

Prófdagur og tími: 06.05.2010 09:00-12:00

Prófstaður:
Háskólatorg - St. 102, 103, 104 og 105



HÁSKÓLI ÍSLANDS

Skráðir til prófs: 111

(113)

TÖL203G Tölvunarfræði 2

Skriflegt próf (Vægi: 70%)

Deild: **Iðnaðarverkfræði-, vélaverkfræði- og tölvunarfræðideild**

Kennarar:

Snorri Agnarsson (snorri@hi.is / S: 5254728 / GSM: 8613270) Umsjónarkennari
Hafsteinn Einarsson (hae15@hi.is / GSM: 6949992) Aðstoðarkennari
Hallgrímur Heiðar Gunnarsson (hhg4@hi.is / S: / GSM: 8967121) Aðstoðarkennari

Kennslumisseri: **Vor 2010**

Úrlausnir skulu merktar með nafni

Prófbók/svarblöð:

Línustrikuð prófbók

Hjálparpögn:

Engin leyfileg hjálparpögn

Önnur fyrirmæli:

Aðgangur að prófverkefni að loknu prófi:

Kennslusvið sendir eintak í prófasafn

Einkunnir skulu skráðar í Uglu eigi síðar en 20.05.2010.

Samkvæmt 60. grein Reglna fyrir Háskola Íslands skulu einkunnir birtar í síðasta lagi tveimur vikum eftir hvert próf, nema eftir desemberpróf, þá eftir þrjár vikur. Einkunnir skulu skráðar í Uglu.

Prentað: 30.04.10

Tölvunarfræði 2

Vorpróf 6. júní 2010.

Engin hjálpar gögn eru leyfileg.

Öll dæmi gilda jafnt.

Munið að skrifa notkunarlysingu með forskilyrði og eftirskilyrði fyrir sérhvert stef og fastayrðingu gagna fyrir sérhverja útfærslu gagnamóts.

Athugið vel: Svara þarf tilskildum fjölda dæma úr hverjum hluta prófsins. Að því skilyrði uppfylltu gilda **10 bestu dæmi** til einkunnar. Byrjið því á að svara dæmum sem krefjast stuttra svara og þið getið auðveldlega svarað.

Hluti I: Röksemdafærsla o.fl.

Svarið a.m.k. 1 dæmi úr þessum hluta

1. Gerið grein fyrir meginreglunni um upplýsingahuld (information hiding):
Hver er reglan og hver er tilgangur hennar?
2. Gerið ráð fyrir að klasar A og B séu skilgreindir á eftirfarandi hátt.

```
class A
{
    ...
    // Notkun: y = x.sin(z);
    // Fyrir: -0.5 <= z <= 0.5
    // Eftir: y er innan 0.001 frá réttu
    //         gildi fyrir sin, þ.e.
    //         |y-sin(z)| < 0.001.
    public double sin( double z ) {...}

    // Notkun: y = x.cos(z);
    // Fyrir: -0.5 <= z <= 0.5
    // Eftir: y er innan 0.001 frá réttu
    //         gildi fyrir cos, þ.e.
    //         |y-cos(z)| < 0.001.
    public double cos( double z ) {...}

    // Notkun: y = x.tan(z);
    // Fyrir: -0.5 <= z <= 0.5
    // Eftir: y er innan 0.001 frá réttu
    //         gildi fyrir tan, þ.e.
}
```

```
//           |y-tan(z)| < 0.001.
public double tan( double z ) {...}

// Notkun: y = x.atan(z);
// Fyrir: -0.5 <= z <= 0.5
// Eftir: y er innan 0.001 frá réttu
//         gildi fyrir atan, þ.e.
//           |y-atan(z)| < 0.001.
public double atan( double z ) {...}

...
}

class B extends A
{
    ...
    // Notkun: y = x.sin(z);
    // Fyrir: -0.25 <= z <= 0.25
    // Eftir: y er innan 0.01 frá réttu
    //         gildi fyrir sin, þ.e.
    //           |y-sin(z)| < 0.01.
    public double sin( double z ) {...}

    // Notkun: y = x.cos(z);
    // Fyrir: -0.75 <= z <= 0.75
    // Eftir: y er innan 0.01 frá réttu
    //         gildi fyrir cos, þ.e.
    //           |y-cos(z)| < 0.01.
    public double cos( double z ) {...}

    // Notkun: y = x.tan(z);
    // Fyrir: -0.25 <= z <= 0.25
    // Eftir: y er innan 0.0001 frá réttu
    //         gildi fyrir tan, þ.e.
    //           |y-tan(z)| < 0.0001.
    public double tan( double z ) {...}

    // Notkun: y = x.atan(z);
    // Fyrir: -0.75 <= z <= 0.75
    // Eftir: y er innan 0.0001 frá réttu
    //         gildi fyrir atan, þ.e.
```

```

//           |y-atan(z)| < 0.0001.
public double atan( double z ) {...}

...
}

