1. B Daraxtlar

An’anaviy ikkilik qidiruv daraxtlarining cheklovlari sizni asabiylashtirishi mumkin. B-daraxti sizga osonlik bilan katta miqdordagi ma’lumotlarni ustida amallar bajarishga imkoniyat beradi. Katta hajmdagi ma’lumotlarni saqlash va qidirish haqida gap ketganda, an’anaviy ikkilik qidiruv daraxtlari yomon ishlashi va xotiradan yuqori foydalanish tufayli amaliy bo‘lmasligi mumkin. An’anaviy ikkilik qidiruv daraxtlaridan farqli o‘laroq, B daraxtlari bitta tugunda saqlashi mumkin bo‘lgan kalitlarning ko‘pligi bilan ajralib turadi, shuning uchun ular “katta kalit” daraxtlari sifatida ham tanilgan. B daraxtidagi har bir tugun bir nechta qiymatlarni o‘z ichiga olishi mumkin, bu daraxtning kattaroq dallanish omiliga va shu bilan sayozroq balandlikka ega bo‘lishiga imkon beradi. B daraxtlari, ayniqsa, qattiq disklar, flesh-xotira va CD-romlar kabi, katta hajmli ma’lumotlarga ega bo‘lgan saqlash tizimlari uchun juda mos keladi.

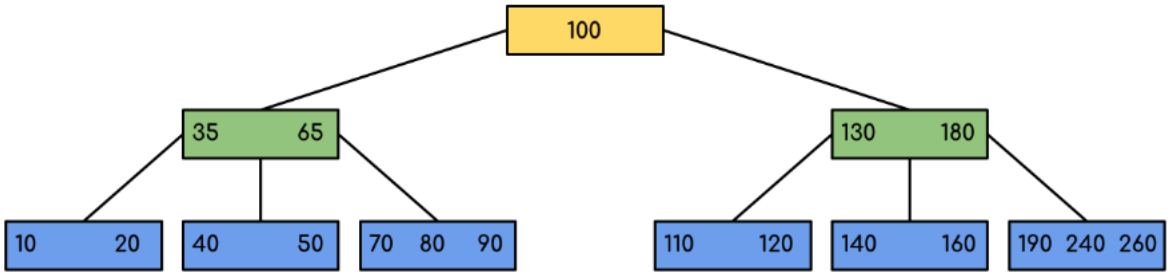
B-daraxtlar har bir tugunning minimal sonli kalitlarga ega bo‘lishini ta’minlash orqali muvozanatni saqlaydi, shuning uchun daraxt har doim muvozanatli bo‘ladi. Ushbu muvozanat daraxtning dastlabki shaklidan qat'i nazar, kiritish, o‘chirish va qidirish kabi operatsiyalar uchun vaqt murakkabligi har doim O(logn) bo‘lishini kafolatlaydi.

	Amallar	**Vaqt murakkabligi**
1.	Qidirish	O(logn)
2.	Kiritish	O(logn)
3.	O‘chirish	O(logn)

Eslatma: “n” - B daraxtidagi elementlarning umumiy soni

B daraxtining xususiyatlari:

- Barcha barglar bir xil darajada.
- B daraxti 't' minimum daraja atamasi bilan belgilanadi. 't' ning qiymati disk bloki hajmiga bog’liq.
- Ildizdan tashqari har bir tugunda kamida t-1 vorisilari bo‘lishi kerak. Ildizda kamida 1ta kalit bo‘lishi mumkin.
- Barcha tugunlarda (shu jumladan ildiz) ko‘pi bilan (2*t – 1) tugunlar bo‘lishi mumkin.
- Tugunning vorislari soni undagi kalitlarning soniga 1ni qo‘shganga teng.
- Tugunning barcha kalitlari ortib boruvchi tartibda saralanadi. K1 va k2 ikkita tugma orasidagi voris K1 va k2 oralig’idagi barcha tugunlarni o‘z ichiga oladi.
- B daraxti o‘sadi va ikkilik qidiruv daraxtiga o‘xshamaydigan ildizdan qisqaradi. Ikkilik qidiruv daraxtlari pastga qarab o‘sadi va pastga qarab qisqaradi.
- Boshqa muvozanatli ikkilik qidiruv daraxtlari singari, qidirish, kiritish va o‘chirish uchun vaqt murakkabligi O(log n).
- B daraxtiga tugunni kiritish faqat barg tugunida sodir bo‘ladi.



Rasm- 1 B daraxtining xususiyatlari

Yuqoridagi rasmda shuni ko‘rishimiz mumkinki, barcha barg tugunlari bir xil darajada va barcha ichki tugunlarning avlodlari tugun kalitidan bittaga ortiq.

