DOIMIY MA’LUMOTLAR STRUKTURALARI

umidatoyirova4@gmail.com
samandar1530@gmail.com
TOYIROVA UMIDA
SAMANDAR TOVBAYEV XUSAN O`G`LI
MIRZO ULUG‘BEK NOMIDAGI MILLIY UNIVERSITETINING
JIZZAX FILIALI AXBOROT TIZIMLARI VA TEXNOLOGIYALARI YONALISHI TALABALARI

Annotatsiya: Ushbu maqolamizda doimiy ma’lumotlar strukturalari mavzusida bo’lib, u quyidagilar haqida ma’lumotlar berib o’tiladi. Ma'lumotlar tuzilmalarini doimiyga o'tkazish usullari, doimiy to'plam haqida qisqacha tushuncha, doimiy navbat va sreklar va shu bilan birgalikda doimiy ustuvor navbat hisoblanadi. Maqolamizda segment daraxt, saralash va shu bilan birgalikda pop, push metodlaridan foydalanib ular bilan birgalikda dasturlar bilan ishlashini keng yoritilgan.

Kalit so’zlar: Doimiy ma'lumotlar strukturasi, Qalin tugun usuli, Segmentlar daraxtini amalga oshirish, Doimiy to'plam, Doimiy burilish, Doimiy ustuvorlik navbati, extractMin – ekstraktiMin.

Doimiy ma'lumotlar strukturasi - har qanday o'zgartirishlar kiritilganda o'zining avvalgi holatini va ushbu holatlarga kirish huquqini saqlab qolgan ma'lumotlar tuzilmasi. To'liq doimiy ma'lumotlar tuzilmalarida siz nafaqat oxirgi, balki ma'lumotlar tuzilmalarining istalgan versiyasini o'zgartirishingiz mumkin, shuningdek, istalgan versiyaga so'rovlar qilishingiz mumkin.

Birlashtirilgan ma'lumotlar tuzilmalari ikkita ma'lumotlar tuzilmasini bittaga (birlashtirilishi mumkin bo'lgan qidiruv daraxtlari) birlashtirishga imkon beradi.

Funktsional ma'lumotlar tuzilmalari ta'rifi bo'yicha to'liq barqarordir, chunki ular halokatli topshiriqlarga ruxsat bermaydi, ya'ni. har qanday o'zgaruvchiga faqat bir marta qiymat berilishi mumkin va uni o'zgartirib bo'lmaydi. Agar ma'lumotlar strukturasi funksional bo'lsa, u holda u qo'shiladi, agar u qo'shilgan bo'lsa, u butunlay doimiy, agar u to'liq doimiy bo'lsa, u ham qisman doimiydir. Shu bilan birga, funktsional emas, balki birlashuvchi ma'lumotlar tuzilmalari mavjud.

Ma'lumotlar tuzilmalarini doimiyga o'tkazish usullari

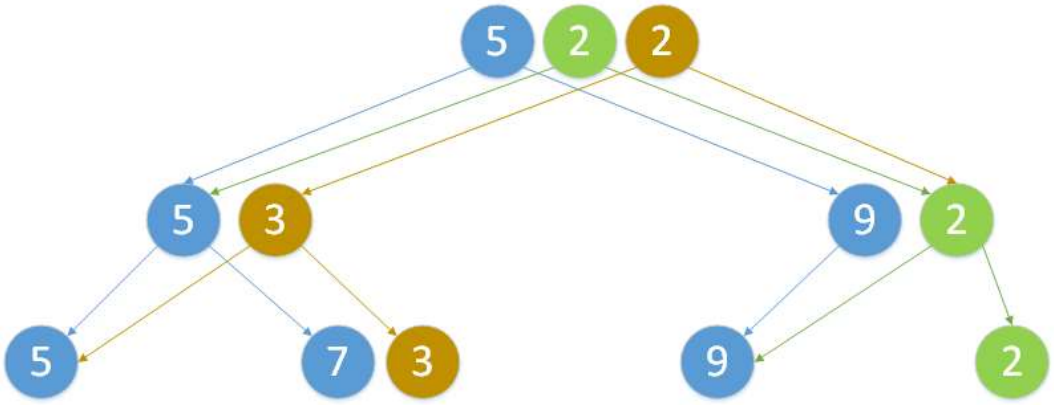
Har qanday tuzilmani barqaror qilishning bir necha yo'li mavjud:

- to'liq zaxira nusxasi (ingl. *full copy*) har qanday o'zgartirish operatsiyasi uchun ma'lumotlar strukturasi to'liq nusxalanganda, natijada yangi nusxa o'zgartiriladi,
- yuqoriga yo'l (ingliz. *Path copying*),

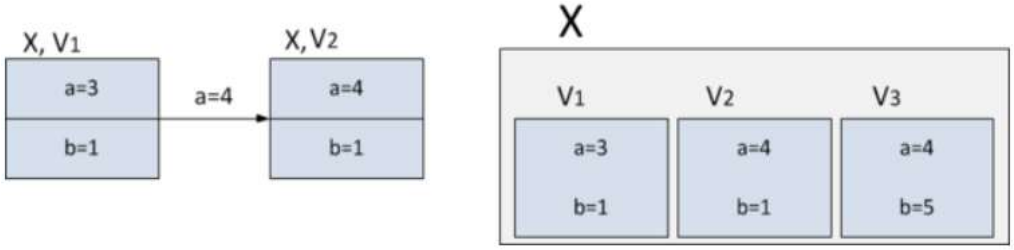
- *fat node* usuli .

Segmentlar daraxtini amalga oshirish - Segment daraxtiga asoslanib, siz to'liq doimiy ma'lumotlar strukturasini yaratishingiz mumkin.

Segmentlarning doimiy daraxtini amalga oshirish uchun daraxtning tuzilishini biroz o'zgartirish qulay. Buning uchun biz aniq mayoqlar foydalanish va bolalar uchun. Bundan tashqari, biz elementlari segmentlari daraxtining ildiziga ishora qiladigan massiv yaratamiz.



Qalin tugun usuli - Aytaylik, ma'lumotlar tuzilmasida siz o'zgartirish kiritishingiz kerak bo'lgan tugun mavjud (masalan, quyidagi rasmda, ma'lumotlar strukturasining birinchi versiyasida tugunda maydon mavjud, ikkinchi versiyada esa bu maydon). teng bo'lishi kerak, lekin ayni paytda siz tugunning eski elementiga kirishni saqlab qolishingiz kerak va vaqtni tejashga hojat yo'q. Bunday holda, siz tugunning ikkala elementini katta kombinatsiyalangan tugunda saqlashingiz mumkin.



Massivni amalga oshirish - Keling, stekni o'zgartiruvchi so'rovlar qatorini yarataylik.

```
struct Query:
  T value
  uint prev
```

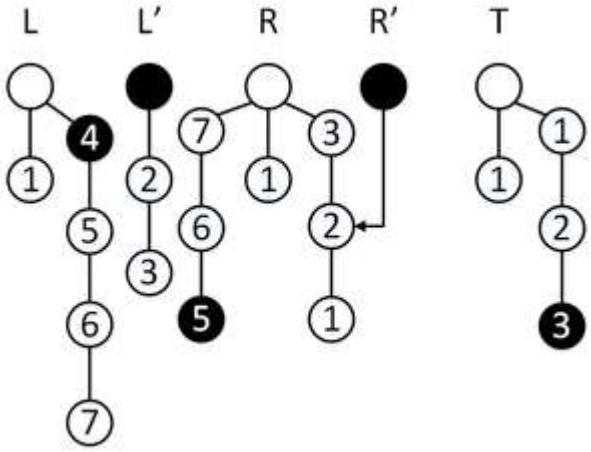
Biz qiymat va stekning oldingi elementiga havolaga ega bo'lgan tugundan foydalanamiz. Bunday holda, tugunning o'zi stekning elementidir.

```
struct Node:
 T value
 Node prev
```

Doimiy burilish – bularning barchasi qat'iylikni amalga oshiradi , ya'ni uning barcha oldingi versiyalariga kirish imkonini beradi. Quyida ko'rsatilgandek, funktsional qat'iylikni amalga oshirish mumkin, ya'ni bunday strukturadagi har bir xotira katakchasi bir marta ishga tushiriladi va kelajakda o'zgarmaydi.

