# Gráðug reiknirit

Bergur Snorrason

20. janúar 2023

# Gráðug reiknirit

- Reiknirit sem tekur í hverju skrefi ákvörðun byggða á því hvað lítur best út á þeim tímapunkti er sagt vera *gráðugt*.
- Höfum í huga að það er alls ekki sjálfsagt að gráðugt reiknirit skili réttri lausn.
- Tökum dæmi.

# Dæmi

- Þú vinnur í sjoppu of þarft að gefa $n$ krónur í afgang. Í boði er ótakmarkað magn af 1 krónum, 5 krónum og 10 krónum. Hver er minnsti fjöldi af klinki sem þú getur gefið?
- Dæmi um lausn er:
- Látum $k$ tákna dýrasta klinkið sem er ekki dýrara en $n$. Gefum til baka $k$ og endurtökum fyrir $n - k$ þangað til $n$ er 0.
- Tökum til dæmis $n = 24$. Þá myndum við gefa $10, 10, 1, 1, 1, 1$ til baka.
- Það vill svo skemmtilega til að þessi gráðuga lausn virkar fyrir öll $n$.
- En hvað ef við breytum aðeins dæminu?

# Dæmi

- Þú vinnur í sjoppu of þarft að gefa $n$ krónur í afgang. Í boði er ótakmarkað magn af 1 krónum, 8 krónum og 20 krónum. Hver er minnsti fjöldi af klinki sem þú getur gefið?
- Tökum til dæmis $n = 24$.
- Þá myndi aðferðin í glærunni á undan gefa $20, 1, 1, 1, 1$ til baka.
- En þetta er ekki besta leiðin.
- Betra væri að gefa $8, 8, 8$.
- Svo gráðuga aðferðin virkar bara stundum.
- Þetta dæmi er þekkt sem Skiptimyntadæmið (e. *The Coin Change Problem*) og það er ekki augljóst hvenær gráðuga lausnin virkar.
- Til að leysa þetta dæmi almennt notum við *kvika bestun*.

# Annað dæmi

- Þú ert yfirmaður hjá leigubílafyrirtæki. Í dag mættu $n$ bílstjórar í vinnuna og það eru $m$ leigubílar til að skipa bílstjóra á. Ekki eru þó allir bílstjórar og leigubílar skapaðir jafnir. Bíll $i$ er $h_i$ hestöfl og bílstjóri $j$ getur keyrt bíla sem eru $g_j$ hestöfl eða minna. Hver er mesti fjöldi bíla sem þú getur skipað á bílstjóra þannig að hver bílstjóri fær mest einn bíl, hver bíll er með mest einn bílstjóra og enginn bílstjóri fær bíl sem hann ræður ekki við. Gerum ráð fyrir að bílarnir og bílstjórarnir séu gefnir í vaxandi röð.
- Tökum eftir að við viljum að bíll sé keyrður af þeim bílstjóra sem er óreyndastur (en þó nógu hæfur til að keyra hann).
- Við getum því í línulegum tíma fundið þann bílstjóra sem getur keyrt fyrsta bílinn.
- Til að finna bílstjóra fyrir næstu bíla leitum við aftur línulega, en höldum áfram þaðan sem við hættum áðan.

- Að neðan er útfærsla á lausninni á glærunum hér á undan. Fyrstu tvær tölurnar á inntakinu eru $n$ og $m$. Næstu $n$ tölur lýsa getu bílstjóranna. Síðustu $m$ tölurnar eru hestöfl bílanna.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, m, i, j, x, r;
6      scanf("%d%d", &n, &m);
7      int a[n], b[m];
8      for (i = 0; i < n; i++) scanf("%d", &a[i]);
9      for (i = 0; i < m; i++) scanf("%d", &b[i]);
10     for (i = r = j = 0; i < m; i++)
11     {
12         while (j < n && a[j] < b[i]) j++;
13         if (j < n) r++, j++;
14     }
15     printf("%d\n", r);
16     return 0;
17 }
```

# Lausn

- Þessi lausn er $\mathcal{O}(n + m)$.

# Gráðuga röðun

- Áðan röðuðum við tölum með tæmandi leit.
- Endurtökum nú leikinn með gráðugri nálgun.
- Ef við ætlum að raða í vaxandi röð þá vitum við að minnsta talan fer fremst.
- Við getum því raðað listanum gráðugt með því að setja ítrekað fremst minnstu töluna sem er eftir.

```
 4 void sort(int *a, int n)
 5 {
 6     int i, j, mn;
 7     for (i = 0; i < n; i++)
 8     {
 9         for (j = mn = i; j < n; j++) if (a[mn] > a[j]) mn = j;
10         swap(a[i], a[mn]);
11     }
12 }
```

- Tímaflækjan á þessu röðunarreikniriti er $\mathcal{O}(n^2)$.
- Við getum þó bætt tímaflækjuna þegar við leitum að minnsta stakinu.
- Það eru ýmsar gagnagrindir sem geta gert þetta fyrir okkur.
- Af sögulegum ástæðum notum við forgangsbiðröð, en það mætti, til dæmis, líka nota ýmis röðunartré.

```
 4 void sort(int *a, int n)
 5 {
 6     int i;
 7     priority_queue<int> q;
 8     for (i = 0; i < n; i++) q.push(-a[i]);
 9     for (i = 0; i < n; i++) a[i] = -q.top(), q.pop();
10 }
```

- Núna finnum við minnsta stakið í $\mathcal{O}(\log n)$.
- Svo tímaflækjan á fallinu er $\mathcal{O}(n \log n)$.
- Við getum í rauninni ekki raðað með betri tímaflækju.
- Þessi röðun kallast *hrúguröðun* (e. *heap sort*) og er þekkt röðunarreiknirit sem er oft kennt í inngangsnámskeiðum.

# Samantekt

- Það fyrsta sem manni dettur í hug þegar maður les dæmi er oft gráðug lausn.
- Það getur samt verið erfitt að sanna að gráðug lausn sé rétt.
- Það dugar þó að sannfæra sjálfan sig (og liðsfélaga í liðakeppnum).
- Ef maður útfærir og sendir inn gráðuga lausn og fær `Wrong Answer` getur verið erfitt að átta sig á því hvort maður hafi gert villu eða hvort gráðuga lausnin sé röng.

# Samanburður

- Tæmandi leit er oft auðvelt að bera kennsl á, en á það til að vera of hæg.
- Hér má búast við `Time Limit Exceeded`, en eitthvað furðulegt hefur gerst ef maður fær `Wrong Answer`.
- Það er oft létt að semja gráðug reiknirit, en það getur verið mikil vinna að sanna að lausnin sem maður fær sé alltaf rétt.
- Gráðug reiknirit eiga ekki að fá `Time Limit Exceeded`, en algengt er að þau fái `Wrong Answer`.