

TÖL203G Tölvunarfræði 2

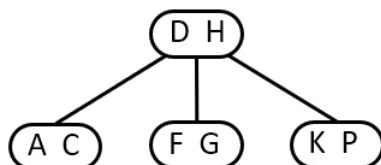
Heimadæmi 7

Í síðustu viku var farið í kafla 3.3 um tvíleitartré í jafnvægi, þar sem aðal gagnagrindurnar voru 2-3 tré og vinstri-hallandi rauð-svart tré. Í næstu viku verða teknar fyrir hakkatöflur úr kafla 3.4 og ýmsir notkunarmöguleikar á táknatöflum úr kafla 3.5.

Heimadæmin eru **til þess að þjálfar ykkur** í efninu – nýtið þau vel! Einkunn fyrir þau mun **ekki lækka lokaeinkunn**, þannig að þið fáið mun meira út úr því að glíma við dæmin sjálf en að fá aðstoð frá Hr. Google.

Æfingadæmi fyrir dæmatíma 6. og 7. mars

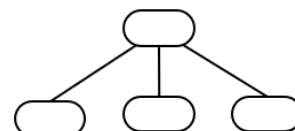
1. Breytið 2-3 trénu hér fyrir neðan í jafngilt vinstri-hallandi rauð-svart tré.



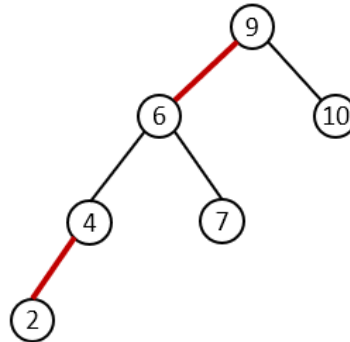
2. Setjið lykilinn **S** inn í 2-3 tréð í dæminu að ofan og sýnið lokatréð.

Heimadæmi (skila í Gradescope)

1. [2-3 tré]
 - a. Hversu mörg ólík 2-3 tré eru til sem innihalda **5** lykla? Rissið upp myndir af öllum mögulegum trjám.
 - b. Hversu mörg ólík 2-3 tré eru til sem innihalda **6** lykla? Rissið upp myndir af öllum mögulegum trjám. Athugið að röð barna skiptir máli.
2. [2-3 tré] Setjið lyklana 1, 2, 3, 4, 5, 6, 7 og 8 inn í 2-3 tré í þeirri röð sem gefur 2-3 tré sem hefur alla hnútana sem 3-hnúta. Lögunin er þá eins og sést á myndinni hér til hægri. Gefið upp röð lyklanna og sýnið mynd af trénu.



3. [Rauð-svart tré] [Próf '22] Sýnið hvert skref þegar stakið 5 er sett inn í eftirfarandi vinstri-hallandi rauð-svart tvíleitartré.



4. [Rauð-svart tré] Setjið lyklana 10, 9, 8, 7, 6, 5, 4 í þessari röð inn í vinstri-hallandi rauð-svart tré. Sýnið tréð eftir hverja innsetningu.
5. [Rauð-svart tré] Smíðið forritið `CompareBST.java`, sem er eins konar framhald af dæmi 5 á Heimadæmum 6. Hér á að bera saman venjuleg tvíleitartré ([BST.java](#)) og vinstri-hallandi rauð-svart tré ([RedBlackBST.java](#)). Forritið fær inn `n` (fjölda lykla) og `trials` (fjölda keyrsla) og setur `n` slembilykla inn í venjulegt tvíleitartré annars vegar og rauð-svart tré hins vegar. Það prentar svo út meðalhæð trjáanna í hvoru tilfelli og tímann sem innsetningin tók í hvoru tilfelli.

Bætið svo líka við möguleikanum að setja inn lyklana í röð. Það er til dæmis hægt að setja inn gildin 0.0, 1.0, 2.0, ... sem lykla. Þið getið gert það á svipaðan hátt og gert var í forritinu [SortCompare.java](#), þ.e. með því að láta þriðja skipanalínuviðfangið vera textann "sorted". Athugið að í því tilviki má `n` ekki vera mjög stór tala, því annars fyllist forritshlaðinn og forritið hættir keyrslu.

Svarið svo eftirfarandi spurningum (með því að keyra forritið):

- Hvor aðferðin gefur lægra tré að meðaltali fyrir slembilykla?
- Hvor aðferðin er hraðvirkari í innsetningu á slembilyklum (keyrið með `n = 500000` og `trials = 10`)?
- Hver er tími og meðaltréhæð á venjulegt tvíleitartré á röðuðu inntaki (keyrið með `n = 1000` og `trials = 1000`)?
- Hvernig hegðar rauð-svart tré sér á röðuðu inntaki?

Skilið PDF-skjali með lausnum ykkar á þessum dæmum fyrir **kl. 23:59 fimmtudaginn 9. mars** í [Gradescope](#). Munið eftir að gefa upp á hvaða blaðsíðum svör við einstökum dæmum eru.