Biltré

Bergur Snorrason

11. febrúar 2021

- Gefinn er listi með n tölum.

- Gefinn er listi með n tölum.
- Næst koma q fyrirspurnir, þar sem hver er af einni af tveimur gerðum:

- 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.

- 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 $[i, j]$.

- 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 $[i, j]$.
- Einföld útfærlsa á þessum fyrirspurnum gefur okkur $\mathcal{O}(\quad)$ fyrir þá fyrri og $\mathcal{O}(\quad)$ fyrir þá seinni.

- 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 $[i, j]$.
- Einföld útfærlsa á þessum fyrirspurnum gefur okkur $\mathcal{O}(1)$ fyrir þá fyrri og $\mathcal{O}(\quad)$ fyrir þá seinni.

- 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 $[i, j]$.
- Einföld útfærlsa á þessum fyrirspurnum gefur okkur $\mathcal{O}(1)$ fyrir þá fyrri og $\mathcal{O}(n)$ fyrir þá seinni.

- 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 $[i, j]$.
- Einföld útfærlsa á þessum fyrirspurnum gefur okkur $\mathcal{O}(\,1\,)$ fyrir þá fyrri og $\mathcal{O}(\,n\,)$ fyrir þá seinni.
- Þar sem allar (eða langflestar) fyrirspurnir gætu verið af seinni gerðin yrði lausnin í heildin $\mathcal{O}(\quad)$.

- 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 $[i, j]$.
- Einföld útfærlsa á þessum fyrirspurnum gefur okkur $\mathcal{O}(1)$ fyrir þá fyrri og $\mathcal{O}(n)$ fyrir þá seinni.
- Þar sem allar (eða langflestar) fyrirspurnir gætu verið af seinni gerðin yrði lausnin í heildin $\mathcal{O}(qn)$.

- 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 $[i, j]$.
- Einföld útfærlsa á þessum fyrirspurnum gefur okkur $\mathcal{O}(\,1\,)$ fyrir þá fyrri og $\mathcal{O}(\,n\,)$ fyrir þá seinni.
- Þar sem allar (eða langflestar) fyrirspurnir gætu verið af seinni gerðin yrði lausnin í heildin $\mathcal{O}(qn)$.
- Það er þó hægt að leysa þetta dæmi hraðar.

- 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 $[i, j]$.
- Einföld útfærlsa á þessum fyrirspurnum gefur okkur $\mathcal{O}(1)$ fyrir þá fyrri og $\mathcal{O}(n)$ fyrir þá seinni.
- Þar sem allar (eða langflestar) fyrirspurnir gætu verið af seinni gerðin yrði lausnin í heildin $\mathcal{O}(qn)$.
- Það er þó hægt að leysa þetta dæmi hraðar.
- Algengt er að nota til þess *biltré* (e. *segment tree*).

Biltré

- Biltré er tvíundartré sem geymir svör við vissum fyrirspurnum af seinni gerðinni.

Biltré

- Biltré er tvíundartré sem geymir svör við vissum fyrirspurnum af seinni gerðinni.
- Rótin geymir svar við fyrirspurninni `1  n` og ef nóða geymir svarið við `i  j` þá geyma börn hennar svör við `i  m` og `m + 1  j`, þar sem m er miðja heiltölubilsins $[i, j]$.

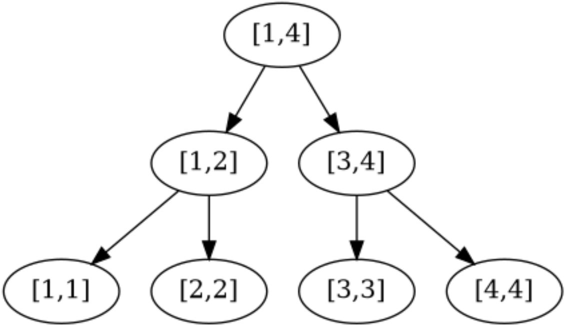
Biltré

- Biltré er tvíundartré sem geymir svör við vissum fyrirspurnum af seinni gerðinni.
- Rótin geymir svar við fyrirspurninni `1  n` og ef nóða geymir svarið við `i  j` þá geyma börn hennar svör við `i  m` og `m + 1  j`, þar sem m er miðja heiltölubilsins $[i, j]$.
- Þær nóður sem geyma svar við fyrirspurnum af gerðinni `i  i` eru lauf trésins.

