



TOSHKENT AMALIY FANLAR UNIVERSITETI

Ma'lumotlar tuzilmasi va algoritmlar fani

“Kompyuter injiniring” kafedrası

Katta o'qituvchi Kendjayeva Dildora Xudayberganovna

B daraxtlar. B daraxt ta'rifi, B daraxtda amallar



MA'RUZA REJASI



B daraxt ta'rifi



B daraxt strukturasi



B daraxtda izlash algoritmi



B daraxtlarda element qo'shish

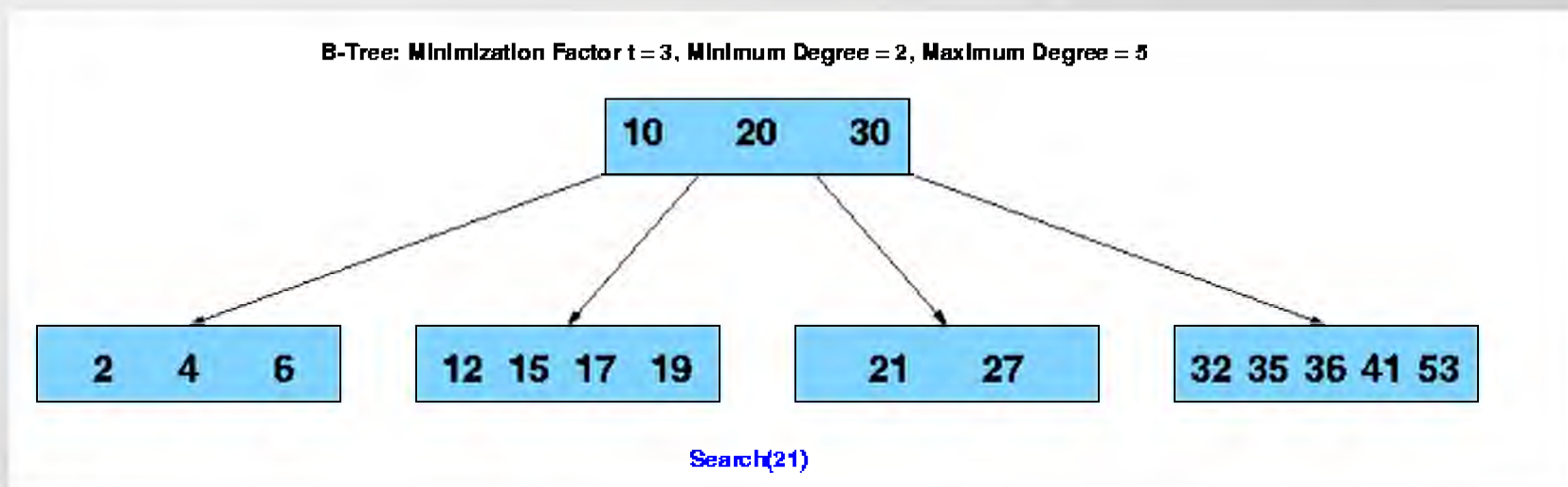


B daraxtda element o'chirish



B daraxt ta'rifi

- **B daraxti (inglizcha B-tree)** - izlash, qo'shish va o'chirish imkonini beradigan, juda ko'pshoxli muvozanatlashgan qidiruv daraxti. Tugunlari n bo'lgan B daraxti $O(\log n)$ balandlikka ega bo'ladi. Tugun shoxlari soni bittadan bir necha minggacha bo'lishi mumkin. B-daraxtlar $O(\log n)$ ko'p dinamik to'plam amallarini o'z vaqtida bajarish uchun ham ishlatilishi mumkin.
- B daraxti birinchi marta 1970-yilda R. Bayer va E. Makkreyt tomonidan taklif qilingan.





