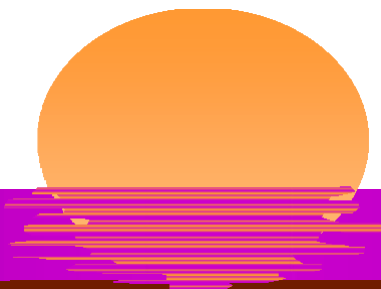


Ү.СS203 - Өгөгдлийн бүтэц ба алгоритм 2022-2023

Д.Золбоо
МТ-н салбарын багш





Хоёртын хайлтын мод



- Толь бичгийн үйлдлүүд:
 - `get(key)`
 - `put(key, value)`
 - `remove(key)`
- Нэмэлт үйлдлүүд:
 - `ascend()`
 - `get(index)` (индексстэй хоёртын хайлтын мод)
 - `remove(index)` (индексстэй хоёртын хайлтын мод)

Толь бичгийн get(), put() and remove() үйлдлүүдийн хугацаа

Өгөгдлийн бүтэц	Муу тохиолдол	Оновчтой
Хэш хүснэгт	$O(n)$	$O(1)$
Хоёртын хайлтын мод	$O(n)$	$O(\log n)$
Тэнцвэртэй хёртын хайлтын мод	$O(\log n)$	$O(\log n)$

n – толь бичгийн элементийн тоо

Бусад `ascend()`, `get(index)`, `remove(index)` үйлдлүүдийн хугацаа

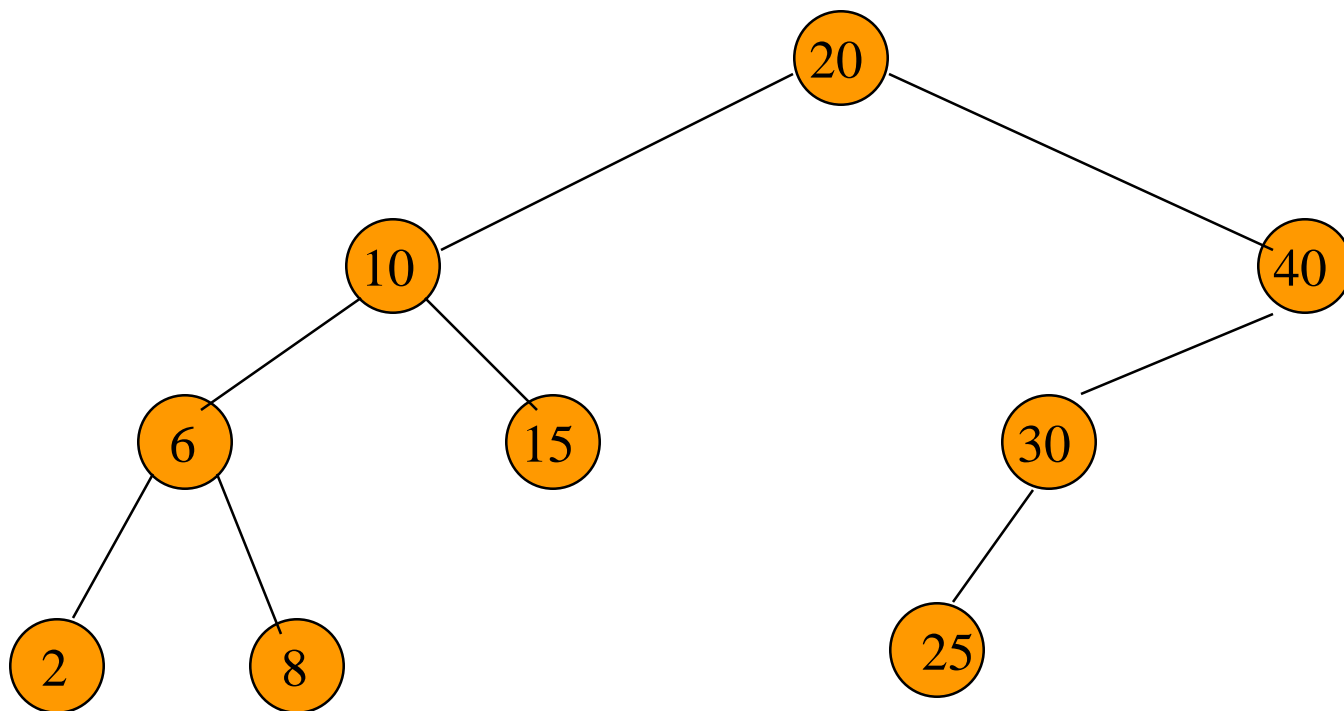
Өгөгдлийн бүтэц	<code>ascend</code>	<code>get and remove</code>
Хэш хүснэгт	$O(D + n \log n)$	$O(D + n \log n)$
Индексстэй BST	$O(n)$	$O(n)$
Индексстэй тэнцвэржсэн BST	$O(n)$	$O(\log n)$