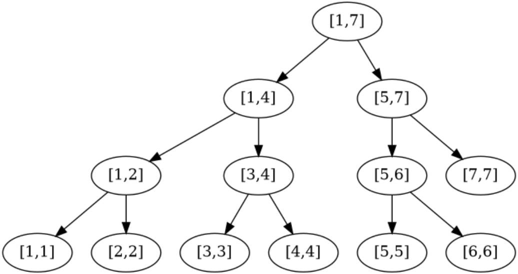
Biltré

- Biltré er tvíundartré sem geymir svör við vissum fyrirspurnum af seinni gerðinni.
- Rótin geymir svar við fyrirspurninni `1 n` og ef nóða geymir svarið við `i j` þá geyma börn hennar svör við `i m` og `m + 1 j`, þar sem m er miðja heiltölubilsins $[i, j]$.
- Þær nóður sem geyma svar við fyrirspurnum af gerðinni `i i` eru lauf trésins.
- Takið eftir að laufin geyma þá gildin í listanum og aðrar nóður geyma summu barna sinna.

Mynd af biltré, $n = 4$



Mynd af biltré, $n = 7$



Notkun biltrjáa

- Gerum ráð fyrir að við höfum biltré eins og lýst er að ofan og látum H tákna hæð trésins.

Notkun biltrjáa

- Gerum ráð fyrir að við höfum biltré eins og lýst er að ofan og látum H tákna hæð trésins.
- Hvernig getum við leyst fyrirspurnirnar á glærunni á undan, og hver er tímaflækjan?

Notkun biltrjáa

- Gerum ráð fyrir að við höfum biltré eins og lýst er að ofan og látum H tákna hæð trésins.
- Hvernig getum við leyst fyrirspurnirnar á glærunni á undan, og hver er tímaflækjan?
- Fyrri fyrirspurnin er einföld.

Notkun biltrjáa

- Gerum ráð fyrir að við höfum biltré eins og lýst er að ofan og látum H tákna hæð trésins.
- Hvernig getum við leyst fyrirspurnirnar á glærunni á undan, og hver er tímaflækjan?
- Fyrri fyrirspurnin er einföld.
- Ef við eigum að breyta i-ta stakinu í k finnum við fyrst laufið sem svarar til fyrirspurnar `i i`, setjum svarið þar sem k og förum svo upp í rót í gegnum foreldrin og uppfærum á leiðinni gildin í þeim nóðu sem við lendum í.

Notkun biltrjáa

- Gerum ráð fyrir að við höfum biltré eins og lýst er að ofan og látum H tákna hæð trésins.
- Hvernig getum við leyst fyrirspurnirnar á glærunni á undan, og hver er tímaflækjan?
- Fyrri fyrirspurnin er einföld.
- Ef við eigum að breyta i-ta stakinu í k finnum við fyrst laufið sem svarar til fyrirspurnar `i i`, setjum svarið þar sem k og förum svo upp í rót í gegnum foreldrin og uppfærum á leiðinni gildin í þeim nóðu sem við lendum í.
- Þar sem við heimsækjum bara þær nóður sem eru á veginum frá rót til laufs (mest H nóður) er tímaflækjan á fyrri fyrirspurninni $\mathcal{O}(\quad)$.

Notkun biltrjáa

- Gerum ráð fyrir að við höfum biltré eins og lýst er að ofan og látum H tákna hæð trésins.
- Hvernig getum við leyst fyrirspurnirnar á glærunni á undan, og hver er tímaflækjan?
- Fyrri fyrirspurnin er einföld.
- Ef við eigum að breyta i-ta stakinu í k finnum við fyrst laufið sem svarar til fyrirspurnar `i i`, setjum svarið þar sem k og förum svo upp í rót í gegnum foreldrin og uppfærum á leiðinni gildin í þeim nóðu sem við lendum í.
- Þar sem við heimsækjum bara þær nóður sem eru á veginum frá rót til laufs (mest H nóður) er tímaflækjan á fyrri fyrirspurninni $\mathcal{O}(H)$.

Biltré í C

```c

void update_rec(int i, int j, int x, int y, int e) // Hjálparfall.
{ // Við erum að leita að laufinu [x, x] og erum í [i, j].
    if (i == j) p[e] += y;
    else
    {
        int m = (i + j)/2;
        if (x <= m) update_rec(i, m, x, y, LEFT(e));
        else update_rec(m + 1, j, x, y, RIGHT(e));
        p[e] = p[LEFT(e)] + p[RIGHT(e)];
    }
}
void update(int x, int y)
{ // Bætum y við x-ta stakið.
    return update_rec(0, n - 1, x, y, 1);
```

- Seinni fyrirspurnin er ögn flóknari.