B daraxt strukturasi

B daraxti mukammal muvozanatlashgan, ya'ni uning barcha barglarining chuqurligi bir xil. B daraxti quyidagi xususiyatlarga ega (t - bu daraxt parametrlari, B daraxtining minimal darajasi deyiladi, 2 dan kam emas):

Ildizdan tashqari har bir tugun hech bo'lmaganda $t-1$ kalitni o'z ichiga oladi va har bir ichki tugun kamida avlodli t tugunlarga ega.

Har bir tugun, ildizdan tashqari, ichki tugunlarda ko'pi bilan $2t - 1$ kalitni va ko'pi bilan $2t$ avlodni o'z ichiga oladi

Ildizda daraxt bo'sh bo'lmasa bittadan $2t-1$ gacha kalit va balandligi 0 dan katta bo'lsa 2 dan $2t$ gacha avlodni o'z ichiga oladi.

Daraxtning har bir tugunida k_1, \dots, k_n kalitlari bo'lgan barglardan tashqari $n + 1$ avlodlari bor. i -avlodda $[k_{i-1}; k_i]$, $k_0 = -\infty$, $k_{n+1} = \infty$ da kesmaning kalitlari mavjud.

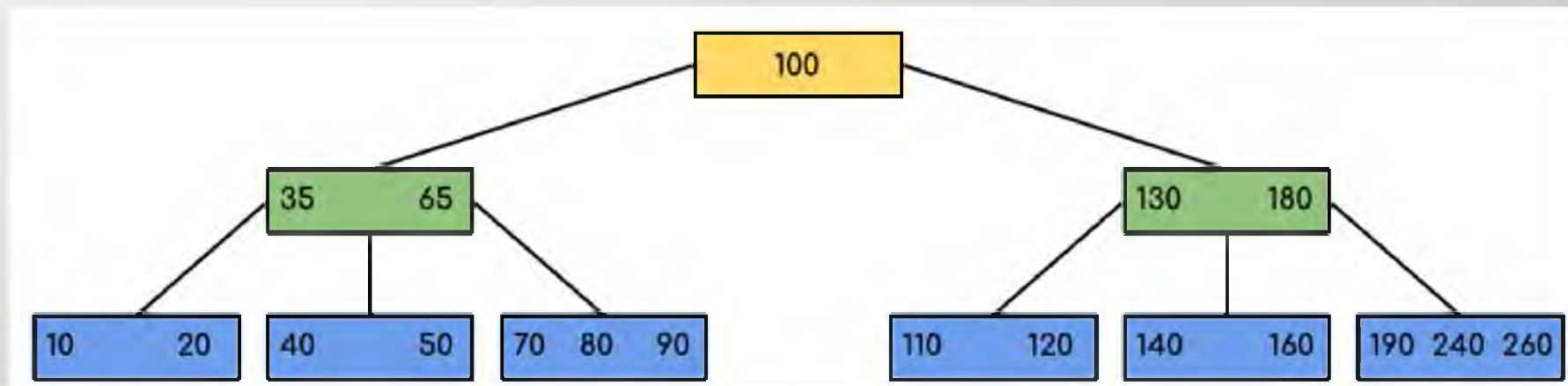
Har bir tugunning kalitlari kamaymaydigan tartibda tartiblangan.

Barcha barglar bir xil darajada.



B daraxt strukturasi

- B daraxtlar disklarda (fayl tizimlarida) yoki boshqa to'g'ridan-to'g'ri kiruvchi bo'lmagan saqlash muhitlarida, shuningdek, ma'lumotlar bazalarida foydalanish uchun mo'ljallangan. B daraxtlar qizil-qora daraxtlarga o'xshaydi, lekin ular diskni o'qish/yozish amallarini minimallashtirishda yaxshiroq hisoblanadi.
- B daraxtlari ham ***muvozanatlashgan daraxtlardir***, shuning uchun standart amallarni bajarish uchun vaqt balandlikka mutanosib. Ammo, boshqa daraxtlardan farqli o'laroq, ular maxsus disk xotirasi bilan ishlash uchun yaratilgan, aniqrog'i ular kirish -chiqish turidagi murojaatlarni kamaytiradi.





