1. Kynning

Hjálmtýr Hafsteinsson
Vor 2022

Í þessum fyrirlestri

- Skipulag námskeiðs
  - Námsefni
  - Námsmat
  - Kennslufræði

- Dæmi um reiknirit fyrir heiltölumargföldun
  - Grindaraðferð (*Lattice*)
  - Skólaaðferð
  - Bændaaðferð (*Peasant*)

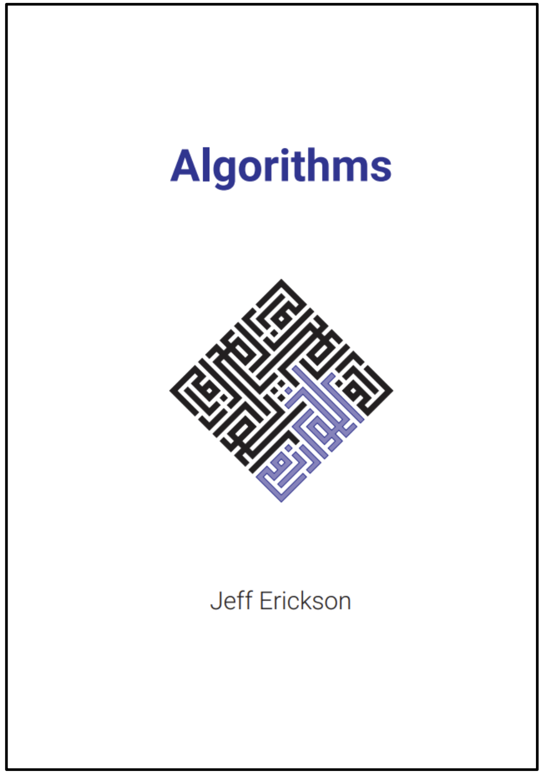
0.1 − 0.2, 0.5 − 0.6

Kennslubók

- Algorithms eftir Jeff Erickson
  - Frí kennslubók á Netinu
  - Kennd við Háskólann í Illinois í Urbana-Champaign

- Höfum notað þessa bók síðustu 4 ár
  - Þar áður var CLRS notuð

- Bókin er hönnuð fyrir byrjunarnámskeið í reikniritum, en hefur ýmislegt aukaefni sem við notum



Námsefni námskeiðs (sjá námsáætlun)

- Kynning, deila-og-drottna [0 og 1] — Vikur 1-2
- Endurrakning (*backtracking*) [2] — Vika 3
- Kvik bestun (*dynamic programming*) [3] — Vikur 4-5
- Gráðug reiknirit (*greedy alg.*) [4] — Vika 6
- Slembin reiknirit (*randomized alg.*) [DC 2 og DC 3] — Vika 7
- Strengleit [DC 7] — Vika 8
- Jafnaðargreining (*amortized analysis*) [DC 9, DC 10, DC 11] — Vikur 9-10
- Netareiknirit, hámarksflæði [5, 6, 10] — Vikur 11-12
- NP hard verkefni [12] — Vikur 13-14

Námsmat

▪ Vikuleg heimadæmi (bestu 8 gilda af 10-11)	25%
▪ Tvö (forritunar)verkefni (gilda 7,5% hvort)	15%
▪ Fyrirlestraæfingar (þarf 20 af 27)	10%
▪ Lokapróf	50-100%

- Heimadæmi, verkefni og æfingar eru til hækkunar
- **Verðið að ná lokaprófi** til að aðrar einkunnir gildi

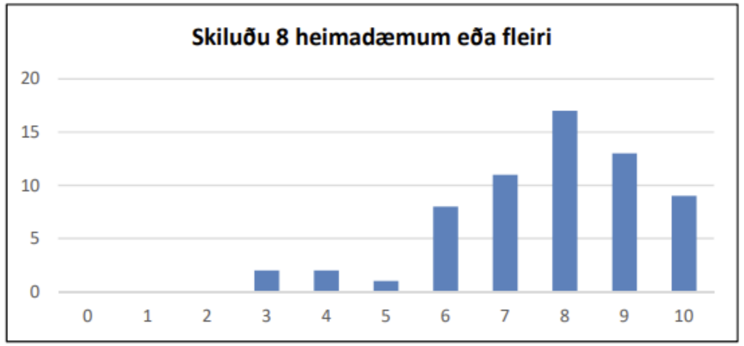
Tilgangur: Ekki "refsa" fyrir æfingar

Nýtið ykkur æfingarnar vel!

