

# Rótarþáttun

Bergur Snorrason

February 19, 2024

## Dæmi

- ▶ Gefinn er listi með  $n$  tölum.
- ▶ Næst koma  $q$  fyrirspurnir, þar sem hver er af einni af tveimur gerðum:
  - ▶ Bættu  $k$  við  $i$ -tu töluna.
  - ▶ Reiknaðu summu allra talna á bilinu  $[i, j]$ .

## Almenn $k$ -þáttun

- ▶ Hvað ef við skiptum fylkinu upp í  $k$  (næstum) jafnstór hólf.
- ▶ Við getum þá haldið utan um, og uppfært, summu hvers hólfis auðveldlega.
- ▶ Til að finna summu á einhverju bili í fylkinu nægir að reikna summu hólfana á milli endapunktana og leggja svo afganginn við (afgangurinn er í mesta lagi lengd tveggja hólfra).
- ▶ Tökum eftirfarandi sýnidæmi sem skipt hefur verið í þrjú hólf,

$$p = [0 \ 1 \ 4 \mid 3 \ 4 \ 5 \mid 0 \ 1 \ 8 \ 9].$$

- ▶ Köllum fylkið sem geymir summu hvers hólfis  $s$ , sem verður þá

$$s = [5 \ 12 \ 18].$$

- ▶ Ef við viljum uppfæra, til dæmis bæta 5 við stak 2, þá þurfum við að sjálfsögðu að uppfæra `p`, en það þarf líka breyta `s`.
- ▶ Til að breyta `p` gerum við einfaldlega `p[2] += 5`.
- ▶ Til að uppfæra `s` þurfum við að finna hólfið sem stak 2 tilheyrir. Þar sem það er í hólfi 0 notum við `s[0] += 5`.
- ▶ Svona líta svo fylkin út, fyrir og eftir uppfærslu.

| Fyrir breytingu                                     | Eftir breytingu                                     |
|---|---|
| $p = [0 \ 1 \ 4 \mid 3 \ 4 \ 5 \mid 0 \ 1 \ 8 \ 9]$ | $p = [0 \ 1 \ 9 \mid 3 \ 4 \ 5 \mid 0 \ 1 \ 8 \ 9]$ |
| $s = [5 \ 12 \ 18]$                                 | $s = [10 \ 12 \ 18]$                                |

- ▶ Ég fór mjög losaralega í hvernig ætti að framkvæma seinni aðgerðina.
- ▶ Skoðum, sem dæmi, hverju eigi að skila fyrir  $2\ 1\ 8$ .
- ▶ Það er aðeins eitt hólf á milli staks  $1$  og staks  $8$ , hólf  $1$ .
- ▶ „Afgangurinn”, eins og ég kallaði hann áðan, eru þau stök sem ekki eru í hólfi  $1$  en eru þó á bilinu frá  $1$  til  $8$ .
- ▶ Þetta eru stök  $1$ ,  $2$ ,  $6$ ,  $7$  og  $8$  (samtals summan er þá 31).
- ▶ Við erum því að leggja saman rauðu stökin á myndinni fyrir neðan,

$$p = [0\ 1\ 9 \mid 3\ 4\ 5 \mid 0\ 1\ 8\ 9]$$

$$s = [10\ 12\ 18]$$

- ▶ En er þetta hraðar en frumstæða aðferðin sem við skoðuðum í upphafi?
- ▶ Það fer að sjálfsögðu allt eftir því hversu stór hólf við veljum.
- ▶ Ef fylkinu er skipt upp í  $n$  hólf er nokkuð ljóst að þessi aðferð er jafngild frumstæðu aðferðinni.
- ▶ Ef fylkinu er skipt upp í 1 hólf gildir það sama.
- ▶ Munið að við létum  $k$  tákna fjölda hólfa.
- ▶ Fyrri aðgerðin er ennþá  $\mathcal{O}(1)$ , en seinni aðgerðin verður  $\mathcal{O}(n/k + k)$ , svo tímaflækjan er  $\mathcal{O}(qn/k + qk)$ .

