Gagnagrindur

Rótarþáttun, Fenwick-tré og fylkjaaðgerðir

Bergur Snorrason

8. febrúar 2021

Efnisyfirlit

Keppnin

- Miðmisseriskeppnin verður haldin í næstu viku.

Keppnin

- Miðmisseriskeppnin verður haldin í næstu viku.
- Í keppninni verða sex dæmi.

Keppnin

- Miðmisseriskeppnin verður haldin í næstu viku.
- Í keppninni verða sex dæmi.
- Keppnin hefst $15:00$ og lýkur $18:00$.

Keppnin

- Miðmisseriskeppnin verður haldin í næstu viku.
- Í keppninni verða sex dæmi.
- Keppnin hefst $15:00$ og lýkur $18:00$.
- Keppnin kemur í stað dæmaskila þá vikuna.

Efnisyfirlit

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fyrsta lína inntaksins inniheldur tvær heiltölur n og m.

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fyrsta lína inntaksins inniheldur tvær heiltölur n og m.
- Næsta lína innheldur lista af n helitölum.

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fyrsta lína inntaksins inniheldur tvær heiltölur n og m.
- Næsta lína innheldur lista af n helitölum.
- Næstu m línur verða af tveimur gerðum:

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fyrsta lína inntaksins inniheldur tvær heiltölur n og m.
- Næsta lína innheldur lista af n helitölum.
- Næstu m línur verða af tveimur gerðum:
    - `1 q p` þýðir að breyta eigi p-ta staki listans í q.

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fyrsta lína inntaksins inniheldur tvær heiltölur n og m.
- Næsta lína innheldur lista af n helitölum.
- Næstu m línur verða af tveimur gerðum:
  - `1 q p` þýðir að breyta eigi p-ta staki listans í q.
  - `2 q p` þýðir að prenta eigi summu stakanna í q-ta til p-ta sæti.

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Rennum í gegnum eitt sýnidæmi. Látum inntakið vera

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Rennum í gegnum eitt sýnidæmi. Látum inntakið vera
- 10 6
  0 1 2 3 4 5 6 7 8 9
  2 0 9
  2 0 4
  1 2 4
  1 6 0
  1 7 1
  2 1 8

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fylkið byrjar sem $[0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fylkið byrjar sem $[0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.
- Eftir `2 0 9` á að skila $0+1+2+3+4+5+6+7+8+9=45$.

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fylkið byrjar sem $[0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.
- Eftir `2 0 9` á að skila $0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45$.
- Eftir `2 0 4` á að skila $0 + 1 + 2 + 3 + 4 = 10$.

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fylkið byrjar sem $[0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.
- Eftir `2 0 9` á að skila $0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45$.
- Eftir `2 0 4` á að skila $0 + 1 + 2 + 3 + 4 = 10$.
- Eftir `1 2 4` verður fylkið $[0\ 1\ 4\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fylkið byrjar sem $[0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.
- Eftir `2 0 9` á að skila $0+1+2+3+4+5+6+7+8+9=45$.
- Eftir `2 0 4` á að skila $0+1+2+3+4=10$.
- Eftir `1 2 4` verður fylkið $[0\ 1\ 4\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.
- Eftir `1 6 0` verður fylkið $[0\ 1\ 4\ 3\ 4\ 5\ 0\ 7\ 8\ 9]$.

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fylkið byrjar sem $[0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.
- Eftir 2 0 9 á að skila $0+1+2+3+4+5+6+7+8+9 = 45$.
- Eftir 2 0 4 á að skila $0+1+2+3+4 = 10$.
- Eftir 1 2 4 verður fylkið $[0\ 1\ 4\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.
- Eftir 1 6 0 verður fylkið $[0\ 1\ 4\ 3\ 4\ 5\ 0\ 7\ 8\ 9]$.
- Eftir 1 7 1 verður fylkið $[0\ 1\ 4\ 3\ 4\ 5\ 0\ 1\ 8\ 9]$.

Dæmi þannig að varla þarf á öðrum dæmi að halda

- Fylkið byrjar sem $[0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.
- Eftir 2 0 9 á að skila $0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45$.
- Eftir 2 0 4 á að skila $0 + 1 + 2 + 3 + 4 = 10$.
- Eftir 1 2 4 verður fylkið $[0\ 1\ 4\ 3\ 4\ 5\ 6\ 7\ 8\ 9]$.
- Eftir 1 6 0 verður fylkið $[0\ 1\ 4\ 3\ 4\ 5\ 0\ 7\ 8\ 9]$.
- Eftir 1 7 1 verður fylkið $[0\ 1\ 4\ 3\ 4\ 5\ 0\ 1\ 8\ 9]$.
- Eftir 2 1 8 á að skila $1 + 4 + 3 + 4 + 5 + 0 + 1 + 8 = 26$.

Frumstæð lausn

- Það liggur helst við að leysa þetta línulega.

Frumstæð lausn

- Það liggur helst við að leysa þetta línulega.
- Við látum allar tölurnar í n staka fylki.

Frumstæð lausn

- Það liggur helst við að leysa þetta línulega.
- Við látum allar tölurnar í n staka fylki.
- Fyrir fyrri aðgerðina breytum við tilheyrandi staki í fylkinu.