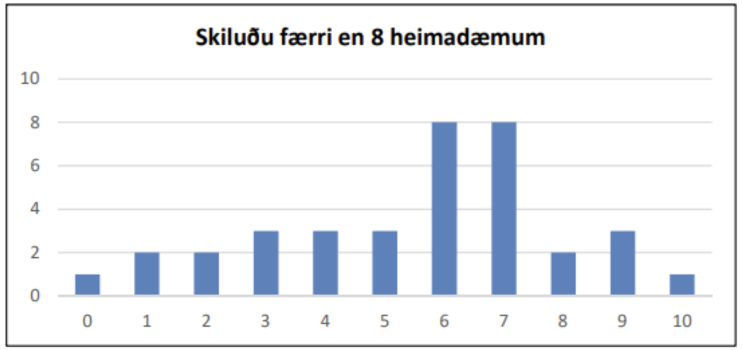
Heimadæmi

- Gríðarlega mikilvægt að gera heimadæmin
  - Vorið 2019 munaði rúmlega 2 heilum á lokaeinkunn hjá þeim sem skiluðu 8 eða fleiri heimadæmum og þeirra sem gerðu það ekki
- Sama gildir um fyrirlestraæfingarnar (og mætingu í fyrirlestra)

- Lykilatriðið er að vinna jafnt og þétt yfir misserið
- Þið lærið mest á því að "gera" (ekki "horfa")
  - Þið þurfið að <u>vinna með</u> námsefnið til að skilja það
  - Annars <u>haldið þið</u> bara að þið skiljið það (*Illusion of familiarity*)



Fall: **6%** Meðaleink: **7.73** Fjöldi: 63



Fall: **31%** Meðaleink: **5.53** Fjöldi: 36

Vorið '20 ekki marktækt, því þá mátti fá staðið og sleppa prófinu

Kennslufræði

- Þegar þið lærið nýtt námsefni þá byrjar sú þekking strax að "leka út"
  - Getum hægt á lekanum með því að rifja námsefnið upp
  - Náum langtímalærdómi með því að vinna með efnið

- Besta aðferðin til að ná langtímaþekkingu er að rifja upp úr minni (*retrieval*), ekki að endurlesa
  - Mjög margar vísindarannsóknir styðja þetta (retrievalpractice.org)

- Hér gerum við þetta með fyrirlestraæfingum og einnig með heimadæmum
  - Þið getið líka gert þetta sjálf:

    Gefið ykkur 5-10 mín eftir fyrirlestur til að rifja upp og skrifa niður punkta úr efninu

Reiknirit

- Reiknirit er skýr, nákvæm, ótvíræð lýsing á aðferð til að leysa verkefni eða framkvæma útreikning

- Lýsingin þarf ekki að vera á neinu sérstöku formi
  - Við munum nota "sauðakóða" (*pseudocode*) sem líkist Python
  - Getum notað íslensku, ensku, flæðirit, ljóð:

```
BottlesOfBeer(n):
    For i ← n down to 1
        Sing "i bottles of beer on the wall, i bottles of beer,"
        Sing "Take one down, pass it around, i − 1 bottles of beer on the wall."
    Sing "No bottles of beer on the wall, no bottles of beer,"
    Sing "Go to the store, buy some more, n bottles of beer on the wall."
```

Hver aðgerð þarf að vera augljóslega framkvæmanleg

Hér er grunnaðgerðin að **syngja** um bjórflöskurnar **ekki** að drekka úr þeim!