## Skynsamlegt val á $k$

- ▶ Þar sem að fyrri aðgerðin er ekki háð skiptingunni þá nægir að lágmarka  $\frac{n}{k} + k$ .
- ▶ Látum  $f(k) = \frac{n}{k} + k$ .
- ▶ Við höfum  $f'(k) = -\frac{n}{k^2} + 1$ .
- ▶ Útgildispunktur fást í

$$\begin{aligned}f'(k) &= 0 \\ \Rightarrow 1 - \frac{n}{k^2} &= 0 \\ \Rightarrow 1 &= \frac{n}{k^2} \\ \Rightarrow k^2 &= n \\ \Rightarrow k &= \sqrt{n}.\end{aligned}$$

- ▶ Nú þarf bara að ganga úr skugga um að þessi skipting sé betri en línuleg.

- ▶ Ef við veljum  $k = \sqrt{n}$  þá er tímaflækja seinni aðgerðarinnar

$$\mathcal{O}\left(\frac{n}{\sqrt{n}} + \sqrt{n}\right) = \mathcal{O}(\sqrt{n} + \sqrt{n}) = \mathcal{O}(\sqrt{n}).$$

- ▶ Því er tímaflækjan á lausninni  $\mathcal{O}(q\sqrt{n})$ .
- ▶ Svo þessi aðferð er betri en sú frumstæða, ef við skiptum í  $\sqrt{n}$  hólf.
- ▶ Við köllum það *rótarþáttun* (e. *squareroot decomposition*) þegar við skiptum upp í  $\sqrt{n}$  hólf.



```

7 void update(int *a, int x, int y)
8 {
9     a[2 + a[0] + x/a[1]] += y, a[2 + x] += y;
10 }
11
12 int query(int *a, int x, int y)
13 {
14     int r = 0;
15     while (x%a[1] != 0 && x < y) r += a[2 + x++];
16     if (x == y) return r;
17     while (y%a[1] != 0) r += a[2 + --y];
18     while (x < y) r += a[2 + a[0] + x/a[1]], x += a[1];
19     return r;
20 }
21
22 void init(int *a, int n)
23 {
24     for (a[0] = n, a[1] = 0; a[1]*a[1] < a[0]; a[1]++);
25     for (int i = 2; i < 2*n; i++) a[i] = 0;
26 }

```

## Lygnar uppfærslur

- ▶ Við getum einnig framkvæmt lygnar uppfærslur þegar við notum rótarþáttun (líkt og með biltré).
- ▶ Við uppfærum þá beint þau gildi sem eru í sömu hólfum og endarpunktur bilsins sem við uppfærum yfir.
- ▶ Við framkvæmum svo lygna uppfærslu á þau hólf sem liggja þar á milli.

```

7 void prop(int *a, int x)
8 {
9     for (int i = 0; i < a[1]; i++) a[3 + x*a[1] + i] += a[a[2] + x];
10    a[3 + a[0] + x] += a[a[2] + x]*a[1], a[a[2] + x] = 0;
11 }
12
13 int query(int *a, int x, int y)
14 {
15     prop(a, x/a[1]), prop(a, (y - 1)/a[1]);
16     int r = 0;
17     while (x%a[1] != 0 && x < y) r += a[3 + x++];
18     if (x == y) return r;
19     while (y%a[1] != 0) r += a[3 + --y];
20     while (x < y) r += a[3 + a[0] + x/a[1]] + a[a[2] + x/a[1]]*a[1], x += a[1];
21     return r;
22 }
23
24 void update(int *a, int x, int y, int z)
25 {
26     prop(a, x/a[1]), prop(a, (y - 1)/a[1]);
27     while (x%a[1] != 0 && x < y) a[3 + x] += z, a[3 + a[0] + x++/a[1]] += z;
28     if (x == y) return;
29     while (y%a[1] != 0) a[3 + --y] += z, a[3 + a[0] + y/a[1]] += z;
30     while (x < y) a[a[2] + x/a[1]] += z, x += a[1];
31 }
32
33 void init(int *a, int n)
34 {
35     for (a[0] = n, a[1] = 0; a[1]*a[1] < a[0]; a[1]++);
36     for (int i = 3; i < 2*a[0]; i++) a[i] = 0;
37     a[2] = (a[0] + a[1] - 1)/a[1] + a[0] + 3;
38 }

```