B daraxtlari haqida qiziqarli ma’lumotlar:

B daraxtining minimal balandligi n tugunlar soni va m tugun bo‘lishi mumkin bo‘lgan maksimal vorislar soni:

$$h_{min} = [\log_m(n+1)] - 1$$

B daraxtining maksimal balandligi n tugunlar soni va t ildiz bo‘lmagan tugunga ega bo‘lishi mumkin bo‘lgan vorislarining minimal soni:

$$h_{max} \left[\log_t \frac{n+1}{2}\right] \text{ va } t = \frac{m}{2}$$

B daraxtida qidirish operatsiyasi:

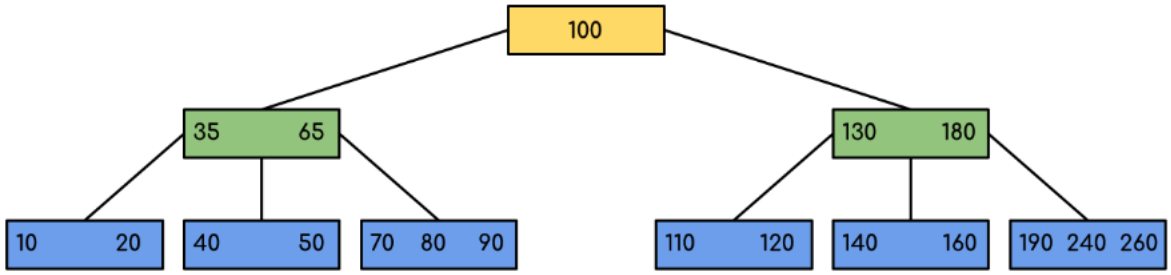
Qidiruv ikkilik qidiruv daraxtidagi qidiruvga o‘xshaydi. Qidiriladigan kalit k bo‘lsin.

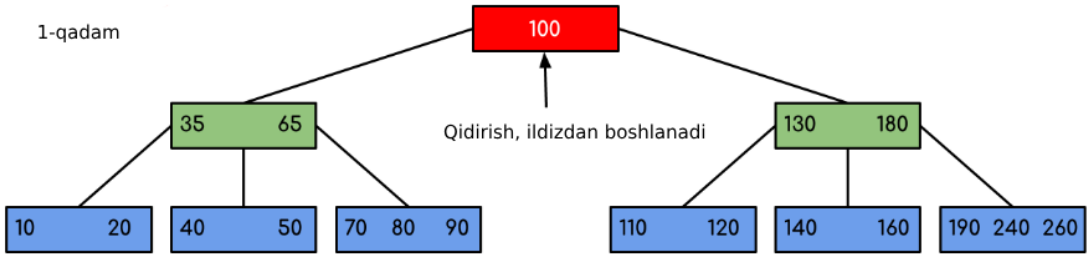
- Ildizdan boshlang va rekursiv ravishda pastga o‘ting.
  - Har bir tashrif buyurilgan barg bo‘lmagan tugun uchun, Agar tugunda kalit bo‘lsa, biz shunchaki tugunni qaytaramiz.
- Aks holda, biz tugunning tegishli vorisiga (birinchi katta kalitdan oldin bo‘lgan vorisga) qaytamiz.
- Agar biz barg tuguniga etib borsak va barg tugunida k ni topmasak, NULL ni qaytaring.

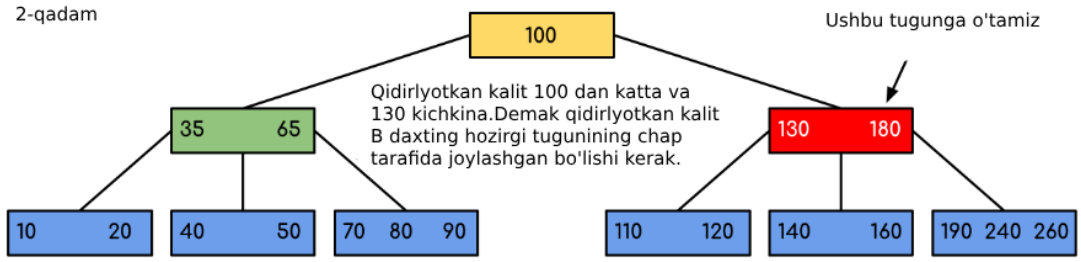
```
struct Node {
   int n;
   int key[MAX_KEYS];
   Node* child[MAX_CHILDREN];
   bool leaf;
};

Node* BtreeSearch(Node* x, int k) {
   int i = 0;
   while (i < x->n && k > x->key[i]) {
        i++;
   }
```