D – багцын тоо

Хоёртын хайлтын модны тодорхойлолт

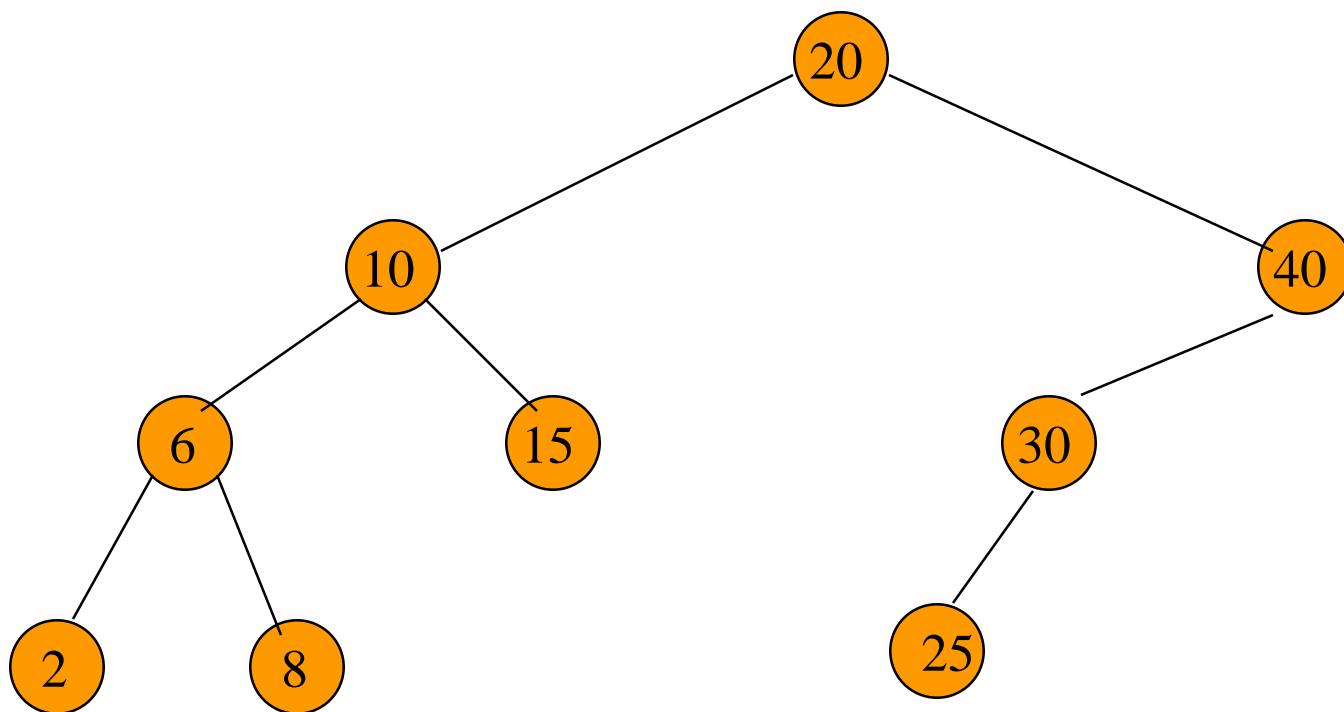
- Хоёртын мод.
- Зангилаа бүр (key, value) хостой.
- Аливаа **x** зангилааны хувьд, түүний зүүн дэд модны түлхүүрүүд **x** –ийнхээс бага байна.
- Аливаа **x** зангилааны хувьд, түүний баруун дэд модны түлхүүрүүд **x** –ийнхээс их байна.