Líka til ýmsar stærðfræðilegar útgáfur af laginu

Greining reiknirita

- Viljum geta borið saman ólík reiknirit fyrir sama verkefnið

- Skoðum aðallega tvennt:
  - **Er reikniritið rétt?**
    - Þurfum oft að nota þrepun (eða aðrar sönnunaraðferðir) til að sýna að niðurstaðan sé alltaf rétt
  - **Hvað tekur reikniritið langan tíma?**
    - Erfitt að vinna með <u>tíma</u>, við notum frekar <u>skref</u> (grunnaðgerðir, t.d. samanburður, reikniaðgerð, ...)

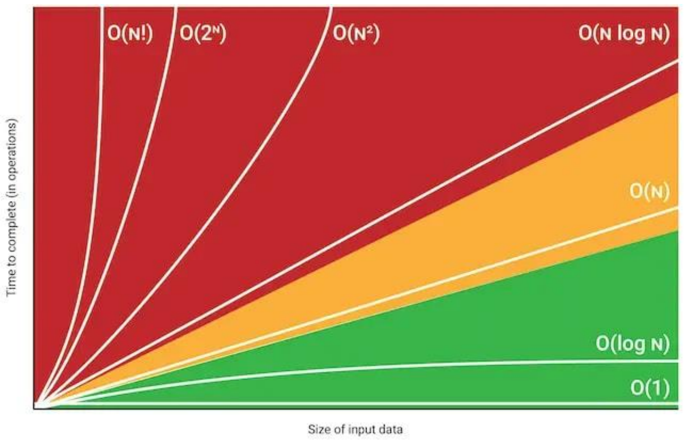
- Við útreikning á tíma (þ.e. fjölda skrefa!) notum við stærðargráður og vaxtarhraða

Líka hægt að skoða <u>magn minnis</u> og <u>hvernig</u> það er notað

Geri ráð fyrir að þið kunnið að vinna með $O(f)$

Tímaflækja (*time complexity*)

- Við höfum reyndar ekki áhuga á <u>fjölda skrefa</u> sem reiknirit tekur...
- ...heldur hvernig fjöldinn vex sem fall af inntaksstærð
  - Margliðutími er í lagi, en veldistími virkar aðeins fyrir lítil tilfelli



- Notum versta-tilfellis tímaflækju:
  - $T(n)$ = mesti fjöldi skrefa sem reikniritið tekur á inntaki af stærð n

- Af hverju ekki meðaltals tímaflækja?
  - Þurfum þá að ákveða dreifingu inntaks − eru öll tilfelli af stærð n jafnlíkleg?
  - Verður oftast mun flóknari útreikningur (fer eftir dreifingu)
  - Versta-tilfellis tímaflækja gefur okkur tryggingu (reikniritið tekur í mesta lagi $T(n)$ tíma)
    - Oftast tekur "meðaltilfellið" ekki mikið betri tíma en versta tilfellið

Greining á bjórflöskulagi

```
BottlesOfBeer(n):
  For i ← n down to 1
      Sing "i bottles of beer on the wall, i bottles of beer,"
      Sing "Take one down, pass it around, i − 1 bottles of beer on the wall."
  Sing "No bottles of beer on the wall, no bottles of beer,"
  Sing "Go to the store, buy some more, n bottles of beer on the wall."
```

- Velja grunnaðgerð: Segja orðið "*beer*"

- Lykkjan er framkvæmd n sinnum, í hverri ítrun sagt "*beer*" þrisvar
  - Svo er sagt "*beer*" þrisvar í lokin
- Samtals: $3n + 3$ sinnum sagt "*beer*" í Bjórflöskulaginu
  - eða $O(n)$ sinnum

Önnur svipuð sönglög

- NDaysOfChristmas(gifts[2..n])
  - Almenn útgáfa af 12 Days Of Christmas (íslensk útgáfa)

Gjöf 1 er nefnd *n* sinnum,
Gjöf 2 er nefnd *n*−1 sinni, o.s.frv.
Samtals: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = O(n^2)$

- Old MacDonald had a Farm
  - Í almennu útgáfunni bætum við einu dýri við í hverri ítrun og endurtökum öll dýrahljóðin
    - Ítrun k hefur k ólík dýrahljóð, svo samtals eru þetta $O(n^2)$ dýrahljóð fyrir n dýr