- Seinni fyrirspurnin er ögn flóknari.
- Auðveldast er að ímynda sér að við förum niður tréð og leitum að hvorum endapunktinum fyrir sig.

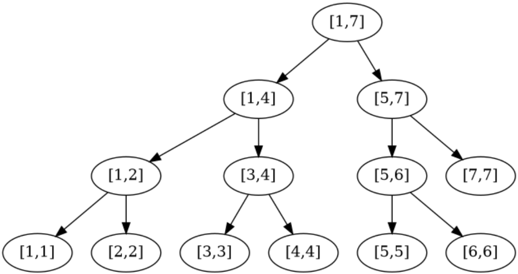
- Seinni fyrirspurnin er ögn flóknari.
- Auðveldast er að ímynda sér að við förum niður tréð og leitum að hvorum endapunktinum fyrir sig.
- Á leiðinni upp getum við svo pússlað saman svarinu, eftir því hvort við erum að skoða hægri eða vinstri endapunktinn.

- Seinni fyrirspurnin er ögn flóknari.
- Auðveldast er að ímynda sér að við förum niður tréð og leitum að hvorum endapunktinum fyrir sig.
- Á leiðinni upp getum við svo pússlað saman svarinu, eftir því hvort við erum að skoða hægri eða vinstri endapunktinn.
- Til dæmis, ef við erum að leita að vinstri endapunkti x og komum upp í bil $[i, j]$ þá bætum við gildinu í nóðu $[i, m]$ við það sem við höfum reiknað hingað til ef $x \in [m + 1, j]$, en annars bætum við engu við (því x er vinstri endapunkturinn).

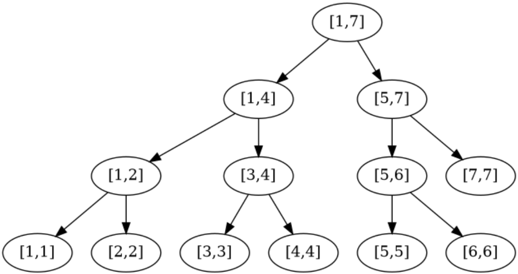
- Seinni fyrirspurnin er ögn flóknari.
- Auðveldast er að ímynda sér að við förum niður tréð og leitum að hvorum endapunktinum fyrir sig.
- Á leiðinni upp getum við svo púslað saman svarinu, eftir því hvort við erum að skoða hægri eða vinstri endapunktinn.
- Til dæmis, ef við erum að leita að vinstri endapunkti x og komum upp í bil $[i, j]$ þá bætum við gildinu í nóðu $[i, m]$ við það sem við höfum reiknað hingað til ef $x \in [m+1, j]$, en annars bætum við engu við (því x er vinstri endapunkturinn).
- Við göngum svona upp þar til við lendum í bili sem inniheldur hinn endapunktinn.

- Seinni fyrirspurnin er ögn flóknari.
- Auðveldast er að ímynda sér að við förum niður tréð og leitum að hvorum endapunktinum fyrir sig.
- Á leiðinni upp getum við svo pússlað saman svarinu, eftir því hvort við erum að skoða hægri eða vinstri endapunktinn.
- Til dæmis, ef við erum að leita að vinstri endapunkti x og komum upp í bil $[i, j]$ þá bætum við gildinu í nóðu $[i, m]$ við það sem við höfum reiknað hingað til ef $x \in [m+1, j]$, en annars bætum við engu við (því x er vinstri endapunkturinn).
- Við göngum svona upp þar til við lendum í bili sem inniheldur hinn endapunktinn.
- Með sömu rökum og áðan er tímaflækjan $\mathcal{O}(\quad)$.