Хоёртын хайлтын модны жишээ



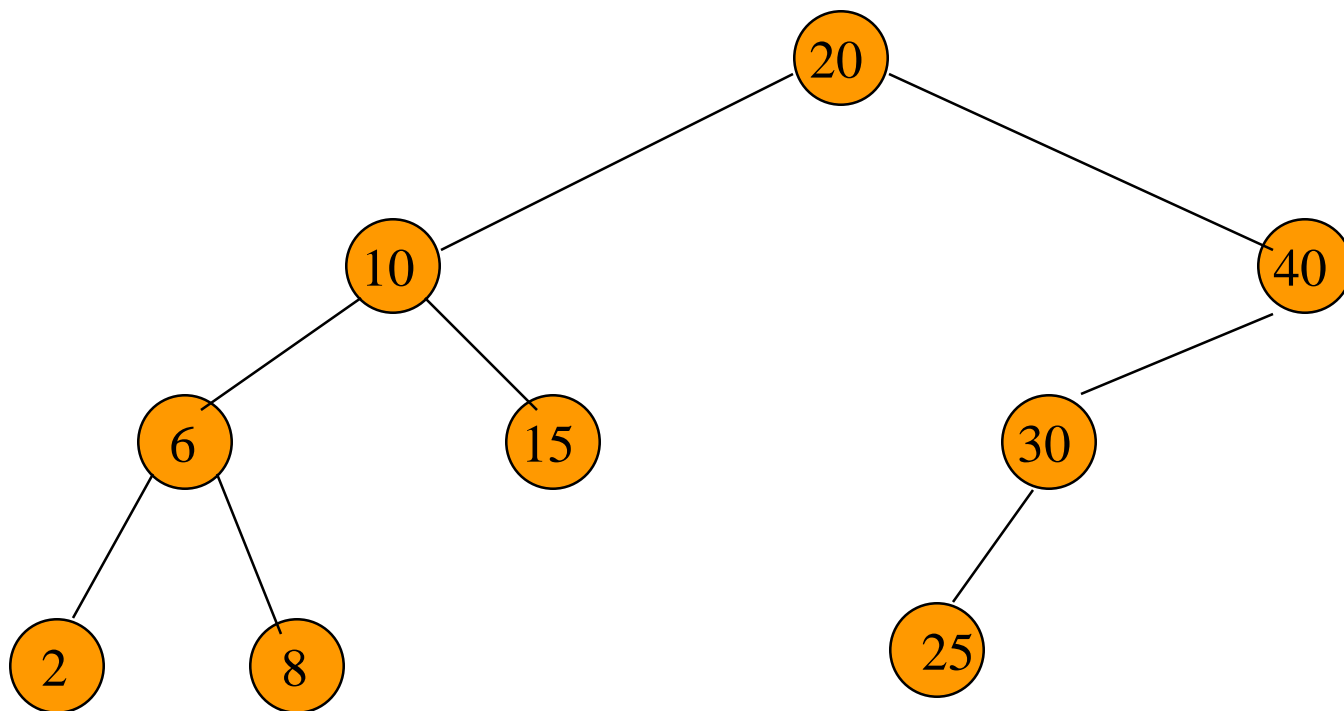
Зөвхөн түлхүүрүүдийг харууллаа.

ascend() үйлдэл



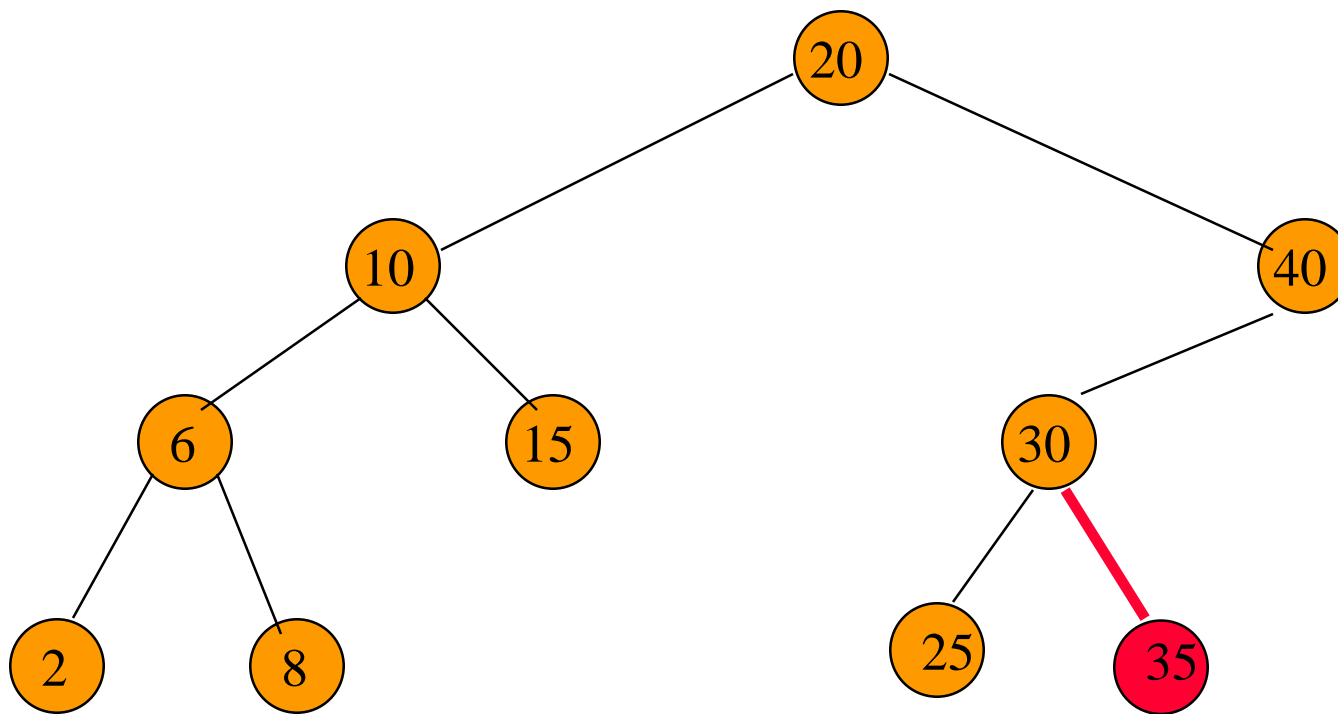
inorder аялалыг хийнэ. $O(n)$ хугацаа.