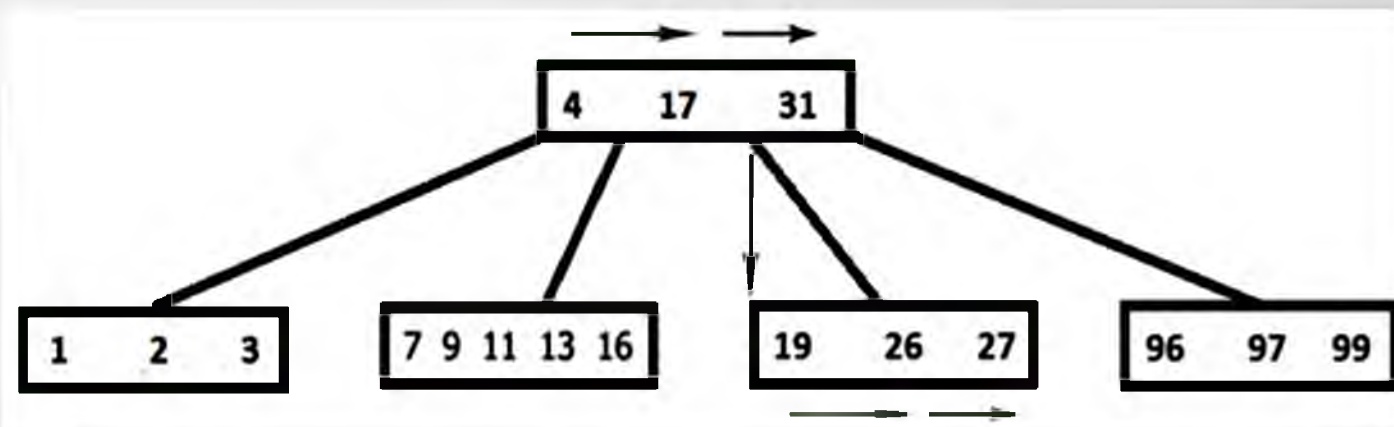
B daraxtda amallar

- ***Daraxtlar*** - bu dinamik to'plam amallarni bajaradigan ma'lumotlar tuzilmalari. Bunday amallar sifatida elementni **qidirish**, **minimal (maksimal) elementni qidirish**, **kiritish**, **o'chirish**, **avlod-ajdodga o'tish**, **avlodga o'tish** kabilarni keltirish mumkin.
- Shunday qilib, daraxt oddiy lug'at sifatida ham, ustivor navbat sifatida ham ishlatilishi mumkin.
- Daraxtlardagi asosiy amallar uning balandligiga mutanosib vaqtda bajariladi. Muvozanatlashgan daraxtlar ularning balandligini kamaytiradi.



B daraxtda izlash algoritmi

- B daraxtida izlash binar daraxtni qidirishga juda o'xshaydi, faqat bu yerda biz avlodga yo'lni 2 variantdan emas, balki bir nechta variantdan tanlashimiz kerak. Aks holda, farqi bo'lmay qoladi. Quyidagi rasmda 27-kalitni qidirish ko'rsatilgan. Tasvirni ko'rib chiqaylik (va shunga mos ravishda standart qidirish algoritmi):
- Biz ildiz kalitlarini kerak bo'lguncha o'tamiz. Bu holda 31 ga yetdik.
- Bu kalitning chap tomonidagi avlodga tushamiz.
- 27 dan kichik bo'lgunga qadar yangi tugunni kalit bo'yicha izlaymiz. Bunday holda, biz 27 ni topdik va to'xtadik.