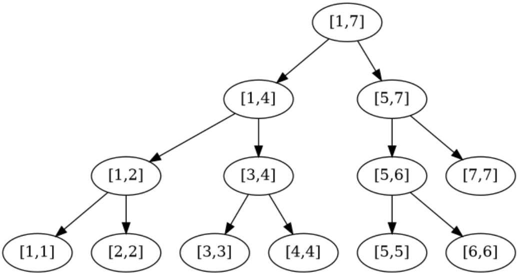
- Seinni fyrirspurnin er ögn flóknari.
- Auðveldast er að ímynda sér að við förum niður tréð og leitum að hvorum endapunktinum fyrir sig.
- Á leiðinni upp getum við svo pússlað saman svarinu, eftir því hvort við erum að skoða hægri eða vinstri endapunktinn.
- Til dæmis, ef við erum að leita að vinstri endapunkti x og komum upp í bil $[i, j]$ þá bætum við gildinu í nóðu $[i, m]$ við það sem við höfum reiknað hingað til ef $x \in [m + 1, j]$, en annars bætum við engu við (því x er vinstri endapunkturinn).
- Við göngum svona upp þar til við lendum í bili sem inniheldur hinn endapunktinn.
- Með sömu rökum og áðan er tímaflækjan $\mathcal{O}(\, H\,)$.



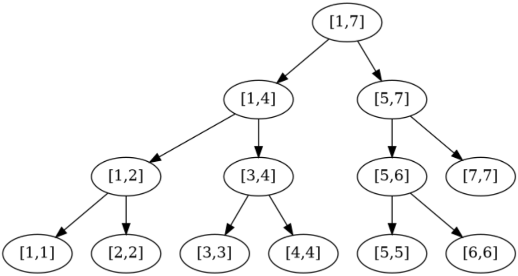
- Látum $f(i, j)$ tákna svar við fyrirspurninni `i j` og skoðum nokkur dæmi.



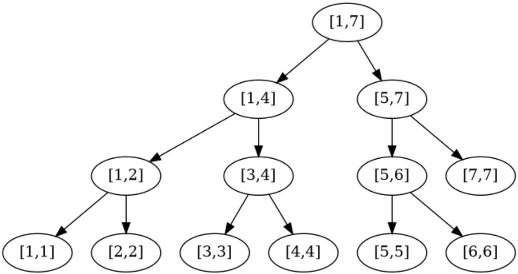
- Látum $f(i,j)$ tákna svar við fyrirspurninni `i j` og skoðum nokkur dæmi.
  - $f(1,3) = f(1,2) + f(3,3)$.



- Látum $f(i, j)$ tákna svar við fyrirspurninni `i j` og skoðum nokkur dæmi.
  - $f(1, 3) = f(1, 2) + f(3, 3)$.
  - $f(2, 5) = f(2, 2) + f(3, 4) + f(5, 5)$.



- Látum $f(i, j)$ tákna svar við fyrirspurninni `i j` og skoðum nokkur dæmi.
  - $f(1, 3) = f(1, 2) + f(3, 3)$.
  - $f(2, 5) = f(2, 2) + f(3, 4) + f(5, 5)$.
  - $f(1, 6) = f(1, 4) + f(5, 6)$.



- Látum $f(i,j)$ tákna svar við fyrirspurninni `i  j` og skoðum nokkur dæmi.
  - $f(1,3) = f(1,2) + f(3,3)$.
  - $f(2,5) = f(2,2) + f(3,4) + f(5,5)$.
  - $f(1,6) = f(1,4) + f(5,6)$.
  - $f(3,6) = f(3,4) + f(5,6)$.

Biltré í C

```c

int query_rec(int i, int j, int x, int y, int e) // Hjálparfall.
{ // Við erum að leita að bili [x, y] og erum í [i, j].
    if (x == i && y == j) return p[e];
    int m = (i + j)/2;
    if (x <= m && y <= m) return query_rec(i, m, x, y, LEFT(e));
    if (x > m && y > m) return query_rec(m + 1, j, x, y, RIGHT(e));
    return query_rec(i, m, x, m, LEFT(e))
           + query_rec(m + 1, j, m + 1, y, RIGHT(e));
}
int query(int x, int y)
{ // Finnum summuna yfir [x, y].
    return query_rec(0, n - 1, x, y, 1);
```

Tímaflækja biltrjáa

- Þar sem lengd hvers bils sem nóða svarar til helmingast þegar farið er niður tréð er $\mathcal{O}(H) = \mathcal{O}(\qquad)$.

Tímaflækja biltrjáa

- Þar sem lengd hvers bils sem nóða svarar til helmingast þegar farið er niður tréð er $\mathcal{O}(H) = \mathcal{O}(\log n)$.