get() үйлдэл



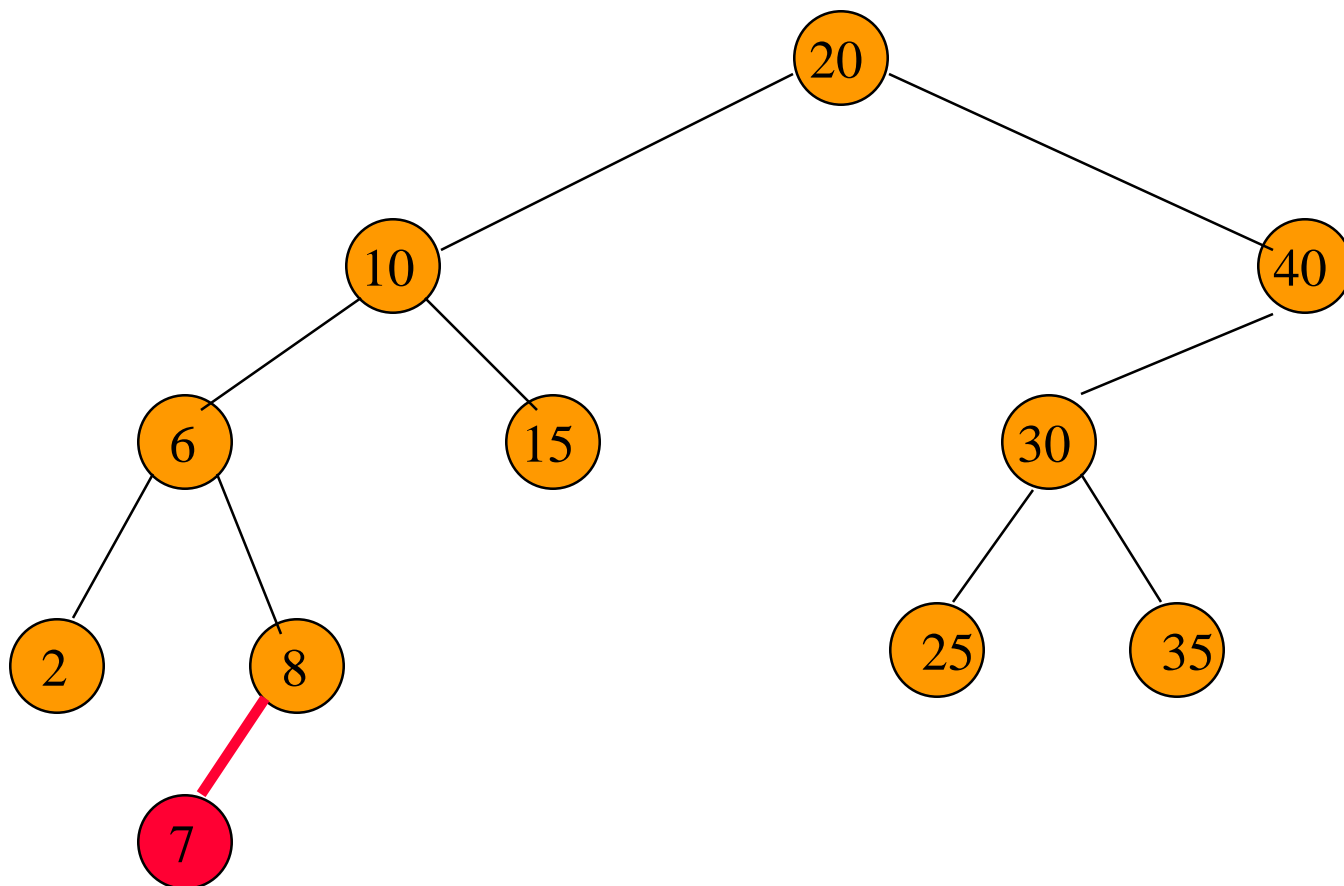
Хугацаа $O(\text{өндөр}) = O(n)$, үүнд n –
зангилаа/элементийн тоо.