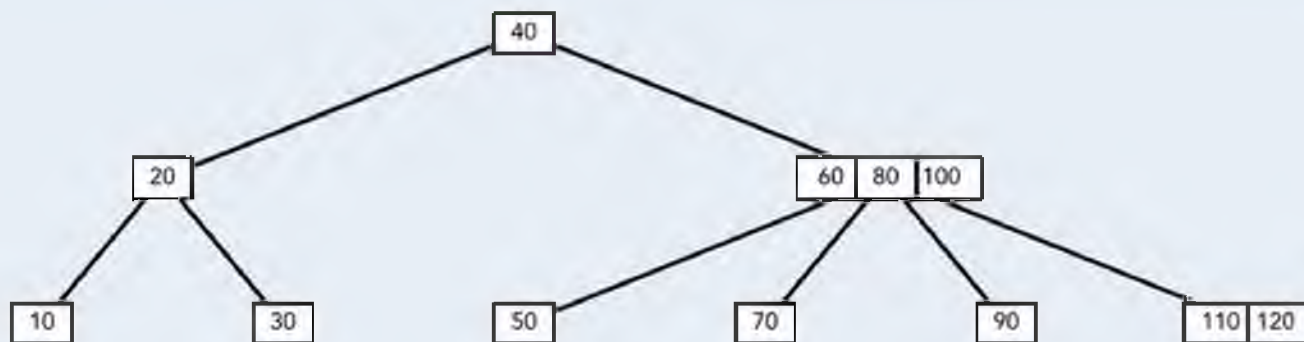


- Izlash amali $O(t \cdot \log t \ n)$ vaqtida bajariladi, bu yerda t - minimal daraja. Bu yerda disk amallarini faqat $O(\log t \ n)$ da bajarishimiz muhim qismidir.



B daraxtlarda element qo'shish

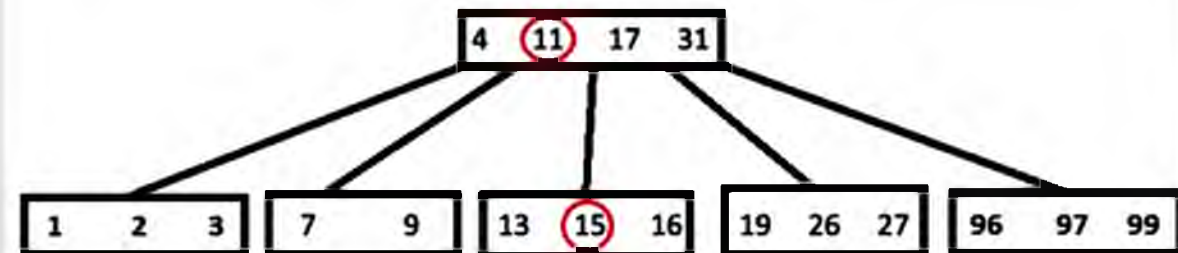
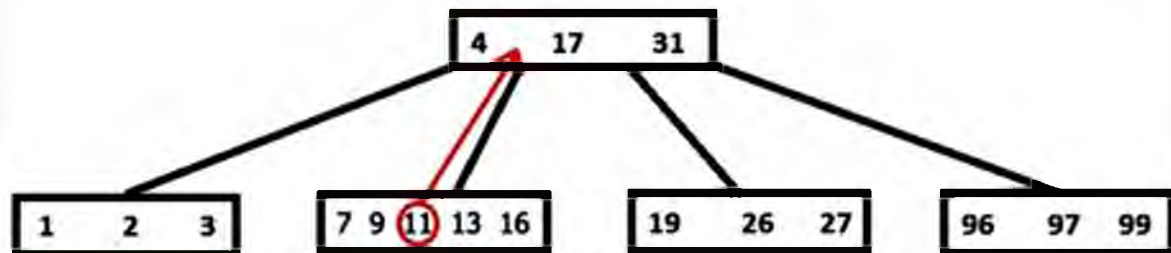
- Izlashdan farqli o'laroq, qo'shish usuli ikkilik daraxtga qaraganda ancha murakkab, chunki yangi barg yaratish va unga kalit qo'yish mumkin emas, chunki B daraxtining xususiyatlari buziladi. Kalitni allaqachon to'ldirilgan bargga kiritish mumkin emas.
- Tugunni ikkiga bo'lish amali kerak, agar barg to'ldirilgan bo'lsa, unda $2t-1$ kalit bor edi. 2 ga $t-1$ ga bo'lamiz undan kam kalitlar va oxirgi $t-1$ ajdod tuguniga o'tkaziladi. Shunga ko'ra, agar avlod-ajdod tuguni ham to'lgan bo'lsa, yana bo'lishimiz kerak va shunga o'xshash ildizgacha.
- Oddiy ikkilik daraxtlar singari, joylashtirish ham ildizdan barggacha bir o'tishda amalga oshiriladi. Har bir iteratsiyada o'tadigan barcha to'ldirilgan tugunlarni ajratamiz. Shunday qilib, agar natijada tugunni ajratish zarur bo'lsa, uning avlod-ajdodi to'ldirilmaganligiga ko'rishimiz mumkin.





