

# Tölvunarfræði 1

## Heimadæmi 10

Í elleftu viku námskeiðsins (4.-6. nóv.) verður farið í notkun á flóknari gagnatögum í Java. Skoðuð verða sérstaklega tögin `String` og `Color`. Þetta er efni úr [kafla 3.1](#) í kennslubókinni.

Hér fyrir neðan eru heimadæmi 10, sem eru að mestu um síun úr kafla 2.4 í kennslubókinni. Þið eigið að skila þessum dæmum **útprentuðum** merktum með **nafni** ykkar, **dæmahópi** og **nafni dæmatímakennara**. Dæmatímakennararnir hafa hólf í anddyri VR-II og þið eigið að skila lausnum ykkar í þau hólfi fyrir kl. 13:00 föstudaginn 6. nóvember. Ekki verður tekið við dæmum sem koma eftir það.

### Æfingadæmi úr kennslubók:

1.4.12, 2.1.15, 2.3.6, 2.4.1, 2.4.3

### Heimadæmi

1. [*Gamalt prófdæmi*] Skrifa á Java forrit sem les fyrst inn af lyklaborði jákvæða heiltölu  $n$ , og síðan  $n$  heiltölur á bilinu frá 1 til 99. Fyrir hverja þessara inlesinna talna er búinn til stafastrengur þannig að fyrst koma jafn margir **a**-stafir og tugastafur tölunnar segir til um og síðan jafnmargir **b**-stafir og einingarstafur tölunnar segir til um. Þannig svarat t.d. talan 23 til strengsins **aabbb**, talan 40 til **aaaa** og talan 5 til **bbbbb**. Þessa strengi skal loks prenta út á skjá, línu fyrir línu, eftir að allar tölur hafa verið lesnar inn.
2. Skriðið Java forrit sem nota `StdDraw` til að teikna upp íslenska fánann. Það er góð lýsing á honum á [Wikipedia](#). Fáninn ykkar þarf að vera í réttum hlutföllum eins og lýst er á síðunni. Það er þægilegast að kvarða ásanna fyrst með `setXscale` og `setYscale`, þá verða allir hnitaútreikningar auðveldari. Skilið forriti og skjámynd af fánanum.
3. [*Gamalt prófdæmi*] Kallið `BuggySum(a, a.length-1)` ætti að reikna summuna af tölunum í fylkinu `a`, en gerir það ekki. Finnið villuna, útskýrið af hverju hún kemur fyrir og sýnið hvernig má laga fallið þannig að það virki rétt.



```
public static double BuggySum(double[] a, int k) {  
    if (k == 0) return 0.0;  
    else return a[k] + BuggySum(a, k-1);  
}
```

4. Dæmi 2.4.2 á bls. 307 í kennslubók. Skriðið líka útgáfu af [Visualize.java](#), sem nota þetta `print`-fall í stað fallsins `show` til að birta útkomuna. Skilið kóðanum fyrir fallið `print`, fyrir breyttu útgáfuna af `Visualize.java` og úttak úr einni keyrslu.

5. Þriðja útgáfa síunar (*percolation*) sem segja má að liggi á milli lóðréttrar síunar og almennrar síunar er bein síun (*directed percolation*). Hún er svipuð almennri síun, nema að ekki er hægt að fara **upp** í henni, bara niður og til hliðar. Útfærsla hennar er einföld, við þurfum bara að sleppa síðasta endurkvæma kallinu í **flow**-fallinu í forritinu [Percolate.java](#).
- Útfærið beina síun og gerið tilraunir með líkindastuðulinn  $p$  til að finna líkindadreifingu þess að slembifylki af stærðinni  $20 \times 20$  síist. Prófið a.m.k. fimm mismunandi gildi á  $p$  í 100.000 hermunum. Sýnið þann kóða sem þið skrifið og niðurstöðurnar.
  - Útfærið nú útvíkkaða beina síun (*expanded directed percolation*), þar sem flæðið getur nú líka farið **á ská** niður til vinstri og niður til hægri. Það eru því fimm möguleg hólf sem þarf að kanna. Skrifið forrit fyrir þessa útgáfu og gerið samskonar tilraunir og í a)-lið. Skilið ykkar kóða, niðurstöðum og smátexa með samanburði á þessum niðurstöðum við útkomuna úr a)-lið.