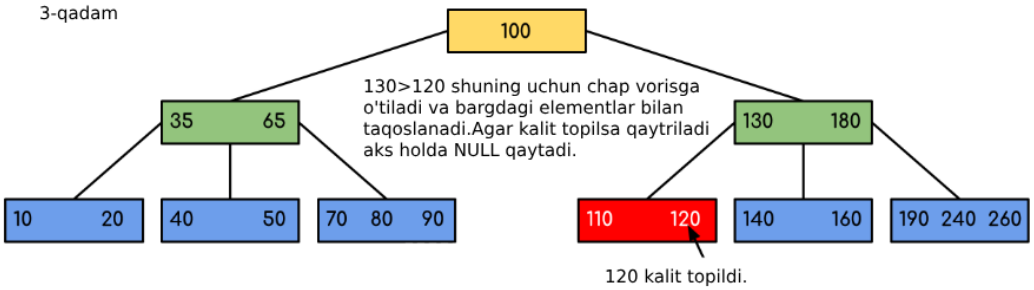
```
    if (i < x->n && k == x->key[i]) {
        return x;
    }
    if (x->leaf) {
        return nullptr;
    }
    return BtreeSearch(x->child[i], k);
}
```

Misol: Berilgan B daraxtidan 120 ni qidiring.









Ushbu misolda biz qidiruvimiz faqat qiymatni o‘z ichiga olgan kalit mavjud bo‘lish imkoniyatlarini cheklash orqali kamayganini ko‘rishimiz mumkin. Xuddi shunday, agar yuqoridagi misol ichida biz 180 ni qidirishimiz kerak bo‘lsa, boshqaruv 2-bosqichda to‘xtaydi, chunki dastur 180 kaliti joriy tugun ichida mavjudligini topadi va shunga o‘xshash, agar u 90 ni qidirish kerak bo‘lsa, u holda 90<100 sifatida u avtomatik ravishda chap pastki daraxtga o‘tadi va shuning uchun boshqaruv oqimi yuqoridagi misolda ko‘rsatilgandek davom etadi.

B daraxtlarining qo‘llanilishi:

- B daraxtlar diskda saqlangan ma’lumotlarga kirish uchun katta ma’lumotlar bazalarida ishlatiladi;
- Ma’lumotlar to‘plamida ma’lumotlarni qidirishga B daraxti yordamida ancha kam vaqt ichida erishish mumkin;
- Indekslash xususiyati bilan ko‘p darajali indekslashga erishish mumkin;
- Ko‘pgina serverlar B daraxti yondashuvidan ham foydalanadilar;
- B daraxtlari CAD (computer-aided design) tizimlarida geometrik ma’lumotlarni tartibga solish va qidirish uchun ishlatiladi;
- B daraxtlari kompyuter tarmoqlari va kriptografiya kabi boshqa sohalarda ham qo‘llaniladi;

B daraxtlarining afzalliklari:

- B daraxtlari qo‘shish, o‘chirish va qidirish kabi asosiy operatsiyalar uchun O(log n) vaqt murakkabligiga ega va bu ularni katta ma’lumotlar to‘plamlari va real vaqtda dasturlar uchun qulay tanlovga aylantiradi;
- B-daraxtlar o‘z-o‘zini muvozanatlash xususiyatiga ega;
- Yuqori muvofiqlik va yuqori o‘tkazuvchanlik;
- Samarali saqlash va foydalanish;

B daraxtlarining kamchiliklari:

- B daraxtlari diskka asoslangan ma’lumotlar tuzilmalarida xotira diskidan yuqori darajada foydalanishi mumkin;
- Barcha holatlar uchun eng yaxshisi emas;
- boshqa ma’lumotlar tuzilmalariga nisbatan sekin ishlaydi;

1.1 B daraxtiga tugunlarni joylashtirish

B daraxtiga elementni kiritish ikkita hodisadan iborat: elementni kiritish uchun tegishli tugunni qidirish va agar kerak bo‘lsa tugunni ajratish. Qo‘shish operatsiyasi har doim pastdan yuqoriga yondashuvda amalga oshiriladi.

1.Agar B daraxti bo‘sh bo‘lsa:

- Ildiz tugunini ajrating va kalitni joylashtiring.

2.Agar B daraxti bo‘sh bo‘lmasa:

a) Kiritish uchun mos tugunni toping.

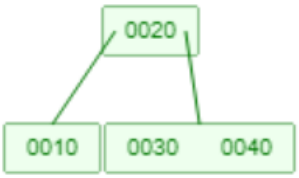
b) Agar tugun to‘liq bo‘lmasa:
   - Kalitni o‘sish tartibida joylashtiring.

c) Agar tugun to‘lgan bo‘lsa:
   - Tugunni medianada ajrating.
   - Median tugmachasini yuqoriga suring va chap tugmachalarni chap bola tuguniga, o‘ng tugmachalarni esa o‘ng bola tuguniga aylantiring.