B daraxtlarda element qo'shish

- Quyidagi rasmda xuddi o'sha daraxt qidirilmoqda ($t = 3$). Faqat hozir biz "15" kalitini qo'shamiz. Yangi kalitning o'rnini qidirib, tugallangan tugunga duch kelamiz (7, 9, 11, 13, 16).



- Algoritmga amal qilib, uni ajratdik - bu holda "11" avlod-ajdod tuguniga o'tadi va manba 2 ga bo'linadi. Keyin "15" tugmasi ikkinchi "ajratish" tuguniga kiritiladi. Bu holda B daraxtining barcha xususiyatlari saqlanib qolgan.
- Element qo'shish amali ham $O(t \log t n)$ vaqtda bajariladi. Shunga qaramay, biz diskdagi amallarni faqat $O(h)$ da bajaramiz, bu yerda h - daraxt balandligi.

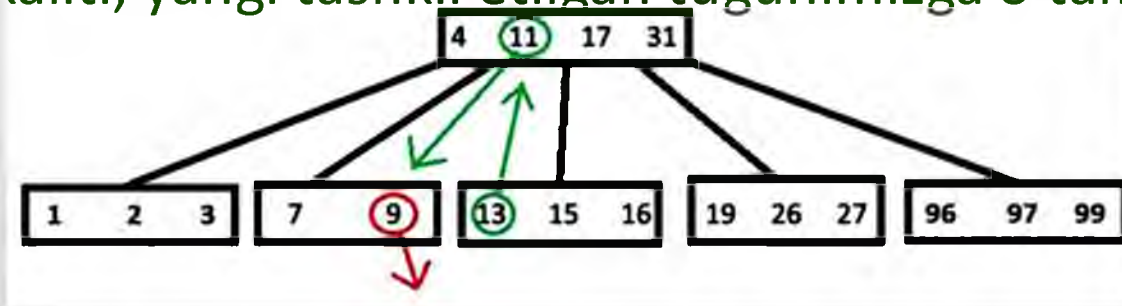


B daraxtda element o'chirish

Kalitni B daraxtidan olib tashlash, unga element qo'shishdan ham murakkab hisoblanadi. Buning sababi shundaki, ichki tugunni olib tashlash daraxtni umuman qayta tiklashni talab qiladi.

Agar bargdan o'chirish sodir bo'lsa, unda qancha kalit borligini tekshirish kerak. Agar $t-1$ dan ko'p bo'lsa, biz shunchaki o'chirib tashlaymiz va boshqa hech narsa qilishimiz shart emas.

