

## TÖL203G Tölvunarfræði 2

Lausn fyrir miðmísserispróf 2012

Öll dæmin hafa sama vægi. **Aðeins þarf að leysa 7 dæmi af 10. Sjö bestu dæmin gilda.** Engin skrifleg hjálpargögn leyfileg.

1. Helmingunarleit. Fyllið inn þar sem spurningamerkin 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 vaxandi 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] >= x) // ?1?
        return leita(f,i,m,x); // ?2?
    else
        return leita(f,m+1,j,x); // ?3?
}
```

```
int leita(double[] f, int i, int j, double x) {
    int p = i, q = j;
    while (p != q) { // ?4?
        // | <x | óþekkt | >=x |
        // i   p       q     j
        int m = (p+q)/2; // ?5?

        if (f[m] < x) // ?6?
            p = m+1; // ?7?
        else
            q = m; // ?8?
    }
    return p;
}
```

2. Er eftirfarandi útfærsla á valröðun (selection sort) stöðug? Rökstyðjið eða sýnið mótdæmi.

```
// Notkun:  ssort(f,a,b);
// Fyrir:   f[i..j-1] er svæði í f
// Eftir:   Búið er að raða svæðinu f[i..j-1] í vaxandi röð.
//          Röðunaraðferðin er selection sort.
static void ssort(Comparable[] f, int i, int j)
{
    for (int p = i; p < j; p++)
    {
        // f[i..p-1] inniheldur minnstu gildi úr f[i..j-1]
        // í vaxandi röð
        int min = p;
        for (int k = p+1; k < j; k++)
        {
            // f[i..p-1] inniheldur minnstu gildi úr f[i..j-1]
            // í vaxandi röð
            // f[min] <= f[p..k-1]
            if (less(f[k], f[min]))
                min = k;
        }

        swap(f, p, min);
    }
}
```

**Lausn:** Nei, hún er ekki stöðug. Mótdæmi: grf. fylkinu {5,5,2} og við skulum merkja fimmurnar með bókstöfum til að þekkja þær í sundur: {5A,5B,2}. Í fyrstu umferð þá finnur innri lykkjan minnsta gildið í fylkinu og skiptir því út fyrir fyrsta gildið. Staðan eftir fyrstu lykkjuna er því: {2,5B,5A}. Í næstu lykkju þá finnur innri lykkjan minnsta gildið í fylkinu og skiptir því út fyrir annað gildið. Innri lykkjan uppfærir min breytuna bara ef það finnur minna gildi, ekki þegar það finnur jöfn gildi. min helst því óbreytt og staðan eftir aðra umferð er: {2,5B,5A}. Sama gerist í þriðju umferð og við endum með lokastöðuna: {2,5B,5A}. Ef röðunin væri stöðug þá væri 5A á undan 5B eftir röðun.

3. Hver er fastayrðing ytri lykkju í insertion sort? Hver er tímaflækja insertion sort í versta falli? En í besta falli?

**Lausn:** Fastayrðing ytri lykkju insertion sort er:

```
| í vaxandi röð | óþekkt |  
i                p        j
```

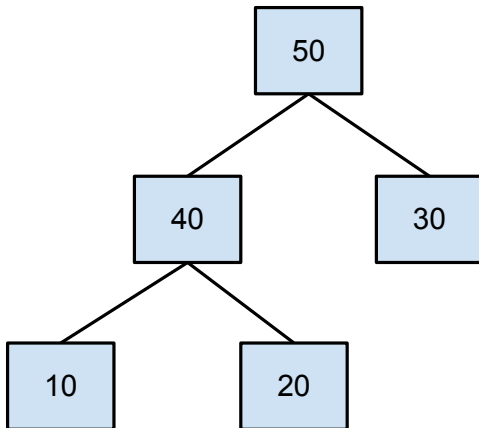
Besti tími er  $O(n)$  en versti er  $O(n^2)$

4. Gerið grein fyrir meginreglunni um upplýsingahuld: Hver er reglan og hver er tilgangur hennar?

**Lausn:** Meginreglan um upplýsingahuld segir að allar veigamiklar ákvörðanir eigi að fela í einingu. Tilgangurinn er að tryggja aðskilnað notanda og forritara. Þá getur forritari t.d. breytt ákvörðun um útfærslu án þess að það kosti stóran uppskurð á kerfinu.

5. Eftirfarandi runu af gildum er bætt í tóma hrúgu með stærsta gildi efst: 40 10 30 20 50. Teiknið mynd af hrúgunni eftir að öll gildin hafa verið sett í hana. Þið þurfið ekki að sýna stöðuna í hverju skrefi heldur bara lokastöðuna eftir að öll gildin hafa verið sett í hrúguna.

**Lausn:**

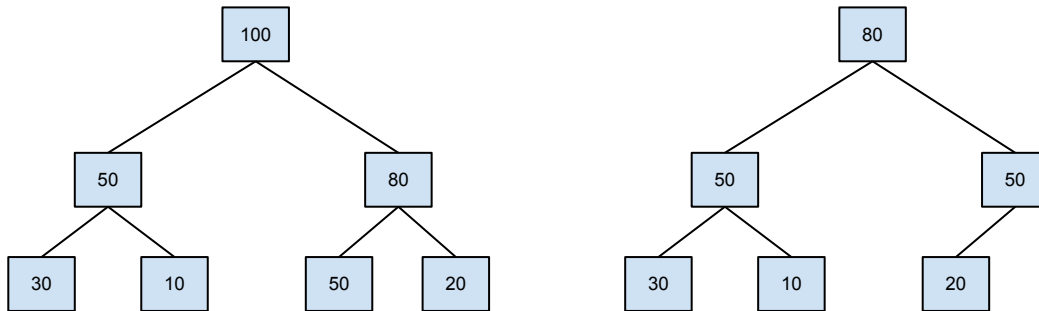


6. Gefið er eftirfarandi fylki sem inniheldur hrúgu með stærsta gildi efst:

```
int[] f = {100,50,80,30,10,50,20};
```

Teiknið mynd af hrúgunni eins og hún er í upphafi. Fjarlægjið síðan stærsta stakið úr henni og teiknið mynd af hrúgunni eftir það.

**Lausn:**



7. Lýsið hugmyndinni í mergesort. Hver er tímaflækja mergesort?

**Lausn:** Til að raða runu gilda með mergesort skiptum við rununni til helminga (eða því sem næst) í tvær runur, röðum þeim með mergesort og samröðum síðan útkomunum. Ef runa inniheldur færri en tvö gildi þarf auðvitað ekkert að gera. Tímaflækjan er  $O(n \log n)$

8. Skrifðu fastayrðingu gagna fyrir útfærslu á hlaða og skrifuðu notkunarlýsingu ásamt útfærslu fyrir Push aðferðina.

**Lausn:**

```
// Gildin á hlaðanum eru geymd í keðjunni
// top frá efsta til neðsta.
private Node top;

// Notkun: s.Push(x);
// Fyrir: Ekkert
// Eftir: Búið er að setja x efst á hlaða s
public void Push(T x)
{
    Node p = new Node();
    p.value = x;
    p.next = top;
    top = p;
}
```

9. Skriðu fall í Java sem fær sem inntak bendi á haus í eintengdum lista heiltalna og skrifar út allar tölurnar.

Skilgreining á hlekk:

```
static class Node
{
    int value;
    Node next;
};
```

**Lausn:**

```
static void print(Node head)
{
    Node p = head;

    while (p != null)
    {
        System.out.println(p.value);
        p = p.next;
    }
}
```

10. Gerum ráð fyrir eftirfarandi fylki af heiltölum:

```
int[] f = {5, 3, 8, 1, 2, 6, 4, 9, 7}
```

Gerum síðan ráð fyrir að fylkinu sé skipt með quicksort partition falli þar sem  $f[0] == 5$  er notað sem vendistak (e. pivot).

Hvað gildir um fylkið eftir skiptinguna?

**Lausn:** Vendistakið er komið í sitt endanlega sæti miðað við rétta röð, öll minni gildi eru fyrir neðan það í fylkinu og öll stærri eru fyrir ofan það.

```
| <= 5 | 5 | >= 5 |
0      4      8
```