B daraxti muvozanatli daraxtdir, ya’ni ildizdan barggacha bo‘lgan barcha yo‘llar bir xil uzunlikka ega. Daraxt minimal darajaga ega t, bu ildiz bo‘lmagan tugundagi

kalitlarning minimal soni. Har bir tugunda ko‘pi bilan 2t-1 tugmachalari va 2t bolalar bo‘lishi mumkin. Ildizda kamida bitta kalit va ko‘pi bilan 2t-1 tugmachalari bo‘lishi mumkin. Barcha ildiz bo‘lmagan tugunlarda kamida t-1 tugmachalari va ko‘pi bilan 2t-1 tugmachalari mavjud. Misol: Keling, 2 ga teng bo‘lgan "t" minimal darajadagi (2t-1=3) daraxtga namunasi 10, 20, 30, 40, 50, 60, 70, 80, va 90 va butun sonlar ketma-ketligi bilan algoritmni tushunaylik.

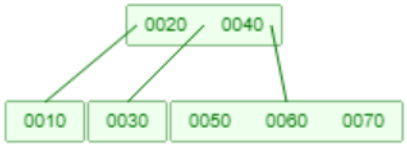
10 ,20,30 barchasi ildizga kiritilad. Chunki tugun sig’adigan kalitlarning maksimal soni 2*t – 1, ya’ni 3.



Ildiz tuguni to‘lganligi sababli, u avval ikkiga bo‘linadi, keyin 40 tegishli vorisga kiritiladi.

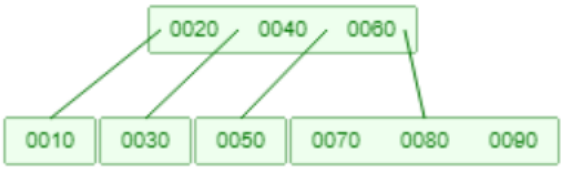
Keling, endi 50 ni joylashtiramiz. Ushbu yangi kalit tegishli bargga bo‘linmasdan kiritiladi.

Ildiz tuguni to‘lganligi sababli, u avval ikkiga bo‘linadi, keyin 60 tegishli vorisga kiritiladi.

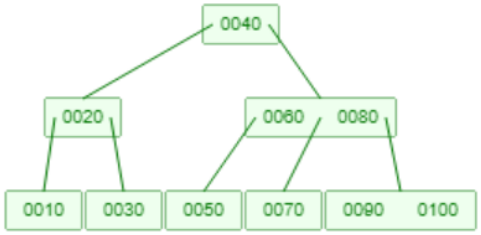


Endi 50 ni joylashtiramiz. Ushbu yangi kalit tegishli bargga bo‘linmasdan kiritiladi.

Ildiz tuguni to‘lganligi sababli, u avval ikkiga bo‘linadi, keyin 80 tegishli vorisga kiritiladi.