```

Hverjar af lýsingunum á boðunum `sin`, `cos`, `tan` og `atan` í undirklasanum B eru rangar miðað við nauðsynlegar röksemdafærslureglur? Rökstyðjíð.

Hluti II: Algrím o.fl.

Svarið a.m.k. 2 dæmum úr þessum hluta

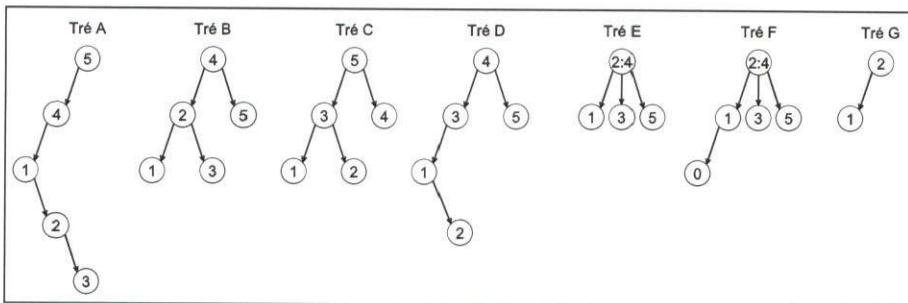
3. Fyllið inn þar sem spurningarmerkin eru, þ.e. skrifið hvað á að koma í stað ?1?, o.s.frv. Þetta eru samtals átta svör.

```

// Notkun: k = leita(f,i,j,x);
// Fyrir: f[i..j-1] er í minnkandi röð
// Eftir: f[i..j-1] er óbreytt, i <= k <= j, og
//         f[i..k-1] > x >= f[k..j-1]
int leita( double[] f, int i, int j, double x ) {
    if( i==j ) return i;
    int m = (i+j)/2;
    if( f[m] ?1? x )
        return leita(f,i,?2?,x);
    else
        return leita(f,?3?,j,x);
}

int leita( double[] f, int i, int j, double x ) {
    int p=i, q=j;
    while( ?4? ) {
        // | >x | óþekkt | <=x |
        //   i     p       q     j
        int m = (?5?)/2;
        if( f[m] ?6? x )
            p = ?7?;
        else
            q = ?8?;
    }
    return p;
}

```



Mynd 1: Ýmis tré.

4. (a) Lýsið hugmyndinni í merge-sort.
 (b) Rökstyðjið að tímaflækja merge-sort sé $O(n \log n)$.
5. (a) Hverjar eru fastayrðingar lykkjunna tveggja í heapsort?
 (b) Rökstyðjið að tímaflækja heapsort sé $O(n \log n)$.
6. Skrifið stef sem finnur rót samfellds falls með helmingunarleit.
7. Gerið ráð fyrir að til sé klasi IntPriQueue í C++ eða Java fyrir forgangsbiðröð heiltalna af ótakmarkaðri stærð, sem hefur sjálfgefinn smið og boðin get, put, og isEmpty með eðlilegum lýsingum þ.a. get boðið fjarlægir og skilar minnstu tölunni í forgangsbiðröðinni.