Tímaflækja biltrjáa

- Þar sem lengd hvers bils sem nóða svarar til helmingast þegar farið er niður tréð er $\mathcal{O}(H) = \mathcal{O}(\log n)$.
- Við erum því komin með lausn á upprunalega dæminu sem er $\mathcal{O}(\qquad\qquad)$.

Tímaflækja biltrjáa

- Þar sem lengd hvers bils sem nóða svarar til helmingast þegar farið er niður tréð er $\mathcal{O}(H) = \mathcal{O}(\log n)$.
- Við erum því komin með lausn á upprunalega dæminu sem er $\mathcal{O}(q \cdot \log n)$.

Tímaflækja biltrjáa

- Þar sem lengd hvers bils sem nóða svarar til helmingast þegar farið er niður tréð er $\mathcal{O}(H) = \mathcal{O}(\log n)$.
- Við erum því komin með lausn á upprunalega dæminu sem er $\mathcal{O}(q \cdot \log n)$.
- Þetta væri nógu hratt ef, til dæmis, $n = q = 10^6$.

Annað dæmi

- Fyrsta lína inntaksins inniheldur tvær tölur, n og m, báðar jákvæðar heiltölur minni en 10^5.

Annað dæmi

- Fyrsta lína inntaksins inniheldur tvær tölur, n og m, báðar jákvæðar heiltölur minni en 10^5.
- Næsta lína inniheldur n heiltölur, á milli -10^9 og 10^9.

Annað dæmi

- Fyrsta lína inntaksins inniheldur tvær tölur, n og m, báðar jákvæðar heiltölur minni en 10^5.
- Næsta lína inniheldur n heiltölur, á milli -10^9 og 10^9.
- Næstu m línur innihalda fyrirspurnir, af tveimur gerðum.

Annað dæmi

- Fyrsta lína inntaksins inniheldur tvær tölur, n og m, báðar jákvæðar heiltölur minni en 10^5.
- Næsta lína inniheldur n heiltölur, á milli -10^9 og 10^9.
- Næstu m línur innihalda fyrirspurnir, af tveimur gerðum.
- Fyrri gerðin hefst á 1 og inniheldur svo tvær tölur, x og y. Hér á að setja x-tu töluna sem y.

Annað dæmi

- Fyrsta lína inntaksins inniheldur tvær tölur, n og m, báðar jákvæðar heiltölur minni en 10^5.
- Næsta lína inniheldur n heiltölur, á milli -10^9 og 10^9.
- Næstu m línur innihalda fyrirspurnir, af tveimur gerðum.
- Fyrri gerðin hefst á 1 og inniheldur svo tvær tölur, x og y. Hér á að setja x-tu töluna sem y.
- Seinni gerðin hefst á 2 og inniheldur svo tvær tölu, x og y. Hér á að prenta út stærstu töluna á hlutbilinu $[x, y]$ í talnalistanum.

Annað dæmi

- Fyrsta lína inntaksins inniheldur tvær tölur, n og m, báðar jákvæðar heiltölur minni en 10^5.
- Næsta lína inniheldur n heiltölur, á milli -10^9 og 10^9.
- Næstu m línur innihalda fyrirspurnir, af tveimur gerðum.
- Fyrri gerðin hefst á 1 og inniheldur svo tvær tölur, x og y. Hér á að setja x-tu töluna sem y.
- Seinni gerðin hefst á 2 og inniheldur svo tvær tölu, x og y. Hér á að prenta út stærstu töluna á hlutbilinu $[x, y]$ í talnalistanum.
- Hvernig leysum við þetta?

- Við getum leyst þetta með biltrjám.

- Við getum leyst þetta með biltrjám.
- Í stað þess að láta nóður (sem eru ekki lauf) geyma summu barna sinna, þá geyma þær stærra stak barna sinna.

