

Tölvunarfræði 2

Skyndipróf

23. febrúar 2010

Allar spurningar gilda 16.7%. Bestu 6 svör gilda til einkunnar. Engin hjálpargögn eru leyfileg. Merkið öll svarblöð með **nafni** og **númeri dæmahóps** (D1, D2 eða D3).

1. 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;
}
```

2. Fyllið inn þar sem spurningarmerkin eru. Þetta er selection sort.

```

// Notkun: swap(f,i,j);
// Fyrir: f[i] og f[j] eru sæti í f
// Eftir: Búið er að víxla gildunum í f[i] og
//        f[j]
void swap( double[] f, int i, int j );

// Notkun: sort(f,i,j);
// Fyrir: f[i..j-1] er ekki-tómt svæði í f
// Eftir: Búið er að raða f[i..j-1]
void sort( double[] f, int i, int j ) {
    int p = ?1?;
    while( ?2? ) {
        // | minnstu gildi í vaxandi röð | óþekkt |
        // i                                p                j
        int q = ?3?;
        while( ?4? ) {
            // a) | minnstu gildi í vax. röð | óþekkt |
            //      i                                p                j
            // b) p < q <= j
            // c) f[p] er minnst af f[p..q-1]
            if( f[q] ?5? f[p] )
                swap(f,p,q);
            q++;
        }
        p++;
    }
}

```

3. Hver er fastayrðing lykkju í ytri lykkju insertion sort?

4. Hver eftirfarandi getur verið fastayrðing seinni lykkju heapsort?

(a) | minnst í vaxandi röð | hrúga (heap) |
0 i n

(b) | minnst í vax. röð | uppfyllir hrúguskilyrði |
0 i n

(c) | hrúga (heap) | stærst í vaxandi röð |
0 i n

5. (a) Hvaða fastayrðingu má nota í fyrri lykkju heapsort?

(b) Hver er tímaflækja heapsort?

6. (a) Lýsið hugmyndinni í merge-sort

(b) Hver er tímaflækja merge-sort?

7. Gefið er eftirfarandi C++ stef:

```

// Notkun:   skipta(f, i, j, k, n);
// Fyrir:   f[i..j-1] er ekki-tómt svæði í fylkinu f
// Eftir:   Búið er að víxla gildum í svæðinu og gefa
//          k og n gildi þ.a.
//          1)  $i \leq k < n \leq j$  og
//          2)  $f[i..k-1] < p$  og
//          3)  $f[k..n-1] = p$  og
//          4)  $f[n..j-1] > p$ 
//          fyrir eitthvert  $p$  sem fyrir var í svæðinu.
int skipta( double f[], int i, int j, int& k, int& n );

```

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

8. Hver er tímaflækja eftirfarandi röðunaraðferða? Hvers konar tímaflækju er um að ræða (versti tími, meðaltími, o.s.frv.)?
 - (a) Insertion-sort
 - (b) Selection-sort
 - (c) Quicksort
 - (d) Merge-sort
 - (e) Radix-sort (gerið ráð fyrir þriggja stafa tölum)
 - (f) Heapsort

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

10. Bubblesort er vel þekkt röðunaraðferð sem virkar þannig að til að raða fylki $f[0..n-1]$ endurtökun við aftur og aftur sömu lykkju þar sem rennt er eftir fylkinu frá vinstri til hægri (líka hægt að fara frá hægri til vinstri) og víxlum gildum í samliggjandi sætum ef gildið í vinstra sæti er stærra en gildið í hægri sæti. Að lokum kemur að því í einhverri umferð ytri lykkju að engar víxlanir þarf í innri lykkju, og þá er röðun lokið.
 - (a) Hver er tímaflækja bubblesort í versta tilfelli?
 - (b) Hver er tímaflækja bubblesort í besta tilfelli?
 - (c) (Aukaspurning, þarf ekki að svara) Hver er tímaflækja bubblesort að meðaltali? Hvaða nýjar forsendur þarf að gefa sér til að svara þeirri spurningu?

11. Gerið ráð fyrir að til sé skilgreining í C++ eða Java á grunnklasa eða skilum (interface), IntBag, fyrir poka heiltalna, með boðum get, put og isEmpty með venjulegum lýsingum fyrir poka.
 - (a) Hverja af eftirfarandi klösum er þá eðlilegt að útfæra sem undirklasa af IntBag:
 - Klasi, IntStack, fyrir hlaða heiltalna.
 - Klasi, IntQueue, fyrir biðröð heiltalna.

- Klasi, IntPriQueue, fyrir forgangsbiðröð heiltalna.
 - Klasi, IntSet, fyrir mengi heiltalna.
- (b) Fyrir þá klasa úr upptalningunni sem ekki er eðlilegt að útfæra sem undirklasa IntBag, rökstyðjið að það sé ekki eðlilegt með því að sýna að eðlilegar lýsingar á einhverju boði geti ekki uppfyllt þær söksemdafærslureglur sem til þarf.
12. 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 notkunarlýsingu (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?