, sümbol).

KKP_10. Prefikskodeerimine

KKP_10-1. Tippude pöördkahendkuhja klass

Lähtudes klassist *KahendkuhiDq.java* (vt ka ül KKP_5, Slaidid_pr_9) töötada välja klass *KahendkuhiTp.java*, milles kirjeks kuhjas on objekt tüüpi *Tipp* ja kirje (tipu *t*) võtmeks on tipu (*t*) väljal *f* olev arv (tipu *t* võti = `Double.parseDouble(t.väli("f"))`).

```
public class KahendkuhiTp{           //           Märkus /**/ -- meetodi sisu muutunud
    protected ArrayList<Tipp> a; // kirjed kuhjas
    public Integer vasak(int k)
    public Integer parem(int k)
    public Integer ülemus(int k)
    public void viia_alla(int i)
    public void viia_üles(int i)
    public Tipp võttaKirje() /**/ // KahendkuhiDq oli: võtta_tipuVõti()
    public void lisadaKirje(Tipp b)
    public boolean onTühi()
    public double võti(int i) /**/
    public double tipuVõti()
    public String toString() /**/
    public Kahendpuu puuna() /**/
    public int n(){return n;} // piilumeetod (uus)
    private void vahetada(int i, int j) /**/
    private boolean kuhjatingimus(int i, int j)
    public static void main(String[] args){// test /**/
```

3.11.2017

KKP_10-2. Teksti sümbolite esinemissageduste tabel

Koostada meetod *sagedustabel*.

```
static Hashtable<Character, Double> sagedustabel(String s){
    // Antud: tekst sõnena s
    // Tulemus: luuakse ja tagastatakse sümbolite esinemissageduste tabel,
    //          tabeli element: <sümbol | sagedus>
    int n = s.length();
    Hashtable<Character, Double> tabel = new Hashtable<Character, Double>();
    // loendada sümbolite kordused (tabeli element: <sümbol | esinemiste arv>) :
    //.....
    // kordused asendada sagedustega (tabeli element: <sümbol | sagedus>) :
    //.....
    return tabel;
} //sagedustabel
```

KKP_10-3. Huffmani koodipuu. Koostada meetod *huffman*.

```
static Kahendpuu huffman(Hashtable<Character, Double> sagedustabel){
    // Antud: sümbolite esinemissageduste tabel
    // Tulemus: konstrueeritakse ja tagastatakse vastav Huffmani koodipuu

    KahendkuhiTp Q = new KahendkuhiTp(); // front (veel vaatlemata tipud)
    // NB! Fronti Q peetakse tippude pöörd-kahendkuhjana!
    // teha frondi tipud ja vastav isoleeritud tippudega graaf g:
    Graaf g = new Graaf();
    for (char c : sagedustabel.keySet()){
        Tipp t = new Tipp("" + (int)c); // tipu märgendiks sümboli c arvkode
        t.seadaVäli("f", "" + sagedustabel.get(c)); // väljale f selle sümboli sagedus
        Q.lisadaKirje(t); // panna fronti
        g.lisada(t); // panna graafi
    } //for
    while(Q.n() > 1){ // Õpik, joonis 5.6
        //.....
    } //while
    Tipp juur = Q.võttaKirje(); // ainus fronti jäänud tipp
    g.esikohale(juur); // paigutada esikohale (saab kahendpuu juurtipuks)
    return new Kahendpuu(g); // graaf g kahendpuuna
} //huffman
```

3.11.2017

KKP_10-4. Prefikskodeerimise ja dekodeerimise tabelid . Koostada meetodid *prefikskood* ja *dekodTabel*.

```
static void prefikskood(Kahendpuu kdp, Tipp t, String tee, Hashtable<String, String> d){
    // Antud: Huffmani koodipuu kdp, selle tipp t ja
    //         tee-- 0-1-rada (sõnena) kdp juurtipust tipuni t ning
    //         d-- täiendatav prefikskoodide tabel(elemendid kujul <sümbol | prefikskood>)
    // Eeldus: lehttipus on märgendiks üks sümbol(arvkoodina)
    // Tulemus: iga lehttipu u korral täiendatud tabel d
    //           elemendiga <sümbol tipus u | 0-1-rada kdp juurtipust tipuni u>
    if (t == null)return; // baasjuht
    if (kdp.aste(t) == 0){ // kui t on lehttipp puus kdp
        String m = "" + (char)(Integer.parseInt(t.märgend00())); // arvkoodist sümboliks
        d.put(m, tee); // m|tee => tabelisse
    }
    prefikskood(kdp, kdp.vasakAlluv(t), tee+"0", d); // edasi vasaku alluvaga
    //..... // edasi parema alluvaga
} //prefikskood

static Hashtable<String, String> dekodTabel(Hashtable<String, String> d){
    // Antud: kodeerimistabel (prefikskoodide tabel) d (rida: <sümbol | prefikskood>)
    // Tulemus: luuakse ja tagastatakse dekodeerimistabel (rida: <prefikskood | sümbol>)
    Hashtable<String, String> tulem = new Hashtable<String, String>();
    //.....
    return tulem;
} //dekodTabel
```

3.11.2017

KKP_10-5. Prefikskodeerimine ja dekodeerimine

Koostada meetodid *kodeerida* ja *dekodeerida*.

```
static String kodeerida(String tekst, Hashtable<String, String> d){
    // Antud: tekst sõnena ja kodeerimistabel d (rida: <sümbol | prefikskood>)
    // Tulemus: luuakse ja tagastatakse tekst kodeerituna prefikskoodides
    String tulem = "";
    for( int i = 0; i <= tekst.length()-1; i++ ) .....;
    return tulem;
}//kodeerida

static String dekodeerida(String kodtekst, Hashtable<String, String> dekodTabel){
    // Antud: kodtekst -- tekst kodeerituna prefikskoodides ja
    //          d -- dekodeerimistabel (rida: <prefikskood | sümbol>)
    // Tulemus: dekodeeritakse kodtekst, ja tagastatakse
    String tulem = "";
    //.....
    return tulem;
}//dekodeerida
```


Iseseisev töö nr 6

Esitamise tähtaeg rühmal inf1(J. Liivi 2-202, E 12 -14): **18. november 2017, kell 12.15**

Esitamise tähtaeg rühmal inf7(J. Liivi 2-202, T 14 -16): **19. november 2017, kell 14.15**