

08.71.23/24 Tölvunarfræði 2/2a

Upptökupróf

Kennari: Hjálmtýr Hafsteinsson

25. ágúst, 2004

kl. 13³⁰ – 16³⁰

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. Skrifid fall í C++ sem fær inn bendi p á eintengdan hringtengdan lista og heiltöluna k . Fallið á að taka út alla hnúta listans með gildi sem er lægra en k og búa til nýjan hringtengdan lista úr þeim. Fallið á síðan að skila bendi á þann lista. Haus fallsins er

```
node* SplitOnNumber(node* &p, int k)
```

Athugið að viðfangið p er tilvísun á bendi, því p gæti þurft að breytast.

2. Ein tegund biðraðar leyfir bæði innsetningu og eyðingu staka frá báðum endum raðarinnar. Aðgerðin `putFront` setur inn stak fremst í biðröðina (munið að `put` setur stak inn aftast) og aðgerðin `getBack` tekur út aftasta stakið (munið að `get` tekur út fremsta stakið). Bætið þessum tveimur nýju aðgerðum við skilgreiningu biðraðar og sýnið útfærslu á þeim í C++ **bæði** fyrir vektor-útgáfuna **og** tengda lista-útgáfuna.

3. Hnútur í tvíundartré er sagður í *ójafnvægi* ef það munar tveimur eða meira á hæð barna hans. Fjöldi slíkra hnúta í tvíundartré gæti verið notaður til að meta hversu gott jafnvægi er á trénu. Skrifid endurkvæmt fall í C++ sem telur fjölda hnúta sem eru í ójafnvægi í gefnu tvíundartré.

4. a) Er niðurstaðan úr Alpha-Beta leit alltaf sú sama og Minimax leit í leikjatré? Rökstyðjið eða sýnið mótdæmi.
- b) Skiptir einhverju máli í Minimax leit, í hvaða röð við skoðum börn hnútar? En í Alpha-Beta leit? Útskýrið.
- c) Gefið er leikjatré af hæð 5 þar sem hver innri hnútur hefur 10 börn. Ef Alpha-Beta leit klippir burtu 30% hnúta í hverju lagi (miðað við Minimax), hversu marga hnúta trésins skoðar Alpha-Beta leit miðað við hversu marga Minimax skoðar?

5. Útfærið í C++ nýja útgáfu af hrúgufallinu `getmax`. Í því er gatinu sem myndast í rótinni ýtt niður í lauf trésins með því að færa sífellt upp stærra barnið. Síðan er aftasta stakið (þ.e. stakið í `pq[N]`) sett í gatið og því víxlað uppá við þar til hrúgueiginleikinn er uppfylltur.

Berið þessa aðferð saman við `getmax` fallið á bls. 386 í kennslubókinni. Hvor aðferðin teljið þið að sé hraðvirkari og hvers vegna?

Aðeins fyrir nemendur í Tölvunarfræði 2:

6. Ein leið til að velja vendistakið (e. partitioning element) í Quicksort er að velja það af handahófi. Þá er kallað á slemitölugjafa til að velja tölu k á bilinu l til r og stakið í sæti $a[k]$ er notað sem vendistak. Vandamálið er að góðir slembitölugjafar gera verið tímafrekir. Reiknið út hversu oft yrði kallað á slembitölugjafann í Quicksort í versta tilfalli. En í besta tilfalli?

Aðeins fyrir nemendur í Tölvunarfræði 2a (verkfræðinema):

7. Setja á öll stök hrúgu (e. heap) yfir í tvíleitartré (e. binary search tree). Hrúgan er gefin sem N -staka vektor og þið getið aðeins notað aðgerðina `insert` á tvíleitartréð. *i)* Myndi það borga sig að setja inn stökin í vektornum í röð frá byrjun hans, þ.e. fyrst `pq[1]`, síðan `pq[2]`, o.s.frv.? *ii)* Hvað með frá hinum endanum (þ.e. byrja á `pq[N]`)? *iii)* Er einhver önnur röð staka úr hrúguvektornum sem væri betri en þær tvær ofangreindu? Rökstyðjið svör ykkar.