Lausn

```
{ // Við erum að leita að bili [x, y] og erum í [i, j].
    if (x == i && y == j) return p[e];
    int m = (i + j)/2;
    if (x <= m && y <= m) return query_rec(i, m, x, y, LEFT(e));
    if (x > m && y > m) return query_rec(m + 1, j, x, y, RIGHT(e));
    return max(query_rec(i, m, x, m, LEFT(e)),
               query_rec(m + 1, j, m + 1, y, RIGHT(e)));
}
int query(int x, int y)
{ // Finnum stærsta gildið á [x, y].
    return query_rec(0, n - 1, x, y, 1);
}

void update_rec(int i, int j, int x, int y, int e) // Hjálparfall.
{ // Við erum að leita að laufinu [x, x] og erum í [i, j].
    if (i == j) p[e] = y;
    else
    {
        int m = (i + j)/2;
        if (x <= m) update_rec(i, m, x, y, LEFT(e));
        else update_rec(m + 1, j, x, y, RIGHT(e));
        p[e] = max(p[LEFT(e)], p[RIGHT(e)]);
    }
}
void update(int x, int y)
{ // Látum x-ta stakið vera y.
    return update_rec(0, n - 1, x, y, 1);
}
```

- Leysa má ýmis dæmi af þessari gerð, með biltrjám.

- Leysa má ýmis dæmi af þessari gerð, með biltrjám.
- Þessi dæmi eru yfirleitt kölluð *punkt-uppfærlsur*, *bil-fyrirspurnir* (e. *point-update*, *range-query*).

- Leysa má ýmis dæmi af þessari gerð, með biltrjám.
- Þessi dæmi eru yfirleitt kölluð *punkt-uppfærlsur, bil-fyrirspurnir* (e. *point-update, range-query*).
- Algengt er að sýna næst hvernig nota megi biltré til að leysa *bil-uppfærslur, punkt-fyrirspurnir* (e. *range-update, point-query*).

- Leysa má ýmis dæmi af þessari gerð, með biltrjám.
- Þessi dæmi eru yfirleitt kölluð *punkt-uppfærlsur, bil-fyrirspurnir* (e. *point-update, range-query*).
- Algengt er að sýna næst hvernig nota megi biltré til að leysa *bil-uppfærslur, punkt-fyrirspurnir* (e. *range-update, point-query*).
- Þetta er, í grófum dráttum, gert með því að snúa trjánum við.

- Leysa má ýmis dæmi af þessari gerð, með biltrjám.
- Þessi dæmi eru yfirleitt kölluð *punkt-uppfærlsur, bil-fyrirspurnir* (e. *point-update, range-query*).
- Algengt er að sýna næst hvernig nota megi biltré til að leysa *bil-uppfærslur, punkt-fyrirspurnir* (e. *range-update, point-query*).
- Þetta er, í grófum dráttum, gert með því að snúa trjánum við.
- Við munum ekki skoða þetta.

- Leysa má ýmis dæmi af þessari gerð, með biltrjám.
- Þessi dæmi eru yfirleitt kölluð *punkt-uppfærlsur*, *bil-fyrirspurnir* (e. *point-update*, *range-query*).
- Algengt er að sýna næst hvernig nota megi biltré til að leysa *bil-uppfærslur*, *punkt-fyrirspurnir* (e. *range-update*, *point-query*).
- Þetta er, í grófum dráttum, gert með því að snúa trjánum við.
- Við munum ekki skoða þetta.
- Við tökum frekar fyrir *lygn biltré*.

- Leysa má ýmis dæmi af þessari gerð, með biltrjám.
- Þessi dæmi eru yfirleitt kölluð *punkt-uppfærlsur, bil-fyrirspurnir* (e. *point-update, range-query*).
- Algengt er að sýna næst hvernig nota megi biltré til að leysa *bil-uppfærslur, punkt-fyrirspurnir* (e. *range-update, point-query*).
- Þetta er, í grófum dráttum, gert með því að snúa trjánum við.
- Við munum ekki skoða þetta.
- Við tökum frekar fyrir *lygn biltré*.
- Þau leyfa okkur að leysa *bil-uppfærlsur, bil-fyrirspurnir* (e. *range-update, range-query*).

Lygn dreifing

- Sem beinagrind munum við nota biltrjáa utfærsluna sem við notuðum til að leysa fyrsta dæmið.

Lygn dreifing

- Sem beinagrind munum við nota biltrjáa utfærsluna sem við notuðum til að leysa fyrsta dæmið.
- Við munum nú láta fyrri fyrirspurnina, `i j k`, þýða "Bættu k við allar tölur á bilinu $[i, j]$".

Lygn dreifing

- Sem beinagrind munum við nota biltrjáa utfærsluna sem við notuðum til að leysa fyrsta dæmið.
- Við munum nú láta fyrri fyrirspurnina, `i j k`, þýða “Bættu k við allar tölur á bilinu $[i, j]$”.
- Uppfærslan er framkvæmd á svipaðan hátt og fyrirspurnirnar eru.