Frumstæð lausn

- Það liggur helst við að leysa þetta línulega.
- Við látum allar tölurnar í n staka fylki.
- Fyrir fyrri aðgerðina breytum við tilheyrandi staki í fylkinu.
- Fyrir seinni aðgerðina löbbum við í gegnum fylkið og reiknum summuna.

Frumstæð lausn

- Það liggur helst við að leysa þetta línulega.
- Við látum allar tölurnar í n staka fylki.
- Fyrir fyrri aðgerðina breytum við tilheyrandi staki í fylkinu.
- Fyrir seinni aðgerðina löbbum við í gegnum fylkið og reiknum summuna.
- Fyrri aðgerðin er $\mathcal{O}(1)$ og seinni aðgerðin er $\mathcal{O}(n)$, svo tímaflækjan er $\mathcal{O}(nm)$.

Efnisyfirlit

Almenn k-þáttun

- Hvað ef við skiptum fylkinu upp í k hólf.

Almenn k-þáttun

- Hvað ef við skiptum fylkinu upp í k hólf.
- Við getum þá haldið utan um, og uppfært, summu hvers hólfs auðveldlega.

Almenn k-þáttun

- Hvað ef við skiptum fylkinu upp í k hólf.
- Við getum þá haldið utan um, og uppfært, summu hvers hólfs 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 hólfa).

Almenn k-þáttun

- Hvað ef við skiptum fylkinu upp í k hólf.
- Við getum þá haldið utan um, og uppfært, summu hvers hólfs 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 hólfa).
- Ef við skiptum fylkinu úr sýnidæminu upp í 3 hólf þá verður það

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

Almenn k-þáttun

- Hvað ef við skiptum fylkinu upp í k hólf.
- Við getum þá haldið utan um, og uppfært, summu hvers hólfs 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 hólfa).
- Ef við skiptum fylkinu úr sýnidæminu upp í 3 hólf þá verður það

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

- Köllum fylkið sem geymir summu hvers hólfs s. Það verður nú

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

Fyrri aðgerð með almennri k-þáttun

- Ef við viljum uppfæra, til dæmis breyta staki 2 í 9 þá þurfum við að sjálfsögðu að uppfæra p, en það þarf líka breyta s.

Fyrri aðgerð með almennri k-þáttun

- Ef við viljum uppfæra, til dæmis breyta staki 2 í 9 þá þurfum við að sjálfsögðu að uppfæra p, en það þarf líka breyta s.
- Til að breyta p gerum við einfaldlega `p[2] = 9`.

Fyrri aðgerð með almennri k-þáttun

- Ef við viljum uppfæra, til dæmis breyta staki 2 í 9 þá þurfum við að sjálfsögðu að uppfæra p, en það þarf líka breyta s.
- Til að breyta p gerum við einfaldlega `p[2] = 9`.
- Til að uppfæra s þurfum við að finna hólfið sem stak 2 tilheyrir. Þar sem það er í hólfi 0 notum við `s[0] = s[0] - p[2] + 9`.

Fyrri aðgerð með almennri k-þáttun

- Ef við viljum uppfæra, til dæmis breyta staki 2 í 9 þá þurfum við að sjálfsögðu að uppfæra p, en það þarf líka breyta s.
- Til að breyta p gerum við einfaldlega `p[2] = 9`.
- Til að uppfæra s þurfum við að finna hólfið sem stak 2 tilheyrir. Þar sem það er í hólfi 0 notum við `s[0] = s[0] - p[2] + 9`.
- Glöggir lesendur taka þó eftir að við þurfum að uppfæra s **áður** en við uppfærum p, þar sem við notum gamla gildi p þegar við uppfærum s.

Fyrri aðgerð með almennri k-þáttun

- Ef við viljum uppfæra, til dæmis breyta staki 2 í 9 þá þurfum við að sjálfsögðu að uppfæra p, en það þarf líka breyta s.
- Til að breyta p gerum við einfaldlega `p[2] = 9`.
- Til að uppfæra s þurfum við að finna hólfið sem stak 2 tilheyrir. Þar sem það er í hólfi 0 notum við `s[0] = s[0] - p[2] + 9`.
- Glöggir lesendur taka þó eftir að við þurfum að uppfæra s **áður** en við uppfærum p, þar sem við notum gamla gildi p þegar við uppfærum s.
- Svona líta svo fylkin út, fyrir og eftir uppfærslu.

$$
\begin{array}{cc}
\text{Fyrir breytingu} & \text{Eftir breytingu} \\
p = [0\ 1\ 4 \mid 3\ 4\ 5 \mid 0\ 1\ 8\ 9] & p = [0\ 1\ 9 \mid 3\ 4\ 5 \mid 0\ 1\ 8\ 9] \\
s = [5\ 12\ 18] & s = [10\ 12\ 18]
\end{array}.
$$

Seinni aðgerðin með almennri k-þáttun

- Ég fór mjög losaralega í hvernig ætti að framkalla seinni aðgerðina.

Seinni aðgerðin með almennri k-þáttun

- Ég fór mjög losaralega í hvernig ætti að framkalla seinni aðgerðina.
- Skoðum, sem dæmi, hverju eigi að skila fyrir `2 1 8`.

Seinni aðgerðin með almennri k-þáttun

- Ég fór mjög losaralega í hvernig ætti að framkalla 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.

