Više o segmentnim stablima

Kosta Grujčić

Matematička gimnazija
Nedelja$^{4}_{\text{informatike}}$

29. mart 2018.

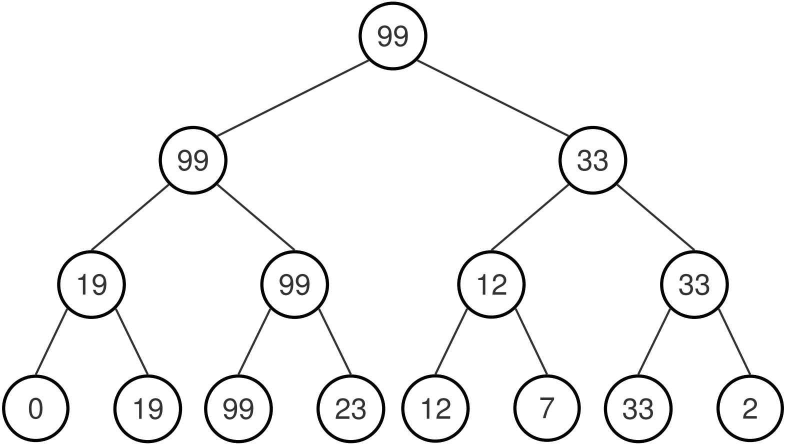
Motivacija

- Želimo da nad nizom podržimo efikasnu operaciju traženja najvećeg elementa na poziciji $[L, R]$ (QUERY)
- Želimo da podržimo efikasnu operaciju promene pojedinačnih elemenata (POINTWISE-UPDATE) ili elemenata na pozicijama $[L, R]$ (RANGE-UPDATE)

Uvod

- Segmentno stablo možemo posmatrati kao nadogradnju niza
- Bez umanjenja opštosti možemo pretpostaviti da je broj elemenata niza nad kojim gradimo stablo stepen broja 2
- U listovima se nalaze informacije pojedinačno za svaki element niza
- U svim ostalim čvorovima informacije se dobijaju kombinovanjem (MERGE) informacija levog i desnog deteta

Segmentno stablo



Slika: Segmentno stablo sa MAX operacijom

Analiza složenosti

Teorema

Prostorna složenost je $O(N)$ gde je N dužina niza kad kojim gradimo stablo.

Dokaz

N listova ima ukupno $\frac{N}{2}$ roditelja, jer svaki od tih čvorova ima po 2 različita lista za svoju decu. Slično, tih $\frac{N}{2}$ roditelja ima ukupno $\frac{N}{4}$ svojih roditelja. Nastavljajući na sličan način, zaključujemo da u stablu ima ukupno $N + \frac{N}{2} + \frac{N}{4} + ... + 1$ čvorova. Primetimo da važi

$$N + \frac{N}{2} + \frac{N}{4} + ... + 1 \leq \sum_{k=0}^{\infty} \frac{N}{2^k} = N \cdot \sum_{k=0}^{\infty} \frac{1}{2^k} = 2N.$$

Prema tome, prostorna složenost je $O(N)$. $\square$

Analiza složenosti

Teorema

Tokom QUERY *operacije ćemo posetiti najviše $2 \cdot \log N$ čvorova.*

Dokaz

Pretpostavimo da postoji nivo stabla L u kom postoje bar tri čvora koja će biti posećena. Označimo ih redom A, B i C. Tada na nivou $L - 1$ mora da postoji čvor M koji u svom intervalu obuhvata i čvor B (zbog polovljenja intervala) te se čvor B neće ni posetiti. Prema tome na svakom nivou može biti posećeno najviše 2 čvora, što je ukupno $2 \cdot \log N$ čvorova. Dakle, QUERY *zahteva $O(\log N)$ vremena.* □

Analiza složenosti

Teorema

Složenost POINTWISE-UPDATE *operacije je* $O(\log N)$.

Dokaz

Da bi promenili vrednost na poziciji P, *neophodno je da prođemo stablom duž puta koji vodi do lista zaduženog za* P, *jer su ti čvorovi odgovorni za intervale koji obuhvataju* P. *Kako je dubina stabla* $\log N$ *to je složenost ove operacije* $O(\log N)$. $\square$

Analiza složenosti

Teorema

Složenost RANGE-UPDATE *operacije je* $O(\log N)$.

Dokaz

Primenićemo lenju propagaciju. Pretpostavimo da se promena dešava na intervalu $[L, R]$. *Čvorovi zaduženi za taj interval su isti oni čvorovi koje koristimo prilikom* QUERY *operacije istog tog intervala. Zato čuvamo pomoćno segmentno stablo u kojem se nalaze akumulirane promene. Umesto da menjamo sve elemente na intervalu* $[L, R]$, *upamtićemo da se promena desila na tom intervalu u čvorovima koji pokrivaju taj interval (kojih ima* $2 \cdot \log N$*), a kada dođe do nekog upita dovoljno je da pogledamo šta se desilo na tom intervalu od prethodnih upita do sad. Prema tome, treba nam* $O(\log N)$ *vremena da bi izvršili neophodne promene.* □

Analiza složenosti

Teorema

Segmentno stablo se može inicijalizovati za $O(N)$ vremena.

