

08.71.23/24 — Tölvunarfræði 2/2a

Vorpróf 30. apríl 2008.

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.**This is a closed-book exam.*

All questions count equally.

Remember to write a usage description with precondition and postcondition for all functions and a data invariant for all implementations of data structures.

Athugið vel: Svára þarf tilskildum fjölda dæma úr hverjum hluta prófsins. Að því skilyrði uppfylltu gilda **10 bestu dæmi** til einkunnar. Þið hafið því 18 mínútur fyrir hvert dæmi að meðaltali. Byrjið því á að svara dæmum sem krefjast stuttra svara og þið getið auðveldlega svarað.

Note carefully: You need to answer the required number of questions from each part of the exam. Fulfilling that condition, the **10 best answers** count for your grade. You therefore have 18 minutes for each question on the average. You should therefore start by answering questions that require short answers and you can easily answer.

Hluti I: Röksemdafærsla

Svarið a.m.k. 1 dæmi úr þessum hluta

Part I: Programming and verification

Answer at least one question from this part.

1. Íhugið eftirfarandi lykkjumynstur:
Consider the following loop pattern:

```
// 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 lykka þessi sé rökrétt?

What relationship do we need between F, C1, I1, I2, S1, C2, S2 and E in order for this loop to be logically valid?

2. Hvaða lágmarkssamband þarf að gilda milli lýsinga boða í undirklasa og yfirklasa til að tryggja sé að forrit sem nota þessa klasa séu rökrétt (miðað við að röksemdafærsla sé að öðru leyti rétt)?

What minimum relationship needs to hold between descriptions of messages (methods) in a subclass and a superclass to ensure that programs using these classes are logically valid (assuming that the logic is otherwise valid)?

3. Íhugið eftirfarandi lykkju, sem hefur þann tilgang að reikna x^y (x^y) þar sem x er flýttala og y er jákvæð heiltala:

Consider the following loop, whose purpose is to compute x^y (x^y) where x is a floating point number and y is a positive integer:

```
// y >= 0
double p=1.0, q=x;
int r=y;
while( r != 0 ) {
    // x^y == p*q^r, r>=0
    ... (breytum hvorki x né y)
}
// ???
```

- Hvert er rökrétt eftirskilyrði lykkjunnar?
What is the logical postcondition of the loop?
- Hvernig getum við forritað stofninn í lykkjuna á hraðvirkan hátt þannig að fjöldi umferða verði $O(\log y)$?
How can we program the body of the loop so that the loop is fast and the number of passes through the loop is $O(\log y)$?

Hluti II: Algrím

Svarið a.m.k. 3 dæmum úr þessum hluta

Part II: Algorithms

Answer at least 3 questions from this part

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.

Fill in for the question marks, i.e. write what should replace ?1?, etc. This is a total of seven answers.

```
// 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.

// Usage: k = leita(f,i,j,x);
// Pre:   f[i..j-1] is in ascending order and
//        contains at least one value greater
//        than x.
// Post:  k is an index to the first value in
//        f[i..j-1] which is greater than x.

int leita( double[] f, int i, int j, double x ) {
    int p=?1?, q=?2?;
    while( ?3? ) {
        // | <=x | óþekkt/unknown | >x |
        // i     p                     q     j
        int m = (?4?)/2;
        if( f[m] ?5? x )
            p = ?6?;
        else
            q = ?7?;
    }
    return p;
}
```

5. Gefið er eftirfarandi stef:

The following function is given:

```
// Notkun: rolldown(f,i,j);
// Fyrir:  Svæðið f[i+1..j] uppfyllir hrúguskilyrði
//        fyrir hrúgur með stærsta gildi efst.
// Eftir:  Búið er að víxla gildum í svæðinu f[i..j]
//        þannig að það svæði uppfyllir nú
//        hrúguskilyrðið.

// Usage:  rolldown(f,i,j);
```

```
// Fyrir:   The area f[i+1..j] fulfills a heap
//         condition for heaps with greatest value
//         on top.
// Eftir:   The values in the area f[i..j] have been
//         permuted in such a way that this area now
//         fulfills the heap condition.
```