Seinni aðgerðin með almennri k-þáttun

- Ég fór mjög losaralega í hvernig ætti að framkalla 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 það áðan, eru þau stök sem ekki eru í hólfi 1 en eru þó á bilinu frá 1 til 8.

Seinni aðgerðin með almennri k-þáttun

- Ég fór mjög losaralega í hvernig ætti að framkalla 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 það 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).

Seinni aðgerðin með almennri k-þáttun

- Ég fór mjög losaralega í hvernig ætti að framkalla 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 það 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.

$$p = [0\ 1\ 9 \mid 3\ 4\ 5 \mid 0\ 1\ 8\ 9]$$
$$s = [10\ 12\ 18]$$

Seinni aðgerðin með almennri k-þáttun

- Ég fór mjög losaralega í hvernig ætti að framkalla 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 það 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.

$$\begin{aligned} p &= [0\ \color{red}{1}\ \color{red}{9}\ |\ \color{blue}{3}\ \color{blue}{4}\ \color{blue}{5}\ |\ \color{red}{0}\ \color{red}{1}\ \color{red}{8}\ 9] \\ s &= [10\ \color{red}{12}\ 18] \end{aligned}.$$

- Þið getið ímyndað ykkur hvað við getum sparað mikinn tíma fyrir stærri fylki (sem við skiptum upp í fleiri hólf).

Tímflækja almennar k-þáttunar

- En er þetta hraðar en frumstæða aðferðin sem við skoðuðum í upphafi?

Tímflækja almennar k-þáttunar

- En er þetta hraðar en frumstæða aðferðin sem við skoðuðum í upphafi?
- Það fer að sjálfsögðu allt eftir því hversu stór hólf við veljum.

Tímflækja almennar k-þáttunar

- 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.

Tímflækja almennar k-þáttunar

- 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.

Tímflækja almennar k-þáttunar

- 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 látum k tákna fjölda hólfa.

Tímflækja almennar k-þáttunar

- 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 látum k tákna fjölda hólfa.
- Fyrri aðgerðin er ennþá $\mathcal{O}(1)$, en seinni aðgerðin verður $\mathcal{O}\left(\frac{n}{k} + k\right)$, svo tímaflækjan er $\mathcal{O}\left(\frac{mn}{k} + mk\right)$.

Skynsamlegt val á k (Þið megið sleppa þessari glæru ef þið viljið)

- Þar sem að fyrrir aðgerðin er ekki háð skiptingunni þá nægir að lágmarka $\frac{n}{k} + k$.

Skynsamlegt val á k (Þið megið sleppa þessari glæru ef þið viljið)

- Þar sem að fyrrir aðgerðin er ekki háð skiptingunni þá nægir að lágmarka $\frac{n}{k} + k$.
- Látum $f(k) = \frac{n}{k} + k$.

Skynsamlegt val á k (Þið megið sleppa þessari glæru ef þið viljið)

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

Skynsamlegt val á k (Þið megið sleppa þessari glæru ef þið viljið)

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

Skynsamlegt val á k (Þið megið sleppa þessari glæru ef þið viljið)

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

$$f'(k) = 0$$

Skynsamlegt val á k (Þið megið sleppa þessari glæru ef þið viljið)

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

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

Skynsamlegt val á k (Þið megið sleppa þessari glæru ef þið viljið)

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

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

Skynsamlegt val á k (Þið megið sleppa þessari glæru ef þið viljið)

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

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

Skynsamlegt val á k (Þið megið sleppa þessari glæru ef þið viljið)

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

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

Skynsamlegt val á k (Þið megið sleppa þessari glæru ef þið viljið)

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

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

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

Skynsamlegt val á k

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

Skynsamlegt val á k

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

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

Skynsamlegt val á k

- 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 $\mathcal{O}(m\sqrt{n})$.

Skynsamlegt val á k

- 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 $\mathcal{O}(m\sqrt{n})$.
- Svo þessi aðferð er betri en sú frumstæða, ef við skiptum í $\sqrt{n}$ hólf.

Skynsamlegt val á k

- 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 $\mathcal{O}(m\sqrt{n})$.
- 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.

Efnisyfirlit

Fenwick-tré

- Er einhver fljót leið til að reikna summu forskeyti listans?

Fenwick-tré

- Er einhver fljót leið til að reikna summu forskeyti listans?
- Svarið er "já, nokkrun veginn".

Fenwick-tré

- Er einhver fljót leið til að reikna summu forskeyti listans?
- Svarið er "já, nokkrun veginn".
- Þetta er unnt að gera með svo kölluðum Fenwick-trjám.

Fenwick-tré

- Er einhver fljót leið til að reikna summu forskeyti listans?
- Svarið er "já, nokkrun veginn".
- Þetta er unnt að gera með svo kölluðum Fenwick-trjám.
- Skoðum fyrst nokkur atriði um framsetningu talna.

Tölur í tvíundakerfi

- Við könnumst flest við tíundaframsetningu talna, t.d.

$$12045 = 1 \cdot 10^4 + 2 \cdot 10^3 + 0 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$$

Tölur í tvíundakerfi

- Við könnumst flest við tíundaframsetningu talna, t.d.

$$12045 = 1 \cdot 10^4 + 2 \cdot 10^3 + 0 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$$

- Hvað ef við notum annan veldisstofn en 10, t.d. 2?