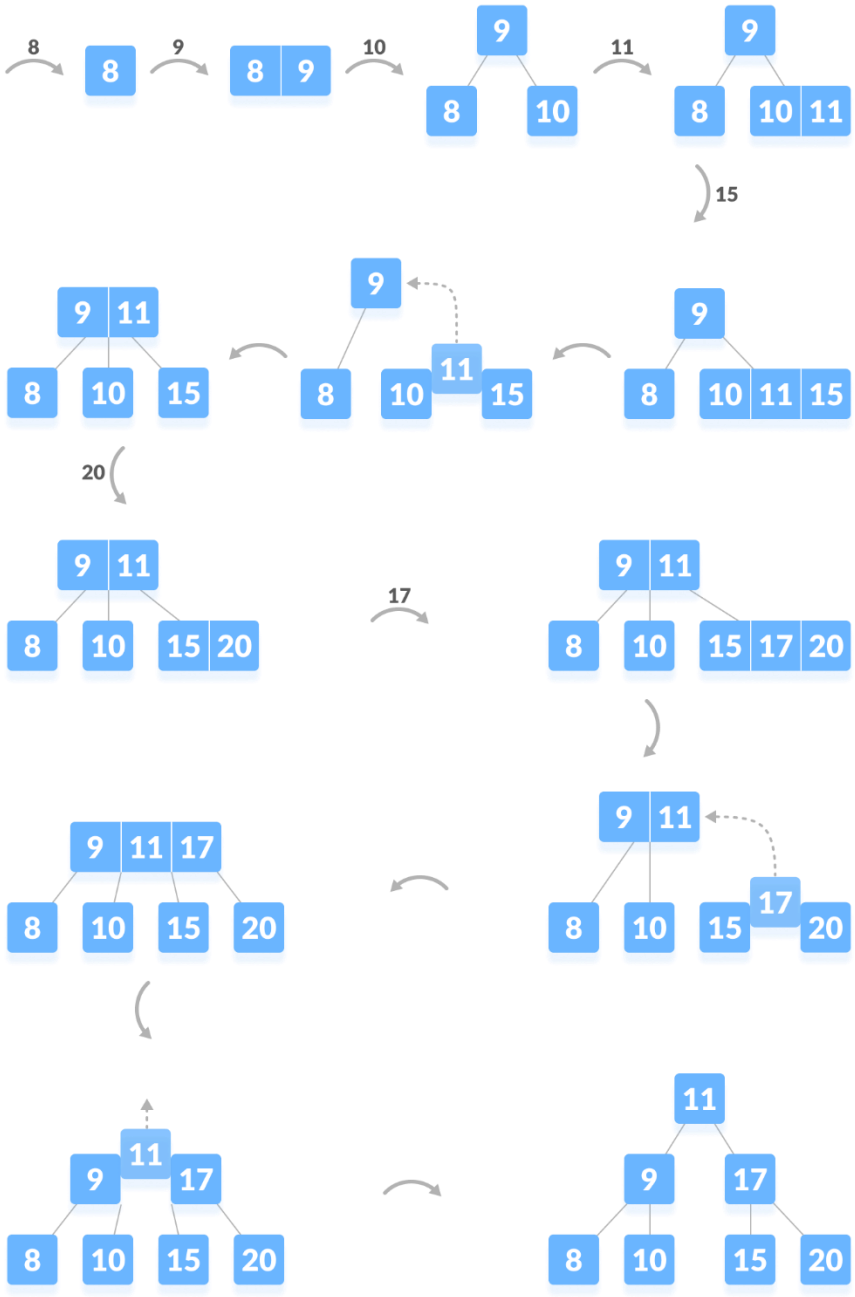


Endi 90 ni joylashtiramiz. Ushbu yangi kalit tegishli bargga bo‘linmasdan kiritiladi.



100 ni qo‘shish bo‘linishga olib keladi. O‘rta kalit ichki tugunga ko‘tariladi va ichki tugun to‘lganligi sababli u yana bo‘linadi. Shuningdek daraxt balandligining darajasi 3 ga teng bo‘ladi.

Misol: Kiritiladigan elementlar ketma-ketligi 8, 9, 10, 11, 15, 20, 17 va "t=" minimal darajadagi (2t-1=1) teng.



Rasm- 2 B daraxtka elementlar kiritish

1.2 B daraxtlarida elementlarni olib tashlash jarayoni

B daraxtidan olib tashlash kiritishdan ko‘ra qiyinroq, chunki biz kalitni faqat bargdan emas, balki istalgan tugundan olib tashlashimiz mumkin va kalitni ichki

tugundan olib tashlaganimizda, voris tugunlarining tartibini o‘zgartirishimiz kerak bo‘ladi.

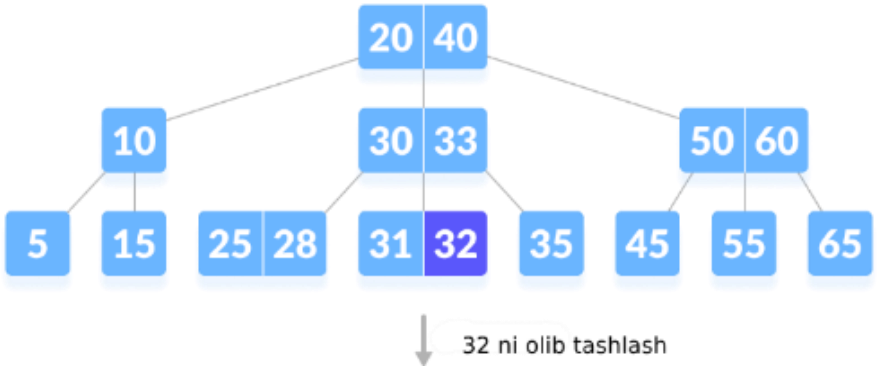
Kiritishda bo‘lgani kabi, olib tashlash B daraxtining xususiyatlarini buzmasligiga ishonch hosil qilishimiz kerak. Qo‘shish tufayli tugun juda katta bo‘lmasligiga ishonch hosil qilishimiz kerak bo‘lganidek, olib tashlash paytida tugun juda kichik bo‘lmasligiga ishonch hosil qilishimiz kerak (faqat ildiz katalogida minimal t-1 tugmalaridan kam bo‘lishi mumkin). Oddiy joylashtirish algoritmi zaxira nusxasini yaratishi kerak bo‘lganidek, agar kalit joylashtirilishi kerak bo‘lgan joyga boradigan tugun to‘ldirilgan bo‘lsa, oddiy o‘chirish usuli zaxira nusxasini yaratishi kerak bo‘lishi mumkin, agar tugun (ildizdan boshqa) joylashgan joyga boradigan bo‘lsa kalitni olib tashlash kerak, minimal miqdordagi kalitlarga ega.

