Prófdagur og tími: 09.05.2011 13:30-16:30

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

Skráðir til prófs: **126**

128

HÁSKÓLI ÍSLANDS

TÖL203G Tölvunarfræði 2

Skriflegt próf (Vægi: 70%)

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

Kennarar:
Snorri Agnarsson (snorri@hi.is / S: 8613270 / GSM: 8613270) Umsjónarkennari
Þórður Ívar Björnsson (thb45@hi.is) Aðstoðarkennari
Björn Orri Guðmundsson (bog1@hi.is / GSM: 8464422) Aðstoðarkennari
Árni Már Þrastarson (ath86@hi.is / S: 8671751 / GSM: 8671751) Aðstoðarkennari

Kennslumisseri: **Vor 2011**

Úrlausnir skulu merktar með nafni

Prófbók/svarblöð:
Línustrikuð prófbók

Hjálpargögn:
Engin leyfileg hjálpargögn

Önnur fyrirmæli:

Aðgangur að prófverkefni að loknu prófi:
Kennslusvið sendir eintak í prófasafn

Einkunnir skulu skráðar í Uglu eigi síðar en 23.05.2011.

Athugið að einhverjar úrlausnir úr fjölmennum prófum geta verið í þunnum umslögum sem auðvelt er að yfirsjást. Góð vinnuregla er að byrja á því að opna öll umslög, telja úrlausnir og athuga hvort fjöldi stemmir við uppgefinn fjölda sem kvittað var fyrir.

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: 27.04.11

Tölvunarfræði 2

Vorpróf 9. maí 2011.

Engin hjálpargögn eru leyfileg.

Öll dæmi gilda jafnt.

Munið að skrifa notkunarlýsingu 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. Íhugið eftirfarandi lykkjumynstur:

```
// F
while( true )
{
    // I1
    S1
    // I2
    if( C ) break;
    S2
}
// E
```

   Hvaða samband þarf að gilda milli F, I1, I2, S1, C, S2 og E til að lykkja þessi sé rökrétt? Athugið að `break` setningin veldur því að hætt er í lykkjunni.

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? Rökstyðjið.

3. Íhugið eftirfarandi lykkju, sem hefur þann tilgang að reikna margfeldi tveggja heiltalna:

```
// x,y >= 0
int p=0, q=x, r=y;
while( r != 0 ) {
    // x*y == p+q*r, r>=0
    ... (breytum hvorki x né y)
}
// ???
```

- Hvert er rökrétt eftirskilyrði lykkjunnar?
- Hvernig getum við forritað stofninn í lykkjuna á hraðvirkan hátt þannig að fjöldi umferða verði $O(\log y)$? (aðeins má deila eða margfalda með 2, engri annarri tölu).

Hluti II: Algrím o.fl.

Svarið a.m.k. 3 dæmum úr þessum hluta

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

```
// Notkun: k = leita(f,i,j,x);
// Fyrir:  f[i..j-1] er í minnkandi röð og
//         inniheldur a.m.k. eitt gildi sem er
//         minna en x.
// Eftir:  k vísar á fremsta gildi í f[i..j-1]
//         sem er minna en x.
int leita( double[] f, int i, int j, double x ) {
  int p=?1?, q=?2?;
  while( ?3? )
  {
    // | >=x | óþekkt | <x |
    // i     p        q    j
    int m = (?4?)/2;
    if( x ?5? f[m] )
      p = ?6?;
    else
      q = ?7?;
  }
  return p;
}
```

5. Gefið er eftirfarandi stef í C++ eða Java:

```
// Notkun:  split(f,i,j);
// Fyrir:   Svæðið f[i..j-1] inniheldur a.m.k. tvö
//          stök.
// Eftir:   Búið er að víxla gildum í svæðinu
//          f[i..j-1] þ.a. f[i..k-1] <= f[k..j-1],
//          þar sem k er:
//            k = (i+j)/2
void split( double[] f, int i, int j );
```

Skrifið quicksort stef (með lýsingu - notkun, fyrir og eftir) með hjálp þessa stefs. Ekki þarf að forrita split stefið.