put() үйлдэл



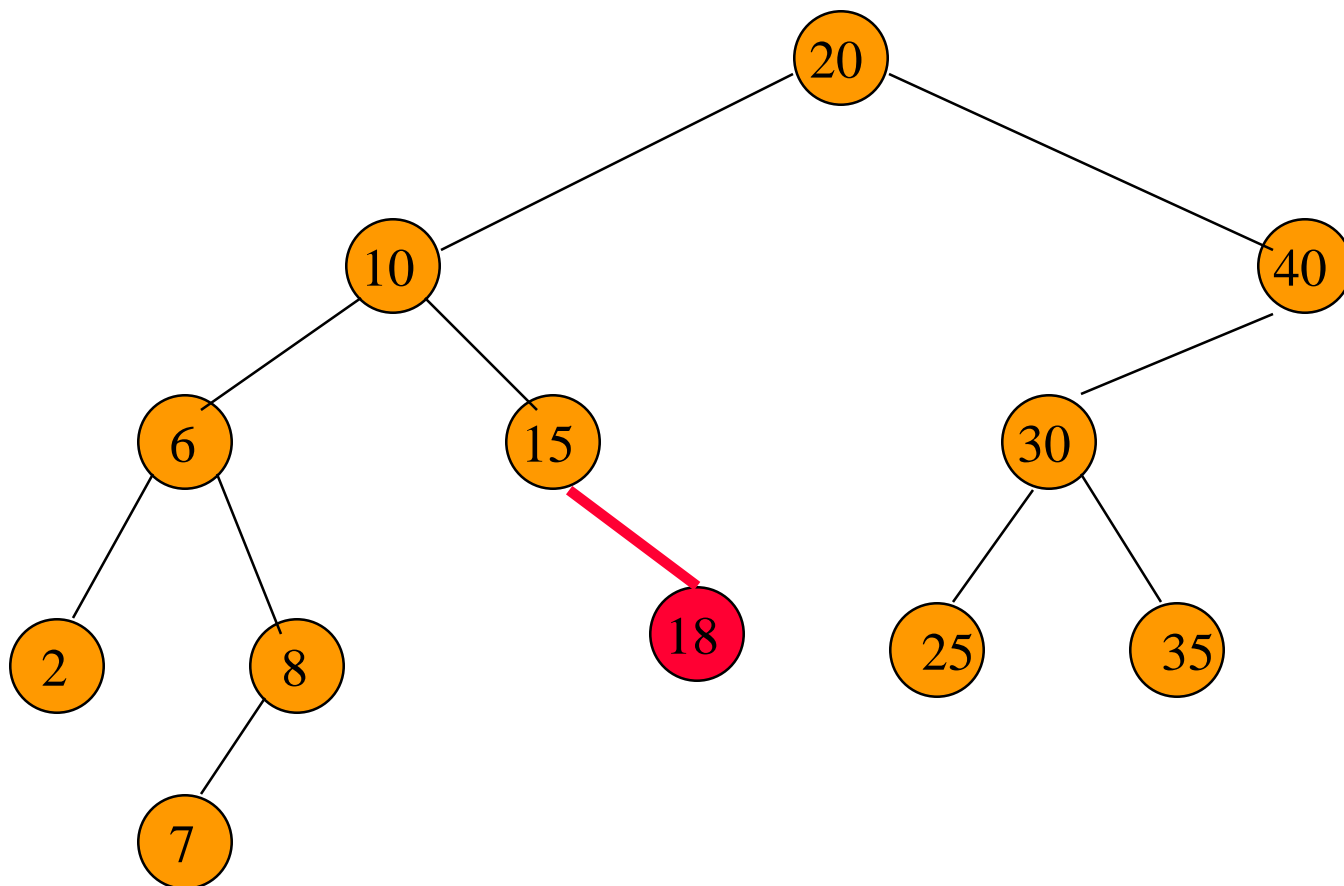
35 түлхүүртэй хосыг хийх

put() үйлдэл



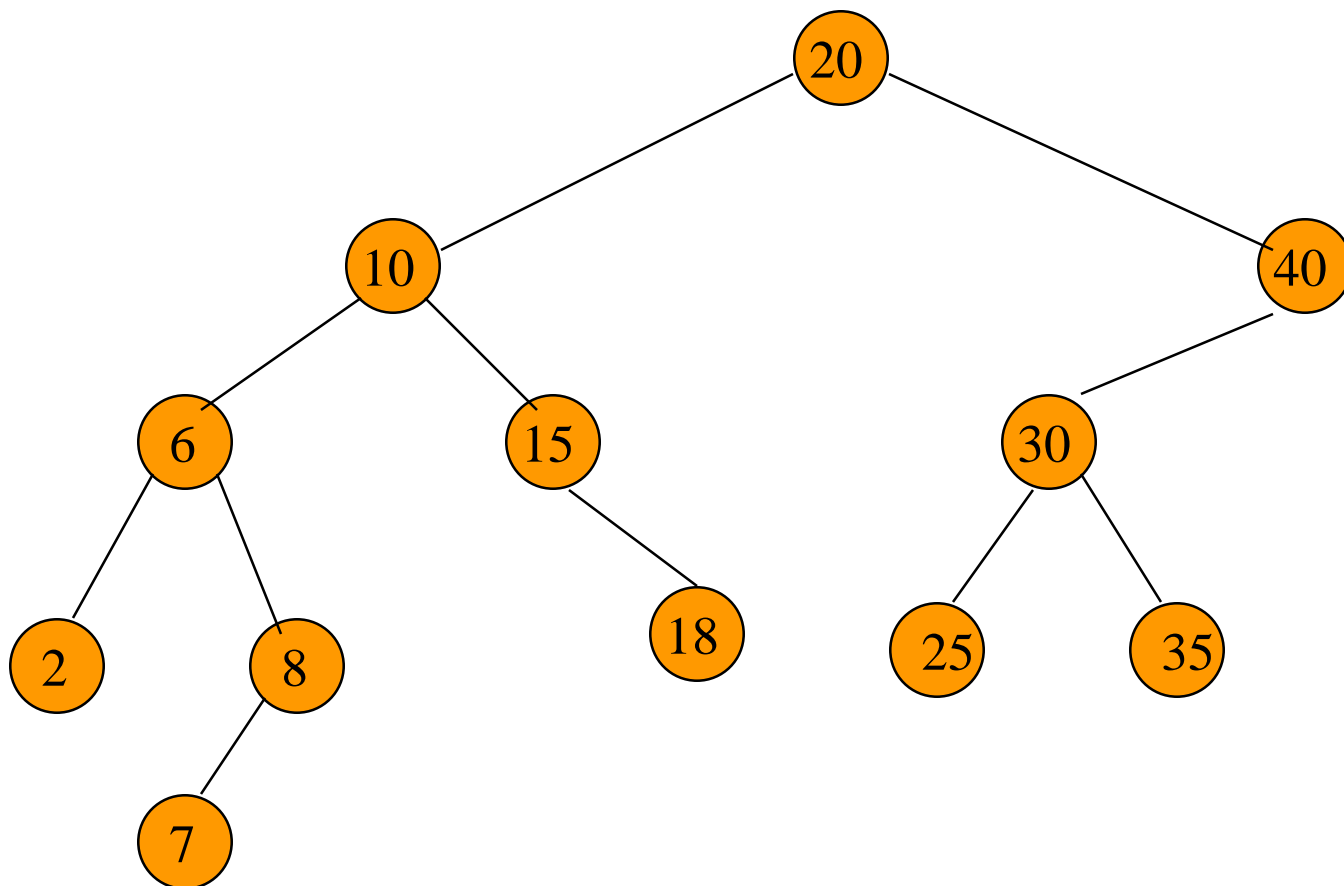
7 түлхүүртэй хосыг хийх