Lygn dreifing

- Sem beinagrind munum við nota biltrjáa utfærsluna sem við notuðum til að leysa fyrsta dæmið.
- Við munum nú láta fyrri fyrirspurnina, `i j k`, þýða “Bættu k við allar tölur á bilinu $[i, j]$”.
- Uppfærslan er framkvæmd á svipaðan hátt og fyrirspurnirnar eru.
- Við geymum í annari hrúgu þær uppfærslur sem við eigum eftir að framkvæma.

Lygn dreifing

- Sem beinagrind munum við nota biltrjáa utfærsluna sem við notuðum til að leysa fyrsta dæmið.
- Við munum nú láta fyrri fyrirspurnina, `i j k`, þýða “Bættu k við allar tölur á bilinu $[i, j]$”.
- Uppfærslan er framkvæmd á svipaðan hátt og fyrirspurnirnar eru.
- Við geymum í annari hrúgu þær uppfærslur sem við eigum eftir að framkvæma.
- Í hverri endurkvæmni (bæði uppfærlum og fyrirspurnum) dreifum við uppfærslunum í nóðunni niður á við.

Lygn dreifing

- Sem beinagrind munum við nota biltrjáa utfærsluna sem við notuðum til að leysa fyrsta dæmið.
- Við munum nú láta fyrri fyrirspurnina, `i j k`, þýða "Bættu k við allar tölur á bilinu $[i, j]$".
- Uppfærslan er framkvæmd á svipaðan hátt og fyrirspurnirnar eru.
- Við geymum í annari hrúgu þær uppfærslur sem við eigum eftir að framkvæma.
- Í hverri endurkvæmni (bæði uppfærlum og fyrirspurnum) dreifum við uppfærslunum í nóðunni niður á við.
- Þetta kallast *lygn drefing* (e. *lazy propagation*), því við framkvæmum hana bara þegar nauðsyn krefur.

Lygn dreifing

- Sem beinagrind munum við nota biltrjáa utfærsluna sem við notuðum til að leysa fyrsta dæmið.
- Við munum nú láta fyrri fyrirspurnina, `i j k`, þýða “Bættu k við allar tölur á bilinu $[i, j]$”.
- Uppfærslan er framkvæmd á svipaðan hátt og fyrirspurnirnar eru.
- Við geymum í annari hrúgu þær uppfærslur sem við eigum eftir að framkvæma.
- Í hverri endurkvæmni (bæði uppfærlum og fyrirspurnum) dreifum við uppfærslunum í nóðunni niður á við.
- Þetta kallast *lygn drefing* (e. *lazy propagation*), því við framkvæmum hana bara þegar nauðsyn krefur.
- Ef biltré hefur lygna dreifingu köllum við það *lygnt biltré* (e. *segment tree with lazy propagation*).

- Látum $i < j$ vera heiltölur þannig að bilið $[i, j]$ svara til nóðu í bilþréi og m vera miðpunkt heiltölu bilsins $[i, j]$.

- Látum $i < j$ vera heiltölur þannig að bilið $[i, j]$ svara til nóðu í bilþréi og m vera miðpunkt heiltölu bilsins $[i, j]$.
- Gerum ráð fyrir að við eigum eftir að framkvæm uppfærslu `i j k`.

- Látum $i < j$ vera heiltölur þannig að bilið $[i, j]$ svara til nóðu í bilþréi og m vera miðpunkt heiltölu bilsins $[i, j]$.
- Gerum ráð fyrir að við eigum eftir að framkvæm uppfærslu `i j k`.
- Næst þegar við köllum á `query_rec(i, j, ...)` eða `update_rec(i, j, ...)` þá munum við dreifa uppfærslunni `i j k`.

- Látum $i < j$ vera heiltölur þannig að bilið $[i, j]$ svara til nóðu í bilþréi og m vera miðpunkt heiltölu bilsins $[i, j]$.
- Gerum ráð fyrir að við eigum eftir að framkvæm uppfærslu `i j k`.
- Næst þegar við köllum á `query_rec(i, j, ...)` eða `update_rec(i, j, ...)` þá munum við dreifa uppfærslunni `i j k`.
- Eftir dreifinguna munum við ekki eiga eftir uppfærslu á bilinu $[i, j]$, en við munum eiga eftir uppfærslurnar `i m k` og `(m + 1) j k`.