Tölur í tvíundakerfi

- Við könnumst flest við tíundaframsetningu talna, t.d.

$$12045 = 1 \cdot 10^4 + 2 \cdot 10^3 + 0 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$$

- Hvað ef við notum annan veldisstofn en 10, t.d. 2?
- Það köllum við bitaframsetningu og munum tákna tölur settar fram með bitaframsetningu með 2 í fótskrift, t.d.

$$13 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1101_2$$

Tölur í tvíundakerfi

- Við könnumst flest við tíundaframsetningu talna, t.d.

$$12045 = 1 \cdot 10^4 + 2 \cdot 10^3 + 0 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$$

- Hvað ef við notum annan veldisstofn en 10, t.d. 2?
- Það köllum við bitaframsetningu og munum tákna tölur settar fram með bitaframsetningu með 2 í fótskrift, t.d.

$$13 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1101_2$$

- Stærsti biti tölunnar b er fremsti ásinn í tvíundaframsetningu b.

Tölur í tvíundakerfi

- Við könnumst flest við tíundaframsetningu talna, t.d.

$$12045 = 1 \cdot 10^4 + 2 \cdot 10^3 + 0 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$$

- Hvað ef við notum annan veldisstofn en 10, t.d. 2?
- Það köllum við bitaframsetningu og munum tákna tölur settar fram með bitaframsetningu með 2 í fótskrift, t.d.

$$13 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1101_2$$

- Stærsti biti tölunnar b er fremsti ásinn í tvíundaframsetningu b.
- Minnsti biti tölunnar b er aftasti ásinn í tvíundaframsetningu b.

Skilgreining Fenwick-trjáa

- Fenwick-tré er tré sem inniheldur lykil k og gildi x í hverri nóðu. Ef það eru n nóður í trénu er $0 \leq k < n$.

Skilgreining Fenwick-trjáa

- Fenwick-tré er tré sem inniheldur lykil k og gildi x í hverri nóðu. Ef það eru n nóður í trénu er $0 \leq k < n$.
- Nóðan með lykil 0 er rót trésins.

Skilgreining Fenwick-trjáa

- Fenwick-tré er tré sem inniheldur lykil k og gildi x í hverri nóðu. Ef það eru n nóður í trénu er $0 \leq k < n$.
- Nóðan með lykil 0 er rót trésins.
- Nóða með lykil k er barn nóðu með lykil m ef og aðeins ef k og m hafa sömu bitaframsetningu fyrir utan minnsta bita k.

Skilgreining Fenwick-trjáa

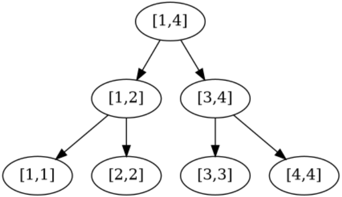
- Fenwick-tré er tré sem inniheldur lykil k og gildi x í hverri nóðu. Ef það eru n nóður í trénu er $0 \leq k < n$.
- Nóðan með lykil 0 er rót trésins.
- Nóða með lykil k er barn nóðu með lykil m ef og aðeins ef k og m hafa sömu bitaframsetningu fyrir utan minnsta bita k.
- ATH: Fenwick-tré eru ekki tvíundatré.

Nokkur dæmi

- Skoðum bitframsetningu talna minni en 16.

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Mynd af Fenwick-tréi með 16 nóður



Útfærslan á járninu

- Þar sem að lykarnir eru einræðir og ekki stærri en n þá getum við einfaldlega geymt gögnin í fylki.

Útfærslan á járninu

- Þar sem að lykarnir eru einræðir og ekki stærri en n þá getum við einfaldlega geymt gögnin í fylki.
- Nánar tiltekið, ef gildi x er í nóðu með lykil k látum við stak k í fylkinu bera gildið x.

Útfærslan á járninu

- Þar sem að lykarnir eru einræðir og ekki stærri en n þá getum við einfaldlega geymt gögnin í fylki.
- Nánar tiltekið, ef gildi x er í nóðu með lykil k látum við stak k í fylkinu bera gildið x.
- Okkar vantar bara leið til að ferðast í trénu.

Útfærslan á járninu

- Þar sem að lykarnir eru einræðir og ekki stærri en n þá getum við einfaldlega geymt gögnin í fylki.
- Nánar tiltekið, ef gildi x er í nóðu með lykil k látum við stak k í fylkinu bera gildið x.
- Okkar vantar bara leið til að ferðast í trénu.
- Við munum nýta okkur að:

Útfærslan á járninu

- Þar sem að lykarnir eru einræðir og ekki stærri en n þá getum við einfaldlega geymt gögnin í fylki.
- Nánar tiltekið, ef gildi x er í nóðu með lykil k látum við stak k í fylkinu bera gildið x.
- Okkar vantar bara leið til að ferðast í trénu.
- Við munum nýta okkur að:
    - Við getum dregið frá minnsta bitann til að fara upp tréð.