Dokaz

Stablo ćemo inicijalizovati odozdo na gore, računajući vrednosti čvorova na svakoj dubini. U listovima se nalaze vrednosti samog niza. Za svaki sledeći čvor se njegova vrednost dobija kao veća od vrednosti dece, koje su već izračunate jer su na većoj dubini. Kako ćemo svaki čvor posetiti tačno jednom i izvršiti po $O(1)$ operacija, to je ukupna složenost $O(N)$. $\square$

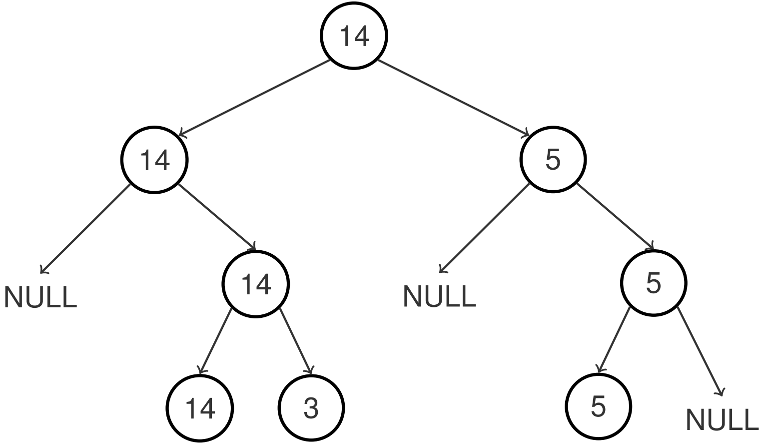
Uvod

- Pored računanja maksimalne vrednosti na intervalima, moguće je računati minimum, zbir, XOR, broj pojavljivanja nekog broja...
- Promene na intervalu mogu sve elemente postaviti na neku vrednost, svim elementima dodati neku vrednost, sve elemente pomnožiti nekim brojem...
- UPDATE i QUERY rade u $O(\log N)$ vremena
- Prostorna složenost je $O(N)$

Motivacija

- Šta ako je stablo *retko*?
- Šta ako želimo da pratimo promene tokom vremena?
- Može li se generalizovati na više dimenzija?
- Možemo li da imamo neku drugu strukturu podataka u čvoru samog stabla?

Implicitno segmentno stablo

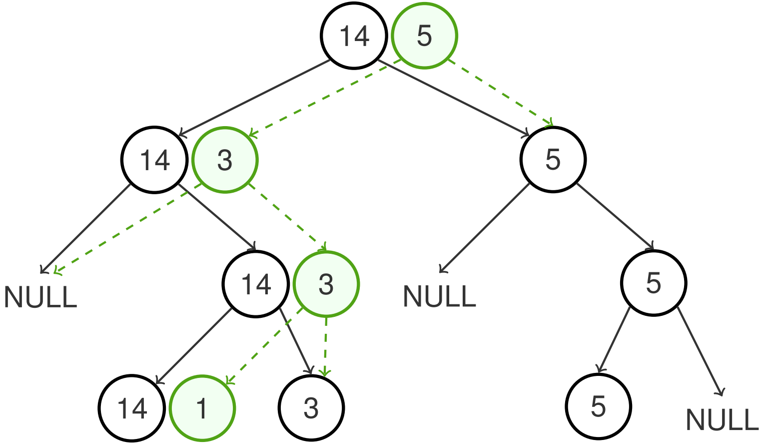


Slika: Implicitno segmentno stablo sa MAX operacijom

Implicitno segmentno stablo

- Pun potencijal pokazuje kada menja obično segmentno stablo koje je retko
- Pokazivači donekle komplikuju implementaciju
- Sada je mnogo važnije koliko se elemenata nalazi u stablu u svakom trenutku
- UPDATE i QUERY rade u $O(\log C)$ gde je C broj listova, odnosno veličina niza nad kojim gradimo stablo
- Prostorna složenost je $O(N \log C)$

Perzistentno segmentno stablo



Slika: Perzistentno segmentno stablo sa MAX operacijom

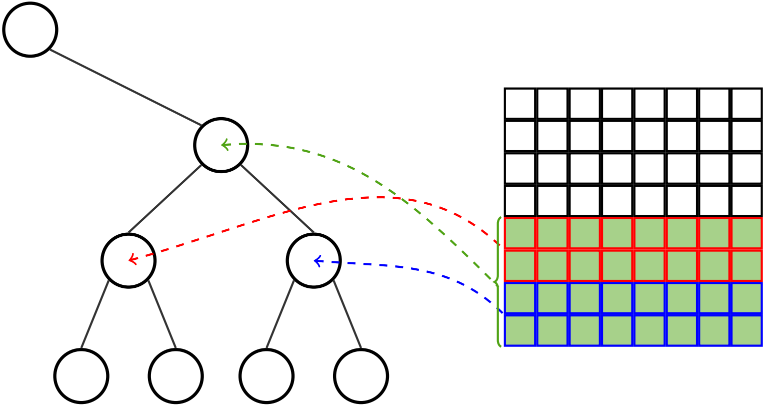
Perzistentno segmentno stablo

- Omogućava praćenje stanja stabla kroz vreme. Za svaki novi *trenutak* se kreira novi koren i duž jednog puta novi čvorovi
- Vreme se može shvatiti i kao pozicija u nizu, pa novi trenutak predstavlja samo sledeći indeks
- Kako je perzistentno segmentno stablo u osnovi implicitno, to su vremenske složenosti za UPDATE i QUERY $O(\log C)$, dok nam za čuvanje svih stabala treba $O(N \log C)$ prostora