B daraxtidagi elementni o‘chirish uchta asosiy hodisadan iborat: o‘chiriladigan kalit mavjud bo‘lgan tugunni qidirish, kalitni o‘chirish va agar kerak bo‘lsa daraxtni muvozanatlash. B daraxta kalitlarni olib tashlash paytida xatolik yuzaga kelishu mumkin.Bunda sabab tugunning minimum darajadagi kalitlar sonini saqlashi lozim.

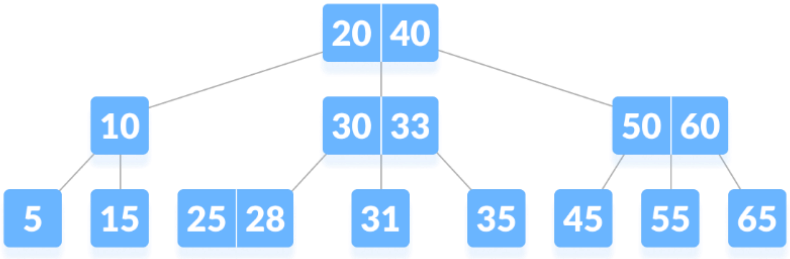
1-holat

O‘chiriladigan kalit bargda joylashgan bo‘lsa. Buning uchun ikkita holat mavjud.

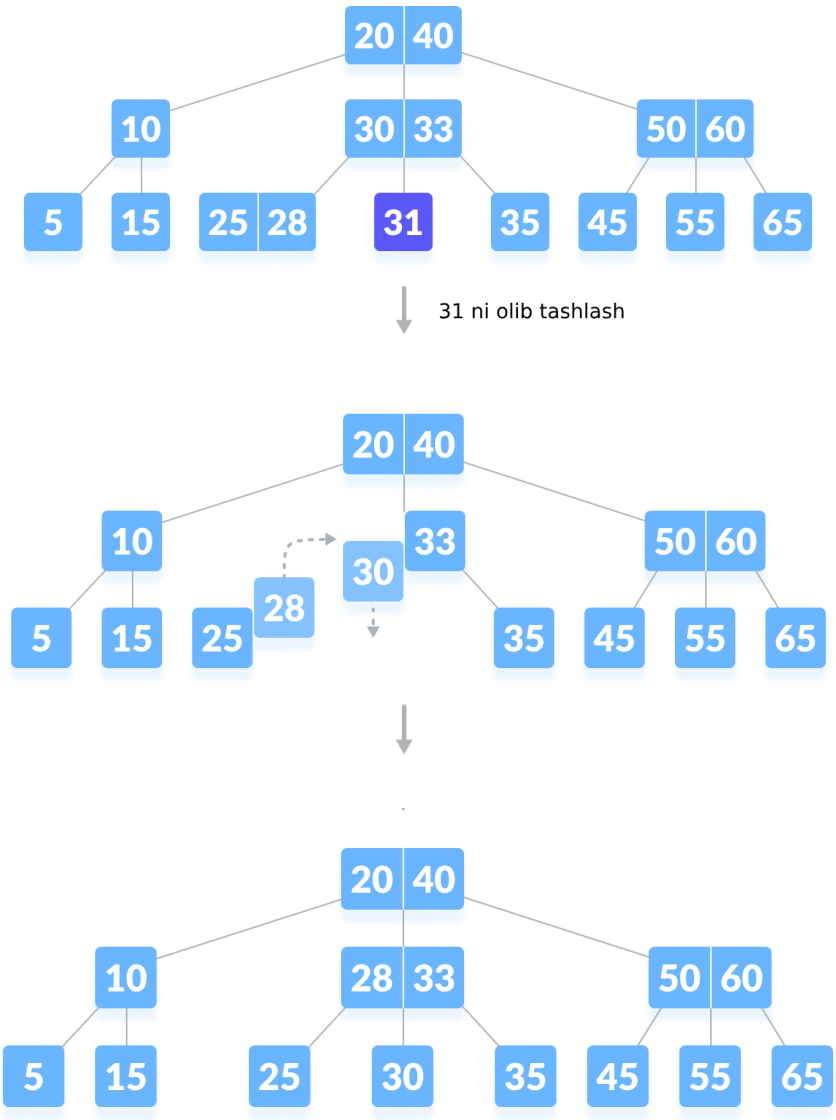
1. Kalitni o‘chirish tugun ushlab turishi kerak bo‘lgan minimal sonli kalitlarning xususiyatini buzmaydi.Quyidagi daraxtda 32 ni o‘chirish yuqoridagi xususiyatlarni buzmaydi.