```
void rolldown( double[] f, int i, int j );
```

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

Write a heapsort function (with description - usage, pre, post) using this function. You do not need to program the rolldown function.

Ef þið viljið megið þið gera ráð fyrir að til sé rollup stef, svipað og rolldown, nema í hina áttina, en þið verðið þá að lýsa því stafi (notkun/fyrir/eftir). Þið þurfið ekki að forrita það stef.

If you wish, you may assume the existence of a rollup function, similar to the rolldown function except in the other direction, but you must then describe that function (usage/pre/post). You do not need to program that function.

Athugið: Við reiknum með því að rót hrúgunnar sé $f[0]$.

Note: We assume that the root of the heap is $f[0]$.

6. Tilgreinið tímaflækju eftirfarandi röðunaraðferða miðað við að raðað sé n slembitölum. Tilgreinið einnig hvort tímaflækjan sé tími í versta tilfelli, meðaltími eða eitthvað annað.

Specify the time complexity of the following sorting methods assuming that we are sorting n random numbers. Also say whether the time complexity is worst case complexity, average complexity or something else.

- (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
Sort by shovelling the numbers into an AVL tree (which is originally empty) and then shovelling them out in ascending order

- (g) Röðum með því að moka gildunum í Splay tré (sem upphaflega er tóm) og moka þeim síðan út í vaxandi röð
Sort by shovelling the numbers into a Splay tree (which is originally empty) and then shovelling them out in ascending order
7. Lýsið quicksort. Tilgreinið tímaflækju quicksort og rökstyðjið hana.
Describe quicksort. Specify the time complexity of quicksort and give reasons to deduce that this is the complexity.
8. Hver er fastayrðing ytri lykkju í insertion-sort? Tilgreinið tímaflækju insertion-sort í versta tilfalli og rökstyðjið hana.
What is the loop invariant in the outer loop of insertion sort? Specify the worst case time complexity of insertion sort and give reasons why this is the complexity.

Hluti III: Gagnamót

Svarið a.m.k. 3 dæmum úr þessum hluta

Part III: Data Structures

Answer at least 3 questions from this part.

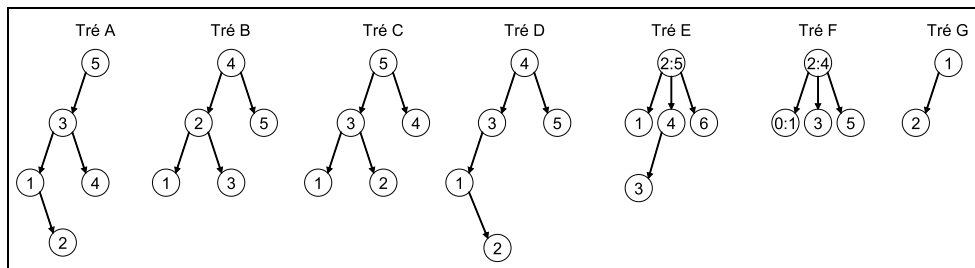
9. Íhugið eftirfarandi myndir af trjám. Segið til fyrir hverja mynd hvort hún getur staðið fyrir eitt eða fleiri af eftirfarandi:
- Tvíleitartré – Binary search tree
 - AVL-tré – AVL tree
 - Splay-tré – Splay Tree
 - Rautt-svart tré (ef svo er, tilgreinið þá einnig hvaða hnúta má mála rauða til að það gangi upp)
Red-black tree (if so, specify which nodes you could paint red in order for this to be true).
 - 2-3 tré – 2-3 tree
 - Hrúga með hæsta gildi efst – Heap with greatest value on top
 - 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.

Note that we want all binary search trees to be in ascending in-order order.

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

Note also that the same picture of a tree can easily stand for more than one of these listed types of trees.



10. Hver er fastayrðing gagna fyrir rauð-svört tré?
What is the data invariant (representation invariant) for red-black trees?
11. Í tætitöflum og í fleiri vandamálum eru notuð stækkanleg fylki sem hafa sömu aðgerðir og venjuleg fylki en auk þess aðgerð til að bæta einu sæti aftast á fylkið. Hver er tímaflækja þessarar aðgerðar og hvers konar tímaflækja er það (versta tilfelli, meðaltal eða innistæðubungin)? Lýsið því hvernig aðgerðin er útfærð til að ná þessari tímaflækju.
In hash tables and in other tasks we use extendible arrays that have the same operations as ordinary arrays but also have an operation for appending one position to the end of the array. What is the time complexity of this operation and what kind of time complexity are we talking about (worst case, average or amortized)? Describe how to implement this operation in order to achieve this time complexity.
12. Gerið grein fyrir einni eða fleiri aðgerðum sem AVL-tré gera kleift að gera á hraðvirkari hátt en tætitöflur gera kleift. Gerið einnig grein fyrir einni eða fleiri aðgerðum sem tætitöflur gera kleift að gera á hraðvirkari hátt en AVL tré.
Describe one or more operations that AVL-trees support with better time complexity than hashtables support. Also describe one or more operations that hash tables support with better time complexity than AVL trees.
13. Skrifid klasa í Java eða C++ fyrir forgangsbiðröð heiltalna. Þið megið sleppa því að forrita boðin, nema fyrir aðferðina til að bæta gildi í forgangsbiðröðina, en munið að hafa notkun, forskilyrði og eftirsilyrði fyrir öll boð og munið að hafa skýra fastayrðingu gagna.
Write a class in Java or C++ for priority queues of integers. You do not need to program the methods, except for the method for adding a value to the priority queue, but remember to have usage, pre and post for all messages (methods), and remember to have a clear data invariant.

Hluti IV: Blandað efni

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

Part IV: Miscellaneous

You do not need to answer any question from this part, but do not forget to answer a total of 10 questions.

14. Skrifðu stef sem skilar minnsta prímtölubátt tölu sem skal vera viðfang stefins.
Write a function that returns the smallest prime factor of a number that should be an argument to the function.
15. Skrifðu forrit sem les strengi frá aðalinntaki og skrifar strengina í minnkandi röð á aðalúttak.
Write a program that reads strings from standard input and writes the strings in descending order to standard output.
16. Skrifðu stef sem finnur rót samfellds falls með helmingunarleit.
Write a function that finds the root of a continuous function using binary search.
17. Skrifðu stef sem finnur lággildi samfellds falls með helmingunarleit.
Write a function that finds the minimum of a continuous function using binary search.