- Látum $i < j$ vera heiltölur þannig að bilið $[i, j]$ svara til nóðu í bilþréi og m vera miðpunkt heiltölu bilsins $[i, j]$.
- Gerum ráð fyrir að við eigum eftir að framkvæm uppfærslu `i j k`.
- Næst þegar við köllum á `query_rec(i, j, ...)` eða `update_rec(i, j, ...)` þá munum við dreifa uppfærslunni `i j k`.
- Eftir dreifinguna munum við ekki eiga eftir uppfærslu á bilinu $[i, j]$, en við munum eiga eftir uppfærslurnar `i m k` og `(m + 1) j k`.
- Þegar við dreifum uppfærslunni `i i k` þá nægir að uppfæra tilheyrandi lauf í biltrénu.

- Áðan var sagt “laufin geyma þá gildin í listanum og aðrar nóður geyma summu barna sinna”.

- Áðan var sagt “laufin geyma þá gildin í listanum og aðrar nóður geyma summu barna sinna”.
- Þetta gildir ekki fyrir lygn biltré.

- Áðan var sagt “laufin geyma þá gildin í listanum og aðrar nóður geyma summu barna sinna”.
- Þetta gildir ekki fyrir lygn biltré.
- Nóður lygna biltrjáa þurfa að geyma summu barna sinna, ásamt því að geyma þá summu sem fengist eftir allar óframkvæmdar útfærslur afkomenda hennar.

- Áðan var sagt “laufin geyma þá gildin í listanum og aðrar nóður geyma summu barna sinna”.
- Þetta gildir ekki fyrir lygn biltré.
- Nóður lygna biltrjáa þurfa að geyma summu barna sinna, ásamt því að geyma þá summu sem fengist eftir allar óframkvæmdar útfærslur afkomenda hennar.
- Þegar við ferðumst í gegnum tréð til að finna hvert við eigum að setja uppfærsluna uppfærum við tréð jafn óðum.

- Áðan var sagt “laufin geyma þá gildin í listanum og aðrar nóður geyma summu barna sinna”.
- Þetta gildir ekki fyrir lygn biltré.
- Nóður lygna biltrjáa þurfa að geyma summu barna sinna, ásamt því að geyma þá summu sem fengist eftir allar óframkvæmdar útfærslur afkomenda hennar.
- Þegar við ferðumst í gegnum tréð til að finna hvert við eigum að setja uppfærsluna uppfærum við tréð jafn óðum.
- Til dæmis, ef við viljum framkvæma uppfærsluna `i j k` þá þurfum við að bæta $k \cdot (j - i - 1)$ við rót biltrésins, því rótin geymir summu allra stakana.

Lyngt biltré

```
{ // Fall sem dreifir lygnum uppfærslum niður um eina hæð.
    p[e] += (y - x + 1)*o[e];
    if (x != y) o[LEFT(e)] += o[e], o[RIGHT(e)] += o[e];
    o[e] = 0;
}
int query_rec(int i, int j, int x, int y, int e)
{ // Við erum að leita að bili [x, y] og erum í [i, j].
    prop(i, j, e);
    if (x == i && y == j) return p[e];
    int m = (i + j)/2;
    if (x <= m && y <= m) return query_rec(i, m, x, y, LEFT(e));
    else if (x > m && y > m) return query_rec(m + 1, j, x, y, RIGHT(e));
    return query_rec(i, m, x, m, LEFT(e))
        + query_rec(m + 1, j, m + 1, y, RIGHT(e));
}
int query(int x, int y)
{ // Finnum summuna yfir [x, y].
    return query_rec(0, n - 1, x, y, 1);
}
void update_rec(int i, int j, int x, int y, int z, int e)
{ // Við erum að leita að bili [x, y] og erum í [i, j].
    prop(i, j, e);
    if (x == i && y == j) { o[e] = z; return; }
    int m = (i + j)/2;
    p[e] += (y - x + 1)*z;
    if (x <= m && y <= m) update_rec(i, m, x, y, z, LEFT(e));
    else if (x > m && y > m) update_rec(m + 1, j, x, y, z, RIGHT(e));
    else update_rec(i, m, x, m, z, LEFT(e)),
        update_rec(m + 1, j, m + 1, y, z, RIGHT(e));
}
void update(int x, int y, int z)
{ // Bætum z við stökin á bilinu [x, y]
    update_rec(0, n - 1, x, y, z, 1);
}
```