

# 08.71.23/24 Tölvunarfræði 2/2a

## Lokapróf

Kennari: Hjálmtýr Hafsteinnsson

5. maí, 2003

kl. 9<sup>00</sup> ? 12<sup>00</sup>

Fyrstu 5 dæmin eru fyrir alla nemendur (bæði í Tölvunarfræði 2 og 2a) Dæmi 6 er aðeins fyrir nemendur í Tölvunarfræði 2, en dæmi 7 er aðeins fyrir nemendur í Tölvunarfræði 2a (verkfræðinema). Í báðum tilfellum gilda **fimm bestu dæmin af sex**. Dæmin hafa jafnt vægi.

**Öll skrifleg hjálpargögn og reiknivél leyfð.**

- ?? Athugið að þegar beðið er um að "Lýsa" eða "Sýna" þá er nóg að gera það í orðum og með teikningum. Ef þið eigið að skrifa C++ kóða þá er beðið um það sérstaklega.  
?? Rökstyðjið öll svör og munið að það er óþarfi að skrifa upp skilgreiningar sem eru í bókinni.

1. Gætu eftirfarandi  $d$ -vektorar hafa komið upp í lausn Union-Find verkefnisins ef notað er reikniritið *Quick-union* (Forrit 1.2 á bls. 15)? Ef vektor er ómögulegur, útskýrið þá hvers vegna, en sýnið annars röð para sem myndi leiða til hans.

a)

0	1	2	3	4	5	6	7
2	3	6	2	4	2	0	1

b)

0	1	2	3	4	5	6	7
0	4	0	6	4	5	0	0

c)

0	1	2	3	4	5	6	7
5	5	5	5	5	5	5	5

2. Útfærið í C++ fallið

```
void removeGT(node *&h, int val)
```

sem tekur inn tilvísun á hnútabendinn  $h$ , sem bendir á eintengdan lista án haushnúts. Fallið fær líka inn heiltöluna  $val$  og það á að taka út úr listanum öll stök með gildi sem eru stærri en  $val$ .

Munið að afgreiða öll sértílfelli í fallinu (tómur listi, o.s.frv.).

3. Skilgreind er ný tegund biðraðar með aðferðunum `insert` og `gettop10`. Aðferðin `gettop10` tekur út úr biðröðinni, og skilar, einu af fremstu 10% af stökunum í biðröðinni ("fremsta" þýðir hér "fyrst sett inn"). Stakið sem tekið er út er valið af handahófi (með slembitölugjafa) af þessum fremstu 10% stökum.

Lýsið í orðum og teikningum útfærslu á þessari gerð biðraðar og greinið tíma (með stærðargráðureikningi) aðferðanna tveggja. Lýsingin þarf að vera nógu nákvæm til þess að auðvelt sé að forrita útfærsluna, en þið þurfið ekki að sýna neinn kóða.

4. a) Hvar koma eins gildi í *hrúgu* (e. heap), t.d. hvar myndu öll stök með gildið *k* vera í hrúgu? Útskýrið það í orðum og teikningum.

b) Hvernig væri hægt að telja öll stök með gildið *k* í hrúgu án þess að skoða endilega öll stökin í hrúgunni (nema þegar það er nauðsynlegt). Lýsið reikniritinu í orðum.

5. Skrifðið í C++ endurkvæma fallið

```
int countEQ( node *h, int k)
```

sem skilar fjölda staka í **tvíleitarrénu** (e. binary search tree) *h* sem hafa lykilinn *k* (t.d. "`t->item.key() == k`"). Skrifðið tvær útgáfur af fallinu:

a) miðað við að innsetning setji eins lykla í hæggra hluttré (þannig er innsetningarfallið á bls. 517 í kennslubókinni).

b) miðað við að eins lykklar geti farið í bæði hæggra og vinstra hluttré.

Í báðum tilfellum á fallið ykkar að fara aðeins í þá hnúta sem nauðsynlegt er að fara í, til þess að telja hnúta með lykilinn *k*. Það má sem sagt ekki bara fara í gegnum alla hnúta trésins og telja saman alla sem hafa lykilinn *k*.

### *Aðeins fyrir nemendur í Tölvunarfræði 2:*

6. Í Innsetningarröðuninni sem gefin er í bókinni (bls. 276) er notað *varðstak* (e. sentinel) til að einfalda skilyrði í `while`-lykkju. Hins vegar er ekki notað varðstak í Innsetningar-röðuninni sem Shell-röðunin á bls. 286 notar.

a) Útskýrið nákvæmlega í orðum hvað þarf að gera til að breyta Shell-röðuninni yfir í að nota Innsetningarröðun með varðstaki.

b) Breytið Shell-röðunarforritinu á bls. 286 þannig að það noti varðstak (sbr. forritið á bls. 276).

### *Aðeins fyrir nemendur í Tölvunarfræði 2a (verkfræðinema):*

7. Í eintengdum, hringtengdum lista með haushnúti er annað hvort hægt að hafa haushnúttinn sem hluta af hringnum, eða fyrir utan hringinn.

a) Lýsið þessum tveimur útfærslum með teikningum.

b) Hvernig væri tómur listi táknaður í útfærslunum tveimur?

c) Hvor aðferðin væri betri við eyðingu hnútar með tiltekið gildi, og hvers vegna?