- Alouette (ensk þýðing)
  - Bætum einn líkamshluta lævirkjans við í hverri ítrun
  - Teljum þá alla upp í hvert sinn, svo þetta er líka $O(n^2)$ fyrir almennt tilvik

Minkurinn í hæsnakofanum
eftir Ómar Ragnarsson

Dæmi um reiknirit: Heiltölumargföldun

Grindarmargföldun (*Lattice multiplication*)

- Gefnar heiltölurnar *x* og *y*, táknaðar sem fylkin *X*[0..*m*−1] og *Y*[0..*n*−1]

$$x = \sum_{i=0}^{m-1} X[i] * 10^i \qquad y = \sum_{j=0}^{n-1} Y[j] * 10^j$$

Til einföldunar eru tölustafirnir í "öfugri" röð, t.d. talan 403 er geymd sem [3, 0, 4]

- Úttakið *z* = *x***y* fer í fylkið *Z*[0..*m*+*n*−1]
- Niðurstaðan úr margfölduninni er

$$xy = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (X[i] * Y[j] * 10^{i+j})$$

tölustafa-margföldun

hliðrun um i+j sæti

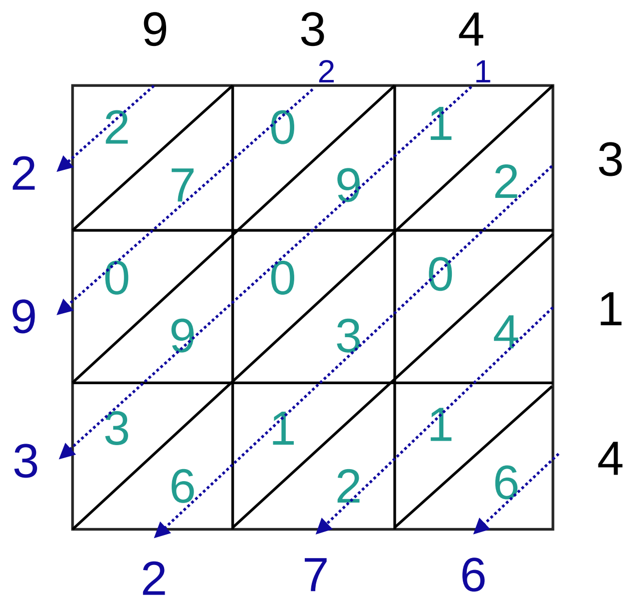
Til nokkrar útgáfur til af reikniritinu eftir útreikningsröð og uppsöfnun hlutsumma

Grindarmargföldun (*Fibonacci*)

$\textsc{FibonacciMultiply}(X[0..m-1], Y[0..n-1])$:
- $hold \leftarrow 0$
- for $k \leftarrow 0$ to $n+m-1$
  - for all i and j such that $i+j=k$
    - $hold \leftarrow hold + X[i]\cdot Y[j]$
  - $Z[k] \leftarrow hold \bmod 10$
  - $hold \leftarrow \lfloor hold/10 \rfloor$
- return $Z[0..m+n-1]$

Finnum hvern tölustaf í útkomunni fyrir sig

- Oft framkvæmt með því að búa til grind (*lattice*)
  - Tölurnar tvær á ásunum
  - Búum til öll hlutmargfeldi í grindinni
    - Tugastafur fyrir ofan skálínu og einingastafur fyrir neðan
- Dæmi: Reiknum 934*314
  - Fyllum inn í alla reitina
  - Summum eftir hornalínum



Útkoma: 293276

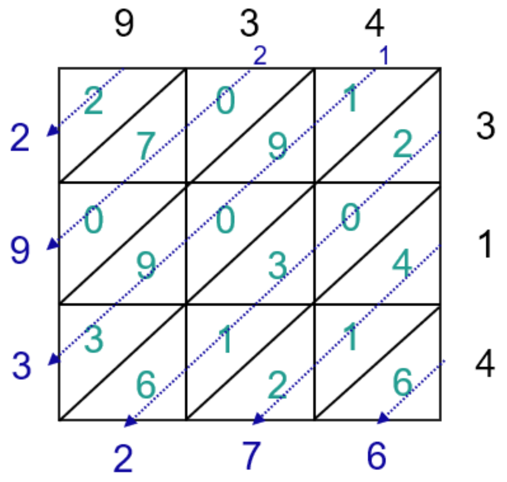
Okkar skólaútgáfa

- Við lærum að margfalda tvær stórar heiltölur með því að margfalda aðra með öllum tölustöfum hinnar og summa svo hliðruðum hlutútkomum
  - Þessi aðferð er önnur útgáfa af grindarmargföldun