Notið þennan klasa til að skrifa röðunarstef fyrir fylki heiltalna. Munið að hafa skýra notkunarlysingu (Notkun, Fyrir, Eftir).

Ef get og put boðin hafa tímaflækju $O(f(n))$, þar sem n er fjöldi gilda í forgangsbiðröðinni, og isEmpty hefur tímaflækjuna $O(1)$, hver er þá tímaflækja röðunarstefins?

Hluti III: Gagnamót

Svarið a.m.k. 3 dæmum úr þessum hluta

8. Íhugið mynd 1 á síðu 4 af trjám. Segið til fyrir hvert tré á myndinni hvort það getur verið eitt eða fleiri af eftirfarandi:
 - (a) Tvíleitartré (Binary search tree)
 - (b) AVL-tré (AVL tree)
 - (c) Splay-tré (Splay Tree)

- (d) Rautt-svart tré (Red-black tree) – ef svo er, tilgreinið þá einnig hvaða hnúta má mála rauða til að það gangi upp
- (e) 2-3 tré (2-3 tree)
- (f) Hrúga með hæsta gildi efst (Heap with greatest value on top)
- (g) Hrúga með minnsta gildi efst (Heap with least value on top)

Athugið að hér er ætlast til að öll tvíleitartré séu með gildin í vaxandi *in-order* röð.

Athugið einnig að sama mynd af tré getur vel staðið fyrir fleiri en eina af þessum upp töldu gerðum af trjám.

9. Hver er fastayrðing gagna fyrir AVL-tré?
10. Berið saman annars vegar splay-tré og hins vegar tætitöflur (hash tables) með tilliti til annars vegar hraða og hins vegar hvers konar aðgerðir er unnt að bjóða upp á. Hvor aðferðin er hraðvirkari? Hvor aðferðin gefur kost á fleiri mögulegum (hraðvirkum) aðgerðum? Gefið dæmi um aðgerðir og tímaflækju sem unnt er að bjóða með annarri aðferðinni en ekki hinni.
11. Lýsið skopplistum.
 - (a) Teiknið mynd sem sýnir dæmi um uppsetningu gagna í skopplista af heiltölum.
 - (b) Lýsið í orðum hvernig eftirfarandi aðgerðir eru framkvæmdar á skopp-lista:
 - i. Leit
 - ii. Innsetning
 - iii. Eyðing
12. Gerið grein fyrir muninum á tímaflækju AVL-trjáa, Splay-trjáa og skopp-lista (skip list). Hvert af þessum gagnamótum gefur besta flækjustigsloforð fyrir einstakar aðgerðir? Hvers vegna er samt sem áður hugsanlegt að hin gagnamótin séu betri í einhverjum skilningi? Í hvaða skilningi?
13. Skriftið klasa í Java eða C++ fyrir biðröð heiltalna. Þið megið sleppa því að forrita boðin, nema fyrir aðferðina til að sækja gildi úr biðröðinni, en munið að hafa notkun, forskilyrði og eftirskilyrði fyrir öll boð og munið að hafa skýra fastayrðingu gagna.

Hluti IV: Blandað efni

Ekki þarf endilega að svara neinu dæmi úr þessum hluta, en ekki gleyma að svara 10 dæmum í heild.

14. Skrifið stef sem skilar út 1000 minnstu prímtölurnar.
15. Skrifið forrit sem les strengi frá aðalinntaki og skrifar strengina í vaxandi stafrófsröð á aðalúttak. Forritið þarf að ráða við ótakmarkaðan fjölda strengja (meðan minnisrými leyfir).
16. Skrifið stef sem finnur **hágildi** samfellds falls með helmingunarleit.