put() үйлдэл



18 түлхүүртэй хосыг хийх

put() үйлдэл



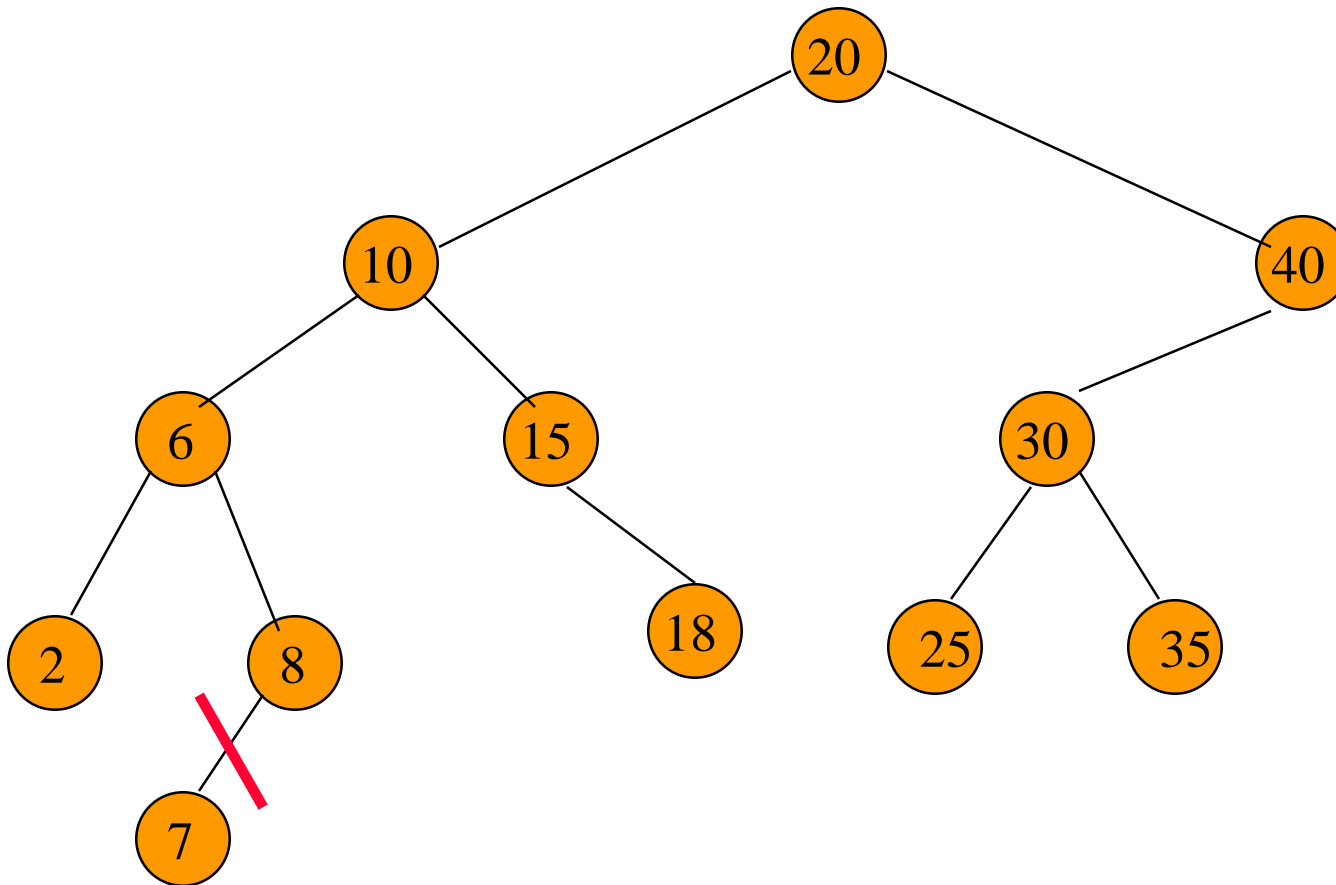
put() үйлдлийн хугацаа $O(\text{өндөр})$.