Aks holda, agarda $t-1$ dan ortiq kalitlarni o'z ichiga oluvchi qo'shni barg bo'lsa, biz qo'shni tugunning qolgan kalitlari orasidagi ajratuvchi bo'lgan bu qo'shnidan kalitni tanlaymiz. Tushunarli bo'lishi uchun quyida "9" tugmasi o'chirilgan rasmda ko'rsatilgan. Agar bizning tugunning barcha qo'shnilarida $t-1$ kalitlari bo'lsa. Keyin biz uni qo'shnisi bilan birlashtiramiz, kerakli kalitni o'chirib tashlaymiz va bu ikkita "sobiq" qo'shnilar uchun ajratuvchi bo'lgan avlod-ajdod tugunining kaliti, yangi tashkil etilgan tugunimizga o'tamiz.





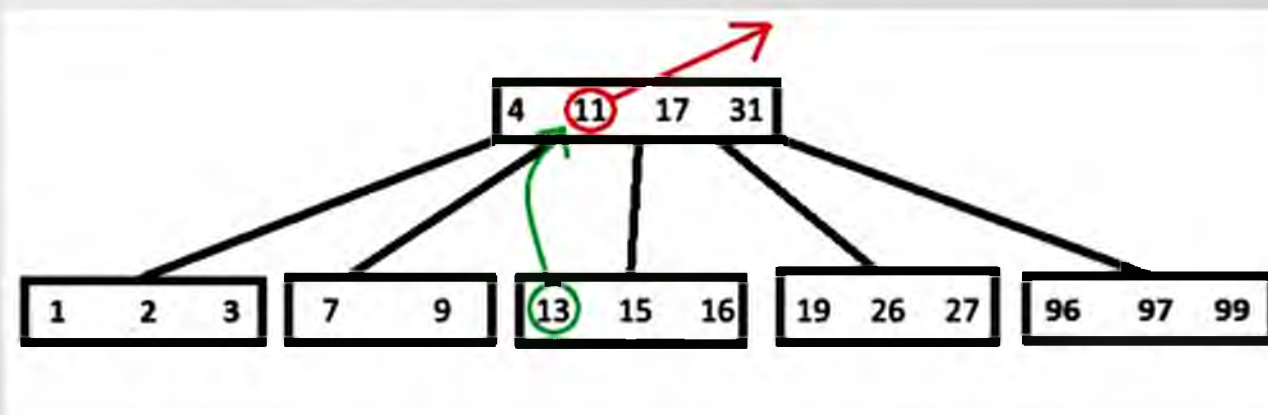
B daraxtda element o'chirish

Endi x ichki tugunidan k kalitini olib tashlashni ko'rib chiqaylik. Agar k kalitidan oldingi avlod tugunida $t-1$ dan ortiq kalitlar bo'lsa, biz k_1 ni topamiz - bu tugunning pastki daraxtida k . Uni o'chirib tashlaymiz. Manba tugundagi k ni k_1 bilan almashtiramiz.

Agar k kalitidan keyingi avlod tugunida $t-1$ dan ortiq kalit bo'lsa, biz ham xuddi shunday ishni bajaramiz. Agar ikkalasida ham $t-1$ kalitlari bo'lsa, biz bu avlodlarni birlashtiramiz, k -ni ularga o'tkazamiz, so'ngra k -ni yangi tugundan olib tashlaymiz.

Agar ildizning oxirgi 2 avlodi birlashsa, ular ildizga aylanadi va oldingi ildiz olib tashlanadi. Rasm quyida ko'rsatilgan, bu yerda "11" ildizdan chiqariladi.

O'chirish jarayoni $O(t \log t n)$ qo'shilishi bilan bir xil vaqtni oladi va disk operatsiyalari faqat $O(h)$ talab qilinadi, bu yerda h - daraxt balandligi.





Mavzu yuzasidan savollar:

1. B daraxt nima
2. B daraxtda izlash qanday amalga oshiriladi?
3. B daraxtda element o'chirish qanday amalga oshiriladi?
4. B daraxtda element qo'shish qanday amalga oshiriladi?
5. B-daraxt ma'lumotlar strukturasi qo'llaniladigan sohalarga qaysilar kiradi?



**Do you have
any questions?**