Útfærslan á járninu

- Þar sem að lykarnir eru einræðir og ekki stærri en n þá getum við einfaldlega geymt gögnin í fylki.
- Nánar tiltekið, ef gildi x er í nóðu með lykil k látum við stak k í fylkinu bera gildið x.
- Okkar vantar bara leið til að ferðast í trénu.
- Við munum nýta okkur að:
  - Við getum dregið frá minnsta bitann til að fara upp tréð.
  - Við getum lagt minnsta við bitann til að fara til *hægri* í trénu.

Trjáa rölt

- En hvernig hjálpar þetta okkur að leysa dæmið?

Trjáa rölt

- En hvernig hjálpar þetta okkur að leysa dæmið?
- Látum p tákna fylkið sem geymir gögnin okkar og f fylkið sem geymir tréð.

Trjáa rölt

- En hvernig hjálpar þetta okkur að leysa dæmið?
- Látum p tákna fylkið sem geymir gögnin okkar og f fylkið sem geymir tréð.
- Stak i í f geymir summmu þeirra staka frá $p[j]$ til $p[i]$, þar sem j er foreldri i í Fenwick-trénu.

Að finna summu fyrstu i stakanna

- Til að finna summu fyrstu i stakanna getum lagt saman þau stök í trénu sem svara til ásanna í bitaframsetningu i.

Að finna summu fyrstu i stakanna

- Til að finna summu fyrstu i stakanna getum lagt saman þau stök í trénu sem svara til ásanna í bitaframsetningu i.
- Tökum sem dæmi $i = 11$. Við vitum að $11 = 1011_2$ svo stökin sem við höfum áhuga á eru $1011_2 = 11$, $1010_2 = 10$ og $1000_2 = 8$.

Að finna summu fyrstu i stakanna

- Til að finna summu fyrstu i stakanna getum lagt saman þau stök í trénu sem svara til ásanna í bitaframsetningu i.
- Tökum sem dæmi $i = 11$. Við vitum að $11 = 1011_2$ svo stökin sem við höfum áhuga á eru $1011_2 = 11$, $1010_2 = 10$ og $1000_2 = 8$.
- Summa þessara þriggja staka í trénu er þá summu fyrstu 11 stakanna í fylkinu.

Að finna summu fyrstu i stakanna

- Til að finna summu fyrstu i stakanna getum lagt saman þau stök í trénu sem svara til ásanna í bitaframsetningu i.
- Tökum sem dæmi $i = 11$. Við vitum að $11 = 1011_2$ svo stökin sem við höfum áhuga á eru $1011_2 = 11$, $1010_2 = 10$ og $1000_2 = 8$.
- Summa þessara þriggja staka í trénu er þá summu fyrstu 11 stakanna í fylkinu.
- Takið eftir að til fá tölurnar drögum við ítrekað frá minnsta bita tölunnar í hvert skipti. Svo útfært er fallið

```c
#define LSB(i) ((i) & -(i))

int sum(int i)
{
    int sum = 0;
    while (i > 0)
    {
        sum = sum + a[i];
        i = i - LSB(i);
    }
    return sum;
}
```

Að uppfæra stak i

- Það nægir ekki að uppfæra stak i í trénu.

Að uppfæra stak i

- Það nægir ekki að uppfæra stak i í trénu.
- Tökum aftur sem dæmi 11. Þá þurfum við að uppfæra $1011_2 = 11$, $1100_2 = 12$ og $10000_2 = 16$.

Að uppfæra stak i

- Það nægir ekki að uppfæra stak i í trénu.
- Tökum aftur sem dæmi 11. Þá þurfum við að uppfæra $1011_2 = 11$, $1100_2 = 12$ og $10000_2 = 16$.
- Núna erum við að leggja minnsta bitann við tölunna í hvert skipti.

Að uppfæra stak i

- Það nægir ekki að uppfæra stak i í trénu.
- Tökum aftur sem dæmi 11. Þá þurfum við að uppfæra $1011_2 = 11$, $1100_2 = 12$ og $10000_2 = 16$.
- Núna erum við að leggja minnsta bitann við tölunna í hvert skipti.
- Við þyrftum svo einnig að leggja við öll önnur veldi af 2 hærri en 16 og minni eða jöfn heildarstærðinni.

Að uppfæra stak i

- Það nægir ekki að uppfæra stak i í trénu.
- Tökum aftur sem dæmi 11. Þá þurfum við að uppfæra $1011_2 = 11$, $1100_2 = 12$ og $10000_2 = 16$.
- Núna erum við að leggja minnsta bitann við tölunna í hvert skipti.
- Við þyrftum svo einnig að leggja við öll önnur veldi af 2 hærri en 16 og minni eða jöfn heildarstærðinni.
- Útfært er þetta

```c
#define LSB(i) ((i) & -(i))

void add(int i, int k)
{
    while (i < n)
    {
        a[i] = a[i] + k;
        i = i + LSB(i);
    }
}
```

Niðurstaða

- Fenwick-tré styðja ekki beint leitir að hlutbilssummum á forminu $[a, b]$.

Niðurstaða

- Fenwick-tré styðja ekki beint leitir að hlutbilssummum á forminu $[a, b]$.
- Við getum þó reiknað hlutbilssummur $[0, a - 1]$ og $[0, b]$ og mismunur þeirra gefur hlutbilssummu $[a, b]$.