remove() үйлдэл

Гурван тохиолдол:

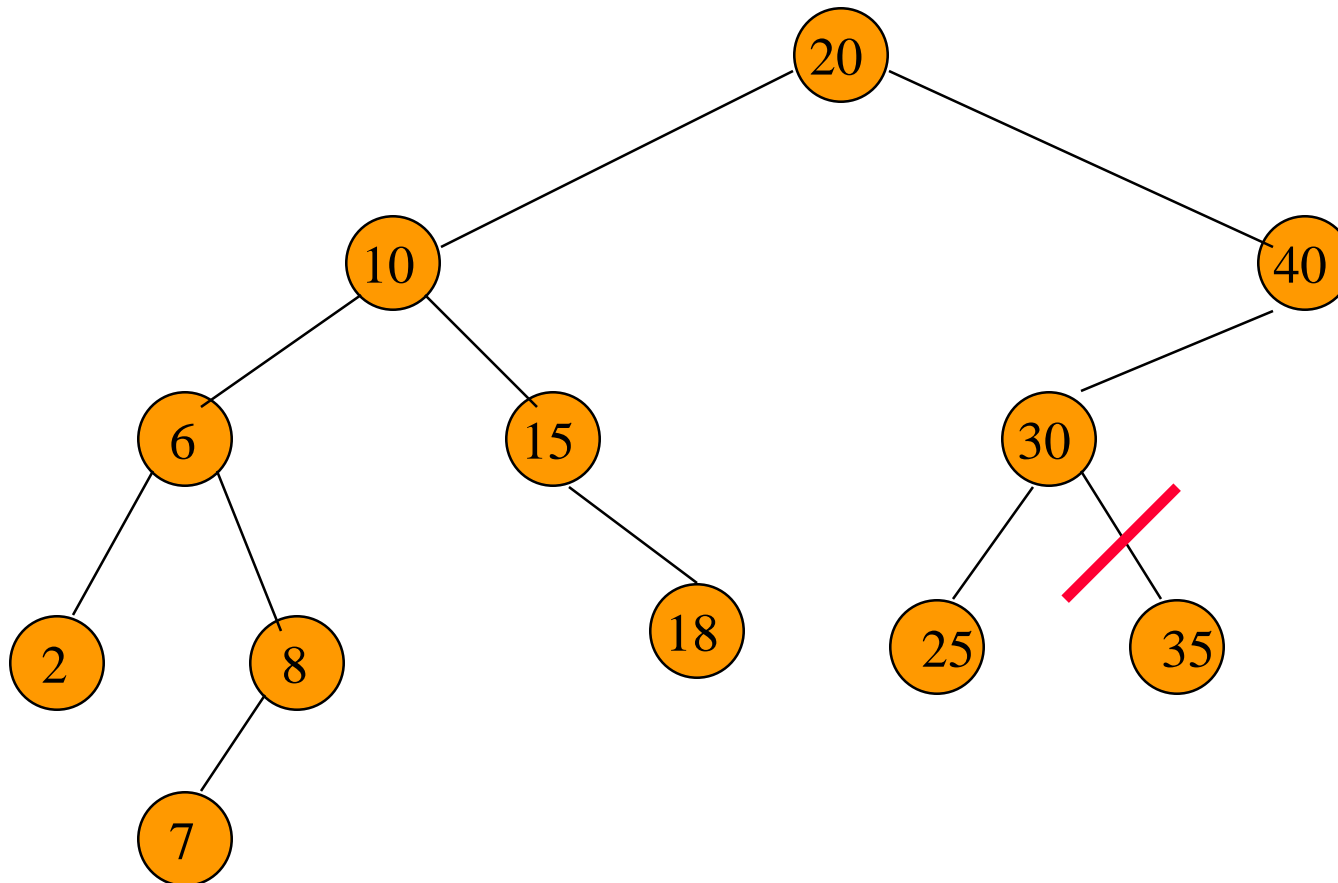
- Элемент навч.
- Элемент 1 зэрэглэлтэй зангилаа.
- Элемент 2 зэрэглэлтэй зангилаа.