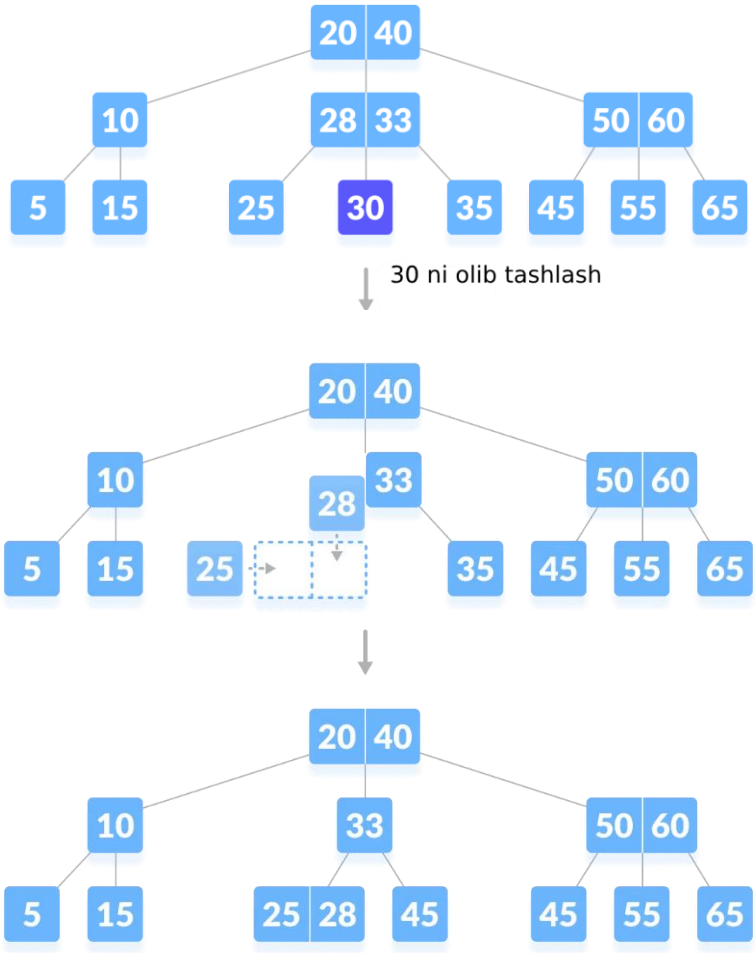
Rasm- 3 B daraxta elementlarni o‘chirish



2. Kalitni olib tashlash tugunni o‘z ichiga olishi kerak bo‘lgan minimal miqdordagi kalitlarning xususiyatini buzadi. Bunday holda, biz kalitni chapdan o‘ngga tartibda uning eng yaqin qo‘shni tugunidan olamiz. Birinchidan, biz eng yaqin chap tugunga tashrif buyuramiz. Agar chap voris tugunida minimal miqdordagi kalitlar bo‘lsa, unda ushbu tugundan kalitni olamiz. Aks holda, eng yaqin o‘ng voris tugunidan qarz olish mumkinligini tekshiramiz.Quyidagi daraxtda 31 ni olib tashlash yuqoridagi shartga olib keladi. Keling, kalitni chap yaqin tugundan olamiz.

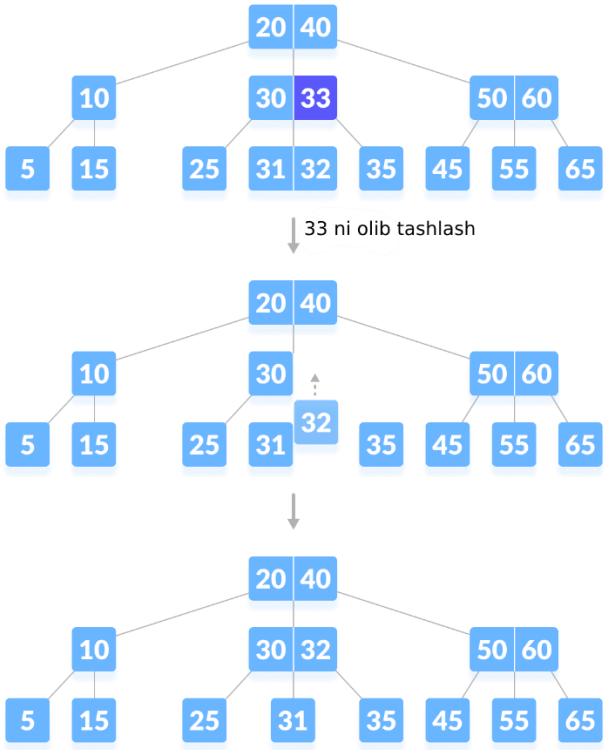


Agar ikkala to‘g’ridan-to‘g’ri bog’liq tugunlar allaqachon minimal miqdordagi kalitlarga ega bo‘lsa, unda tugunni chap yaqin tugun yoki o‘ng yaqin tugun bilan birlashtiriladi. Ushbu birlashma ildiz tuguni orqali amalga oshiriladi.30 ni olib tashlash yuqoridagi holatga olib keladi.