Niðurstaða

- Fenwick-tré styðja ekki beint leitir að hlutbilssummum á forminu $[a, b]$.
- Við getum þó reiknað hlutbilssummur $[0, a-1]$ og $[0, b]$ og mismunur þeirra gefur hlutbilssummu $[a, b]$.
- Við getum nú uppfært og reiknað hlutbilssummur í $\mathcal{O}(\log n)$, sem er betra en rótarþáttunin.

Efnisyfirlit

Fylki í stærðfræði

- *Fylki* í stærðfræði er annað en *fylki* í tölvunarfræði.

Fylki í stærðfræði

- *Fylki* í stærðfræði er annað en *fylki* í tölvunarfræði.
- Í stærðfræði eru fylki eins og tvívíð tölvunarfræði-fylki.

Fylki í stærðfræði

- *Fylki* í stærðfræði er annað en *fylki* í tölvunarfræði.
- Í stærðfræði eru fylki eins og tvívíð tölvunarfræði-fylki.
- Við segjum að fylki sé $a \times b$ (lesið "a sinnum b") ef það hefur a línur og b dálka.

Fylki í stærðfræði

- *Fylki* í stærðfræði er annað en *fylki* í tölvunarfræði.
- Í stærðfræði eru fylki eins og tvívíð tölvunarfræði-fylki.
- Við segjum að fylki sé $a \times b$ (lesið "a sinnum b") ef það hefur a línur og b dálka.
- Dæmi um fylki er

$$\left(\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right) \text{ eða } \left[\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right]$$

Fylki í stærðfræði

- *Fylki* í stærðfræði er annað en *fylki* í tölvunarfræði.
- Í stærðfræði eru fylki eins og tvívíð tölvunarfræði-fylki.
- Við segjum að fylki sé $a \times b$ (lesið "a sinnum b") ef það hefur a línur og b dálka.
- Dæmi um fylki er

$$\left(\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right) \text{ eða } \left[\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right]$$

- Þetta fylki er 2×3.

Frægustu fylkin

- Mikilvægasta flykið sem við rekumst á er *einingarfylkið*

$$I := \left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right)$$

Frægustu fylkin

- Mikilvægasta flykið sem við rekumst á er *einingarfylkið*

$$I := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- Annað fylki sem mun mögulega nýtast ykkur seinna meira er *snúnigsfylkið*

$$\sigma(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Skölun og samlagning

- Ef a er rauntala og A er fylki, þá látum við aA vera fylkið sem er eins og A nema að a hefur verið margfaldið við sérhvert stak.

Skölun og samlagning

- Ef a er rauntala og A er fylki, þá látum við aA vera fylkið sem er eins og A nema að a hefur verið margfaldið við sérhvert stak.
- Ef A og B eru fylki sem eru bæði $a \times b$ þá getum við lagt þau saman. Summan $A + B$ er stakvís samlagning fylkjanna A og B.

Fylkjasamlagning í C

```
for (i = 0; i < a; i++)
{
    for (j = 0; j < b; j++)
    {
        r[i][j] = s[i][j] + t[i][j];
    }
}
```

Innfeldi

- Ef fylkið er $1 \times n$ ($n \times 1$) þá köllum við það *línuvigur* (*dálkvigur*) af vídd n.

Innfeldi

- Ef fylkið er $1 \times n$ ($n \times 1$) þá köllum við það *línuvigur* (*dálkvigur*) af vídd n.
- Ef $a = (a_1, a_2, ..., a_n)$ er línuvigur og $b = (b_1, b_2, ..., b_n)$ er dálkvigur þá köllum við

$$\sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + ... + a_n b_n$$

innfeldi vigranna og táknum það $a \cdot b$.

Margfeldi

- Ef A er $a \times b$ og B er $b \times c$ þá getum við skilgreint margföldun fyrir fylkin.

Margfeldi

- Ef A er $a \times b$ og B er $b \times c$ þá getum við skilgreint margföldun fyrir fylkin.
- Margfeldið AB verður þá $a \times c$.

Margfeldi

- Ef A er $a \times b$ og B er $b \times c$ þá getum við skilgreint margföldun fyrir fylkin.
- Margfeldið AB verður þá $a \times c$.
- Stakið í línu i og dálki j í AB fæst með því að reikna innfeldi línu i í A og dálk j í B.

Margfeldi

- Ef A er $a \times b$ og B er $b \times c$ þá getum við skilgreint margföldun fyrir fylkin.
- Margfeldið AB verður þá $a \times c$.
- Stakið í línu i og dálki j í AB fæst með því að reikna innfeldi línu i í A og dálk j í B.
- Nánar er

$$c_{ij} = \sum_{k \geq 1} a_{ik} b_{kj}, \quad \text{þar sem } c_{ij} \text{ er stakið í línu } i \text{ og dálki } j \text{ í } AB.$$

Margfeldi

- Ef A er $a \times b$ og B er $b \times c$ þá getum við skilgreint margföldun fyrir fylkin.
- Margfeldið AB verður þá $a \times c$.
- Stakið í línu i og dálki j í AB fæst með því að reikna innfeldi línu i í A og dálk j í B.
- Nánar er

$$c_{ij} = \sum_{k \geq 1} a_{ik} b_{kj}, \quad \text{þar sem } c_{ij} \text{ er stakið í línu } i \text{ og dálki } j \text{ í } AB.$$

- Dæmi um einfalda fylkjamargföldun er

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{pmatrix}$$

Fylkjamargföldun í C

```
for (i = 0; i < a; i++)
{
    for (j = 0; j < c; j++)
    {
        r[i][j] = 0;
        for (k = 0; k < b; k++)
        {
            r[i][j] += s[i][k]*t[k][j];
        }
    }
}
```

Andhverfur

- Ef A og B eru $n \times n$ og $AB = I$ þá segjum við að B sé *andhverfa* A, oft táknað A^{-1}.