Sa drugim strukturama

- U svakom čvoru segmentnom stabla se umesto samo jednog broja može čuvati bilo koja druga struktura podataka
- Najčešće su to strukture iz STL-a – stablo binarne pretrage (`set`, `map`), binarni hip (`priority_queue`) i dinamički niz (`vector`), ali i proizvoljno složene (segmentno stablo, Fenvikovo stablo, trip...)
- Složenosti operacija dodatnih struktura utiču na ukupnu složenost, pa bi npr. UPDATE u segmentnom stablu sa stablom binarne pretrage u svakom čvoru zahtevao $O(\log^2 N)$ vremena.

2D Segmentno stablo



2D Segmentno stablo

- U svakom čvoru segmentnog stabla se nalazi segmentno stablo koje je zaduženo za određenu podmatricu
- UPDATE i QUERY rade za $O(\log^2 N)$ vremena
- Prostorna složenost je $O(N^2)$
- Sličnim postupkom se može izgraditi i segmentno stablo proizvoljne dimenzije kao i implicitno i perzistentno

Sa drugim strukturama

- Postoji li broj X na poziciji $[L, R]$ u nizu ukoliko se pojedinačni elementi mogu menjati?
  hint:

Sa drugim strukturama

- Postoji li broj X na poziciji $[L, R]$ u nizu ukoliko se pojedinačni elementi mogu menjati?
  hint: `multiset`

Sa drugim strukturama

- Postoji li broj X na poziciji $[L, R]$ u nizu ukoliko se pojedinačni elementi mogu menjati?
  hint: `multiset`

- Kako napraviti *mergesort* stablo?
  hint:

Sa drugim strukturama

- Postoji li broj X na poziciji $[L, R]$ u nizu ukoliko se pojedinačni elementi mogu menjati?
  hint: `multiset`

- Kako napraviti *mergesort* stablo?
  hint: `vector`

Sa drugim strukturama

- Postoji li broj X na poziciji $[L, R]$ u nizu ukoliko se pojedinačni elementi mogu menjati?
  hint: `multiset`

- Kako napraviti *mergesort* stablo?
  hint: `vector`

- Kako naći zbir elemenata u podmatrici $[R_1, C_2, R_2, C_2]$ date matrice ukoliko se pojedinačni elementi mogu menjati?
  hint:

Sa drugim strukturama

- Postoji li broj X na poziciji $[L, R]$ u nizu ukoliko se pojedinačni elementi mogu menjati?
  hint: `multiset`
- Kako napraviti *mergesort* stablo?
  hint: `vector`
- Kako naći zbir elemenata u podmatrici $[R_1, C_2, R_2, C_2]$ date matrice ukoliko se pojedinačni elementi mogu menjati?
  hint: `2D segtree`

Neki zadaci

1. Za svaki upit oblika (L, R) ispisati indeks najvećeg elementa datog niza na poziciji $[L, R]$. Ukoliko ih ima više ispisati najmanji. Elementi se mogu menjati.

Neki zadaci

1. Za svaki upit oblika (L, R) ispisati indeks najvećeg elementa datog niza na poziciji $[L, R]$. Ukoliko ih ima više ispisati najmanji. Elementi se mogu menjati.

2. Za svaki upit oblika (A) ispisati sumu podstabla čvora A datog stabla. Elementi se mogu menjati,

Neki zadaci

1. Za svaki upit oblika (L, R) ispisati indeks najvećeg elementa datog niza na poziciji $[L, R]$. Ukoliko ih ima više ispisati najmanji. Elementi se mogu menjati.
2. Za svaki upit oblika (A) ispisati sumu podstabla čvora A datog stabla. Elementi se mogu menjati,
3. Za svaki upit oblika (L, R, K) ispisati k-ti broj na poziciji $[L, R]$ datog niza. Elementi se mogu menjati.

Neki zadaci

1. Za svaki upit oblika (L, R) ispisati indeks najvećeg elementa datog niza na poziciji $[L, R]$. Ukoliko ih ima više ispisati najmanji. Elementi se mogu menjati.
2. Za svaki upit oblika (A) ispisati sumu podstabla čvora A datog stabla. Elementi se mogu menjati,
3. Za svaki upit oblika (L, R, K) ispisati k-ti broj na poziciji $[L, R]$ datog niza. Elementi se mogu menjati.
4. Za svaki upit oblika (A, B, K) ispisati k-ti broj na putu od čvora A do čvora B u datom stablu. Vrednosti čvorova se mogu menjati.