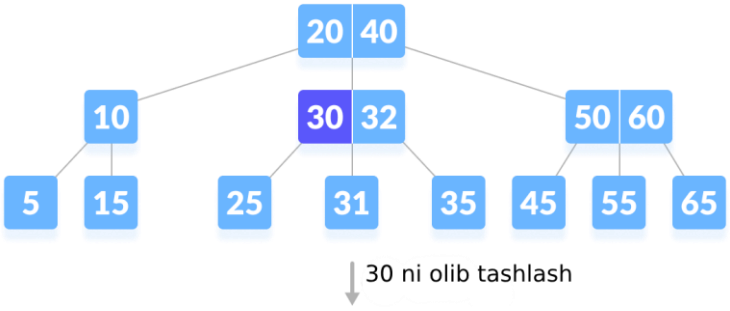


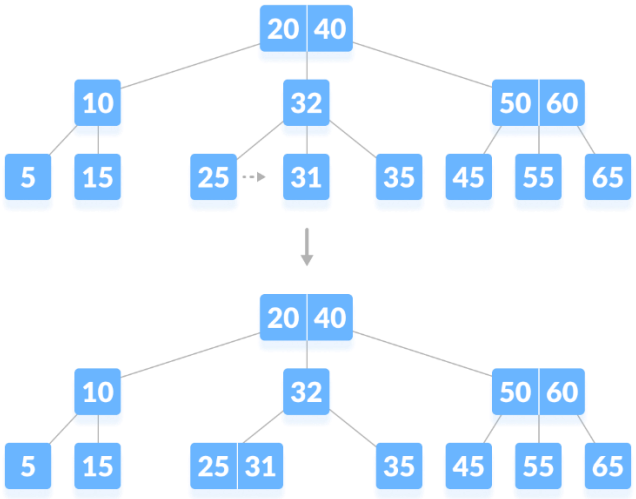
2-holat: Agar olib tashlanadigan kalit ichki tugunda bo‘lsa, quyidagi holatlar yuzaga keladi.

- Olib tashlangan ichki tugun, agar chap voris minimal miqdordagi kalitlarga ega bo‘lsa, oldingi tartib bilan almashtiriladi.



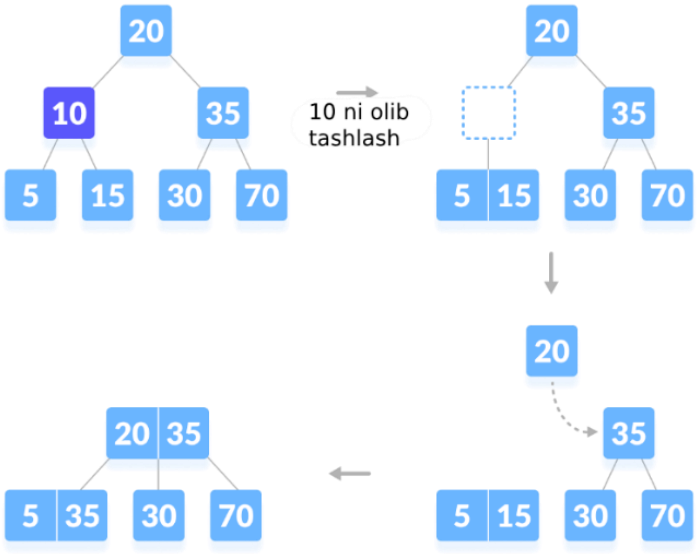
- Olib tashlangan ichki tugun, agar o‘ng voris tugunida minimal miqdordagi kalitlar bo‘lsa, ketma-ket voris bilan almashtiriladi.
- Agar biron bir voris tugunida aniq minimal miqdordagi kalitlar bo‘lsa, chap va o‘ng bola tugunlarini birlashtiriladi..





3-holat: Bunday holda, daraxtning balandligi pasayadi. Agar maqsadli kalit ichki tugunda bo‘lsa va kalitni olib tashlash tugundagi kalitlar sonining kamayishiga olib kelsa (ya’ni minimal talabdan kam), keyin ketma-ket oldingi va ketma-ket vorisni qidiramiz. Agar ikkala voris ham minimal miqdordagi kalitlarni o‘z ichiga olsa, unda qarz olish mumkin emas. Bu 2(3) holatga, ya’ni vorislar elementlarining birlashishiga olib keladi.

Kalitni qarzga olish uchun yana yaqin tugunni toping. Ammo, agar yaqin tugunda faqat minimal miqdordagi kalitlar bo‘lsa, tugunni ildiz tuguni va yaqin tugunlar bilan birga birlashtiring. Voris tugunlarini mos ravishda joylashtiring (o‘sish tartibida).



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?