Andhverfur

- Ef A og B eru $n \times n$ og $AB = I$ þá segjum við að B sé *andhverfa* A, oft táknað A^{-1}.
- Fyrir 2×2 fylki er til auðveld lokuð formúla fyrir andhverfuna, þ.e.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

Andhverfur

- Ef A og B eru $n \times n$ og $AB = I$ þá segjum við að B sé *andhverfa* A, oft táknað A^{-1}.
- Fyrir 2×2 fylki er til auðveld lokuð formúla fyrir andhverfuna, þ.e.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

- Ljóst er að einhver vandi verður ef $ad - bc = 0$.

Andhverfur

- Ef A og B eru $n \times n$ og $AB = I$ þá segjum við að B sé *andhverfa* A, oft táknað A^{-1}.
- Fyrir 2×2 fylki er til auðveld lokuð formúla fyrir andhverfuna, þ.e.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

- Ljóst er að einhver vandi verður ef $ad - bc = 0$.
- Ef $ad - bc = 0$ þá hefur fylkið enga andhverfu og við segjum að það sé *óandhverfanlegt*.

Andhverfur

- Ef A og B eru $n \times n$ og $AB = I$ þá segjum við að B sé *andhverfa* A, oft táknað A^{-1}.
- Fyrir 2×2 fylki er til auðveld lokuð formúla fyrir andhverfuna, þ.e.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

- Ljóst er að einhver vandi verður ef $ad - bc = 0$.
- Ef $ad - bc = 0$ þá hefur fylkið enga andhverfu og við segjum að það sé *óandhverfanlegt*.
- **Æfing:** Gangið úr skugga um að

$$\frac{1}{ad - bc} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Lausnir línulegra kerfa

- Fylki koma að mestum notum þegar við viljum leysa línuleg jöfnuhneppi.

Lausnir línulegra kerfa

- Fylki koma að mestum notum þegar við viljum leysa línuleg jöfnuhneppi.
- Tökum sem dæmi hneppið

$$
\begin{aligned}
x + 2y + 6z &= 1 \\
3x + 2y + 3z &= 2 \\
4x + 2y + z &= 3
\end{aligned}
$$

Lausnir línulegra kerfa

- Fylki koma að mestum notum þegar við viljum leysa línuleg jöfnuhneppi.
- Tökum sem dæmi hneppið

$$
\begin{aligned}
x + 2y + 6z &= 1 \\
3x + 2y + 3z &= 2 \\
4x + 2y + z &= 3
\end{aligned}
$$

- Við getum einnig táknað þetta hneppi með fylkjajöfnunni

$$
\begin{pmatrix} 1 & 2 & 6 \\ 3 & 2 & 3 \\ 4 & 2 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}
$$

- Kynnum nú einn örlítið breyttan ritháttt. Í stað fylkjajöfnunnar á glærunni hér á undann skrifum við stundum

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right).$$

Við köllum þetta fylki *aukið*.

Línuaðgerðir

- Kynnum nú einn örlítið breyttan rithátt. Í stað fylkjajöfnunnar á glærunni hér á undann skrifum við stundum

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right).$$

  Við köllum þetta fylki *aukið*.

- Við munum nú skilgreina þrjár *línuaðgerðir* á fylki.

Línuaðgerðir

- Kynnum nú einn örlítið breyttan rithátt. Í stað fylkjajöfnunnar á glærunni hér á undann skrifum við stundum

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right).$$

  Við köllum þetta fylki *aukið*.

- Við munum nú skilgreina þrjár *línuaðgerðir* á fylki.
  - Skölun línu með fasta (sem er ekki 0). (L_1)

Línuaðgerðir

- Kynnum nú einn örlítið breyttan rithátt. Í stað fylkjajöfnunnar á glærunni hér á undann skrifum við stundum

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right).$$

  Við köllum þetta fylki *aukið*.
- Við munum nú skilgreina þrjár *línuaðgerðir* á fylki.
  - Skölun línu með fasta (sem er ekki 0). (L_1)
  - Leggja eina línu við aðra línu. (L_2)

Línuaðgerðir

- Kynnum nú einn örlítið breyttan ritháttt. Í stað fylkjajöfnunnar á glærunni hér á undann skrifum við stundum

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right).$$

  Við köllum þetta fylki *aukið*.
- Við munum nú skilgreina þrjár *línuaðgerðir* á fylki.
  - Skölun línu með fasta (sem er ekki 0). (L_1)
  - Leggja eina línu við aðra línu. (L_2)
  - Skipta á tveimur línum. (L_3)

Línuaðgerðir

- Kynnum nú einn örlítið breyttan rithátt. Í stað fylkjajöfnunnar á glærunni hér á undann skrifum við stundum

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right).$$

  Við köllum þetta fylki *aukið*.
- Við munum nú skilgreina þrjár *línuaðgerðir* á fylki.
  - Skölun línu með fasta (sem er ekki 0). (L_1)
  - Leggja eina línu við aðra línu. (L_2)
  - Skipta á tveimur línum. (L_3)
- Lykilatriðið er að línuaðgerðirnar okkar breyta ekki línulega kerfinu okkar.