Doimiy navbatni yaratish uchun uni steklarda amalga oshirishdan foydalanish juda qulay , chunki steklarni doimiy qilish oson va bu holda biz funktsional qat'iylikka erishamiz. Buning uchun ikkita stekda amalga oshirish mos emas, chunki eng yomon holatda bu vaqt talab etadi va shuning uchun qat'iylik holatida operatsiya uchun xotira. Keling, birinchi navbatda vaqti-vaqti bilan real vaqt rejimida navbat yaratish va keyin uni doimiy navbatga aylantirishni ko'rsatamiz.

Operatsiyani qilaylik **pop**, biz nusxa ko'chirish rejimidamiz, shuning uchun element olinmoqda **R', t o C o p y =2**. Uchta operatsiyada biz stekning uchta elementini nusxalash uchun vaqtimiz bor **L** stek ustida R.



Doimiy ustuvorlik navbati (Ing. *Persistent priority queue*) -- ustuvor navbat , qat'iylikni amalga oshiradi , ya'ni uning barcha oldingi elementlariga kirish imkonini beradi.

Segment daraxti bilan amalga oshirish

Ma'lumki, segment daraxti asosida to'liq doimiy ma'lumotlar strukturasi qurilishi mumkin . Shuning uchun biz navbat elementlarini kiritish va o'chirishda o'zgartirish operatsiyasini o'zgartirib, undan foydalanamiz.

Amalga oshirish

Biz aniq ko'rsatgichlarni bolalar elementlariga saqlaymiz. Keyin daraxt tugunining tuzilishi quyidagi shaklni oladi:

```
struct Node:
```

```
Node left
Node right
T key
Node(val : T)          // varaqlar uchun konstruktor
Node(leftChild : Node, rightChild : Node)
/* tugunlar uchun konstruktor (kalit maydoniga leftChild va rightChild orasida minimal miqdorni tayinlaydi)*/
```

Keling, quyidagi funktsiyalarni bajaraylik:

- build - daraxt qurish,
- insert - yangi navbat elementini kiritish,
- extractMin - navbatning minimal elementini olib tashlash,
- merge - birlashish.

Har bir funktsiya yangi daraxtning ildizini qaytaradi, keyin uni yana o'zgartirish mumkin. Amalga oshirishda siz daraxt ildizlarining alohida qatorini saqlashingiz mumkin va har bir elementga har bir yangi o'zgartirish kiritilganda, o'zgartirilganini massiv oxiriga qo'shishingiz mumkin.

Foydalanilgan adabiyolar:

1. M.O‘. ASHUROV, SH.A.SATTAROVA, SH.U.USMONQULOV, “ALGORITMLAR” , «Fan va texnologiya» nashriyoti, 2018.
2. Richard Bellman . Sayohatchi sotuvchi muammosini dinamik dasturlash bilan davolash // ACM jurnali . - 1962 .-- T. 9 . - S. 61–63.
3. Kazuo Ivama, Takuya Nakashima. TSP kub grafigi uchun takomillashtirilgan aniq algoritm // Proc. Hisoblash va kombinatorika bo‘yicha 13-yillik xalqaro konferensiya (COCOON 2007). - 2007. - T. 4598. - S. 108-117. - (Informatika fanidan ma'ruza matnlari).

Foydalanilgan internet saytlar:

1. https://neerc.ifmo.ru/wiki/index.php
2. http://ziyonet.uz/