

## Tölvunarfræði 2/2a

### Sjúkra- og upptökupróf 11. júní 2009.

*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( C1 )
{
    // I1
    S1
    // I2
    if( C2 ) continue;
    S2
}
// E
```

Hvaða samband þarf að gilda milli F, C1, I1, I2, S1, C2, S2 og E til að lykkja þessi sé rökrétt? Athugið að `continue` setningin veldur því að farið er aftur í byrjun lykkjunnar (í næstu umferð) þar sem reiknað er skilyrðið C1 til að athuga hvort halda eigi áfram.

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:

```
System.out.println(2);
int i=3;
while( i!=1000 )
{
    // Búið er að skrifa allar prímtölur
    // minni en i.  3 <= i <= 1000.
    int k=2;
    while( k!=i )
    {
        // Búið er að skrifa allar prímtölur
        // minni en i.  3 <= i <= 1000.
        // Engin prímtala minni en k gengur
        // upp í i.  2 <= k <= i.
        ...
    }
}
```

```

    }
    i++;
}
// ???

```

- Hvert er rökrétt eftirskilyrði ytri lykkjunnar?
- Forritið stofninn í lykkjunni. Vísbending: Þið munuð þurfa að nota break setninguna til að hætta í innri lykkjunni.

## 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 vaxandi röð og inniheldur
//       a.m.k. eitt gildi sem er stærra en x.
// Eftir: k vísar á fremsta gildi í f[i..j-1] sem
//       er stærra 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: k = 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. i < k < j og  
//          f[i..k-1] <= f[k..j-1].  
int 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  $0 \dots k$ , fyrir eitthvert  $k > 0$ . Tilgreinið einnig hvort tímaflækjan sé tími í versta tilfalli, 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 radix sort (bucket sort). Tilgreinið tímaflækju radix sort og rökstyðjið hana.
8. Lýsið mergesort. Tilgreinið tímaflækju mergesort og rökstyðjið hana.

### Hluti III: Gagnamót

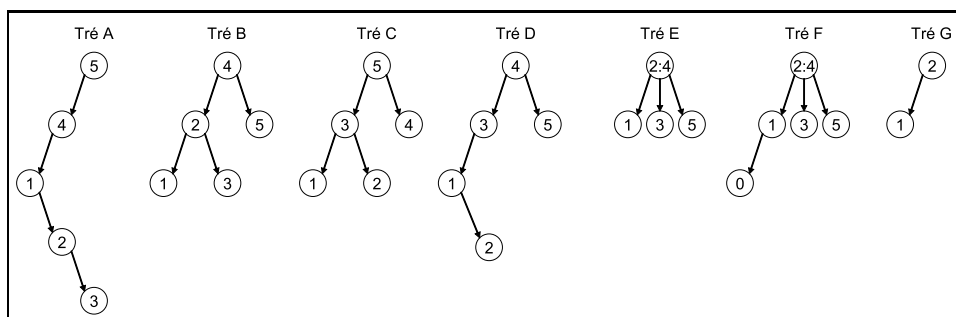
Svarið a.m.k. 3 dæmum úr þessum hluta

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 hrúgu gilda með stærsta gildi efst, sem geymd eru í fylki?
11. 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
12. Gerið grein fyrir muninum á tímaflækju AVL-trjáa, Splay-trjáa og skopplista (skip list). Hvert af þessum gagnamótum gefur besta flækjustigloforð fyrir einstakar aðgerðir? Hvers vegna er samt sem áður hugsanlegt að hin gagnamótin séu betri í einhverjum tilvikum?
13. Skriðið 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. Skrifðu stef sem skilar stærsta prímtölupætti tölu sem skal vera viðfang stefsins.
15. Skrifðu 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. Skrifðu stef sem finnur rót samfellds falls með helmingunarleit.
17. Skrifðu stef sem finnur **hágildi** samfellds falls með helmingunarleit.