Að leysa línuleg kerfi

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

Að leysa línuleg kerfi

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right)$$

Að leysa línuleg kerfi

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & -4 & -15 & -1 \\ 4 & 2 & 1 & 3 \end{array}\right)$$

Að leysa línuleg kerfi

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & -4 & -15 & -1 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right)$$

Að leysa línuleg kerfi

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & -4 & -15 & -1 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right)$$

$$\overset{L_1}{\sim} \left(\begin{array}{ccc|c} 2 & 4 & 12 & 2 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right)$$

Að leysa línuleg kerfi

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & -4 & -15 & -1 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right)$$

$$\overset{L_1}{\sim} \left(\begin{array}{ccc|c} 2 & 4 & 12 & 2 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

Að leysa línuleg kerfi

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

$$
\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & -4 & -15 & -1 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right)
$$

$$
\overset{L_1}{\sim} \left(\begin{array}{ccc|c} 2 & 4 & 12 & 2 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right)
$$

Að leysa línuleg kerfi

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & -4 & -15 & -1 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right)$$

$$\overset{L_1}{\sim} \left(\begin{array}{ccc|c} 2 & 4 & 12 & 2 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

$$\overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & 0 & -2 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

Að leysa línuleg kerfi

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & -4 & -15 & -1 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right)$$

$$\overset{L_1}{\sim} \left(\begin{array}{ccc|c} 2 & 4 & 12 & 2 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

$$\overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & 0 & -2 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right) \overset{L_1}{\sim} \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & -4 & -15 & -1 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right)$$

$$\overset{L_1}{\sim} \left(\begin{array}{ccc|c} 2 & 4 & 12 & 2 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

$$\overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & 0 & -2 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right) \overset{L_1}{\sim} \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right) \overset{L_1}{\sim} \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

Að leysa línuleg kerfi

- Sjáum nú hvernig við getum beytt þessum línuaðgerðum til þess að leysa jöfnuhneppið okkar:

$$\left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 3 & 2 & 3 & 2 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & -4 & -15 & -1 \\ 4 & 2 & 1 & 3 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 1 & 2 & 6 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right)$$

$$\overset{L_1}{\sim} \left(\begin{array}{ccc|c} 2 & 4 & 12 & 2 \\ 0 & 4 & 15 & 1 \\ 0 & 6 & 23 & 1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 15 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right) \overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & -3 & 1 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

$$\overset{L_2}{\sim} \left(\begin{array}{ccc|c} 2 & 0 & 0 & -2 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right) \overset{L_1}{\sim} \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 4 & 0 & 16 \\ 0 & 0 & 1 & -1 \end{array}\right) \overset{L_1}{\sim} \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

- Svo $x = -1$, $y = 4$ og $z = -1$ leysir jöfnuhneppið okkar.

Gauss eyðing

- Ferlið sem ég var að ganga í gegnum er kallað *Gauss-eyðing* og er algengasta leiðin til að leysa jöfnuhneppi.

Gauss eyðing

- Ferlið sem ég var að ganga í gegnum er kallað *Gauss-eyðing* og er algengasta leiðin til að leysa jöfnuhneppi.
- Fyrir $n \times n$ fylki er Gauss-eyðing $\mathcal{O}(n^3)$.

Gauss-eðying í C++

```
#define rep(i,a,b) for (__typeof(a) i=(a); i<(b); ++i)
const double EPS = 1e-9;
template <class K> bool eq(K a, K b) { return a == b; }
template <> bool eq<double>(double a, double b) {
    return abs(a - b) < EPS; }
template <class T> struct matrix {
  int rows, cols, cnt; vector<T> data;
  inline T& at(int i, int j) { return data[i * cols + j]; }
  matrix(int r, int c) : rows(r), cols(c), cnt(r * c) {
    data.assign(cnt, T(0)); }
  matrix(const matrix& other) : rows(other.rows),
    cols(other.cols), cnt(other.cnt), data(other.data) { }
  T& operator()(int i, int j) { return at(i, j); }
  matrix<T> rref(T &det, int &rank) {
    matrix<T> mat(*this); det = T(1), rank = 0;
    for (int r = 0, c = 0; c < cols; c++) {
      int k = r;
      rep(i,k+1,rows) if (abs(mat(i,c)) > abs(mat(k,c))) k = i;
      if (k >= rows || eq<T>(mat(k, c), T(0))) continue;
      if (k != r) {
        det *= T(-1);
        rep(i,0,cols) swap(mat.at(k, i), mat.at(r, i));
      } det *= mat(r, r); rank++;
      T d = mat(r,c);
      rep(i,0,cols) mat(r, i) /= d;
      rep(i,0,rows) {
        T m = mat(i, c);
        if (i != r && !eq<T>(m, T(0)))
          rep(j,0,cols) mat(i, j) -= m * mat(r, j); } r++;
    } return mat; }
};
```