6. Tilgreinið tímaflækju eftirfarandi röðunaraðferða miðað við að raðað sé n slembitölum á bilinu $0 \ldots k$, fyrir eitthvert $k > 0$. Tilgreinið einnig hvort tímaflækjan sé tími í versta tilfelli, meðaltími eða innistæðubundin.

(a) Heapsort

(b) Merge-sort

(c) Quicksort

(d) Insertion-sort

(e) Radix-sort

(f) Röðum með því að moka gildunum í AVL tré (sem upphaflega er tómt) og moka þeim síðan út í vaxandi röð

(g) Röðum með því að moka gildunum í Splay tré (sem upphaflega er tómt) og moka þeim síðan út í vaxandi röð

(h) Röðum með því að moka gildunum í skopplista (skiplist) (sem upphaflega er tómur) og moka þeim síðan út í vaxandi röð

7. Lýsið heapsort. Hverjar eru fastayrðingar lykkjanna tveggja í heapsort? Hver er tímaflækja heapsort? Rökstyðjið tímaflækjuna.

8. Lýsið mergesort. Tilgreinið tímaflækju mergesort og rökstyðjið hana.

Hluti III: Gagnamót

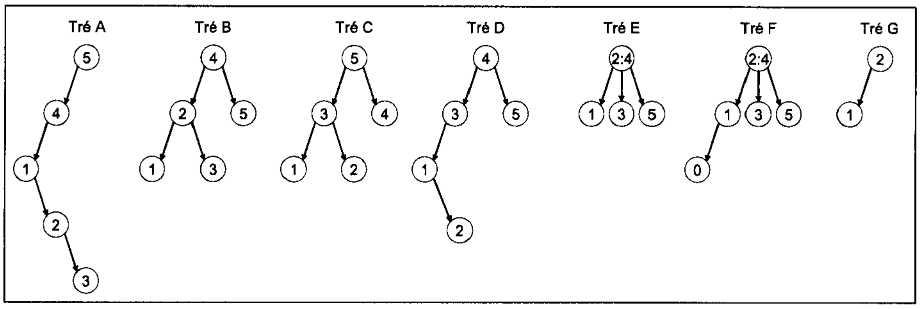
Svarið a.m.k. 3 dæmum úr þessum hluta

9. Íhugið eftirfarandi myndir af trjám. Segið til fyrir hverja mynd hvort hún getur staðið fyrir eitt eða fleiri af eftirfarandi:

   - Tvíleitartré
   - AVL-tré
   - Splay-tré
   - Rautt-svart tré (ef svo er, tilgreinið þá einnig hvaða hnúta má mála rauða til að það gangi upp)
   - 2-3 tré
   - Hrúga með hæsta gildi efst
   - Hrúga með minnsta gildi efst

   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.



10. Hver er fastayrðing gagna fyrir AVL tré?

11. Hver er fastayrðing gagna fyrir rauð-svört tré?

12. Lýsið skopplistum.

    - Teiknið mynd sem sýnir dæmi um uppsetningu gagna í skopplista af heiltölum.
    - Lýsið í orðum hvernig eftirfarandi aðgerðir eru framkvæmdar á skopplista:
        - Leit
        - Innsetning
        - Eyðing

13. Gerið grein fyrir muninum á tímaflækju AVL-trjáa, Splay-trjáa og skopplista (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 tilvikum?

14. Skrifið 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.

15. Skrifið stef (fall, aðferð) sem skilar **stærsta** prímtöluþætti tölu sem skal vera viðfang stefsins.

16. Skrifið forrit sem les fleytitölur frá aðalinntaki og skrifar þær í öfugri lesröð á aðalúttak. Forritið þarf að ráða við ótakmarkaðan fjölda talna (meðan minnisrými leyfir).

17. Skrifið stef sem finnur rót samfellds falls með helmingunarleit.

18. Skrifið forrit sem les eina eða fleiri heiltölur (ótakmarkaðan fjölda) af aðalinntaki og skrifar stærsta sameiginlega deili þeirra, t.d.. ef lesnar eru x, y og z þá skal skrifaða gildið vera stærsta heiltala sem gengur upp í x og y og z.