- Dæmi: Margfalda 934 og 314:

```
      9 3 4
    * 3 1 4
  ---------
    3 7 3 6
    9 3 4
  2 8 0 2
  ---------
  2 9 3 2 7 6
```

Grunnaðgerð:
Margföldun tveggja tölustafa



Sami fjöldi grunnaðgerða í báðum aðferðum: ***m*n***

Bændaaðferðin

- Ókostur við grindaraðferð: Þurfum að kunna margföldunartöflur frá 1 til 9

Þess vegna verið að kenna margföldunartöflurnar í grunnskóla!

- Eldri aðferð: (Rússneska) bændaaðferðin
  - Var þekkt af forn-egyptum, kennd í skólum í austur Evrópu

- Hugmynd:
  - Viljum finna $x*y$.
  - Helmingum aðra töluna (t.d. x) og tvöföldum hina þar til við fáum $1*z = x*y$, þá er z útkoman!

  - Dæmi: 6*8 = 48 (viljum enda með 1*48 = 48)

en þá er 3*16 líka = 48   en þar sem 3 er oddatala þá þarf að geyma eitt eintak af 16:

þá er 2*16 + 16 = 48

og þá er 1*32 + 16 = 48

Bændaaðferðin, reiknirit

```
PeasantMultiply(x, y):
    prod ← 0
    while x > 0
        if x is odd
            prod ← prod + y
        x ← ⌊x/2⌋
        y ← y + y
    return prod
```

Byggir á jöfnunni:

$$x \cdot y = \begin{cases} 0, & \text{ef } x = 0 \\ \left\lfloor \frac{x}{2} \right\rfloor \cdot 2y, & \text{ef } x \text{ er jöfn} \\ \left\lfloor \frac{x}{2} \right\rfloor \cdot 2y + y, & \text{ef } x \text{ er odda} \end{cases}$$

- Einfaldari grunnaðgerðir:
  - Samlagning
  - Deiling með 2
  - Margföldun með 2 (þ.e. samlagning)
- Fjöldi ítrana:
  - Gildið *x* helmingast í hvert sinn, svo $\log_2 x$ ítranir
  - Hver aðgerð er reyndar á stærri tölum (t.d. samlagning á *n*-tölustafa tölum)

Bændaaðferð, greining

```
PEASANTMULTIPLY(x, y):
  prod ← 0
  while x > 0
      if x is odd
          prod ← prod + y
      x ← ⌊x/2⌋
      y ← y + y
  return prod
```

- Þetta er reiknirit er í grunninn endurkvæmt
  - Sýnum ítrunarútgáfu!

- Hver aðgerð í lykkjunni þarf $O(\log(xy)) = O(\log(\max\{x, y\})$ eins-tölustafs aðgerðir
  - Fjöldi ítrana er $O(\log(\min\{x, y\}))$, því við getum víxlað á x og y
- Samtals: $O(\log \min\{x, y\} \cdot \log \max\{x, y\}) = \boldsymbol{O(\log(x) \cdot \log(y))}$

Tímaflækja er oftast gefin sem fall af **inntaksstærðinni**

- Munið að *x* og *y* eru gildin, ekki inntaksstærðin
- Við þurfum því $O(mn)$ tíma til að margfalda *m*-stafa tölu við *n*-stafa tölu

Fyrirlestraæfingar

1. Hvers vegna getur verið gott að taka stutt próf úr einhverju efni **áður en** farið er í efnið?

2. Margfaldið tölurnar 17 og 24 með grindarmargföldun (*lattice multiplication*)

3. Margfaldið tölurnar 17 og 24 með bændareikniritinu