Навчийг устгах



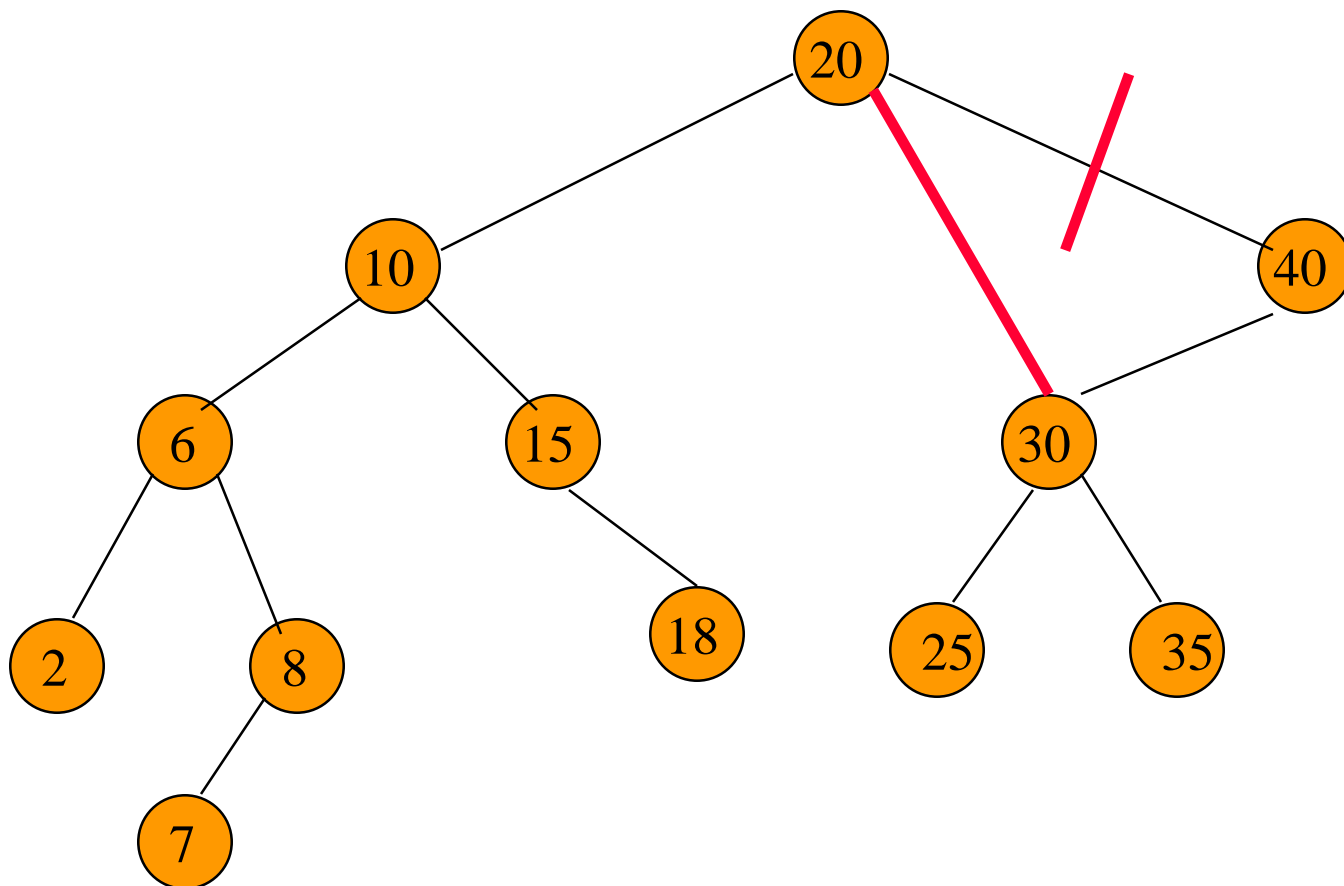
element.key = 7 навчийг устгах

Навчийг устгах



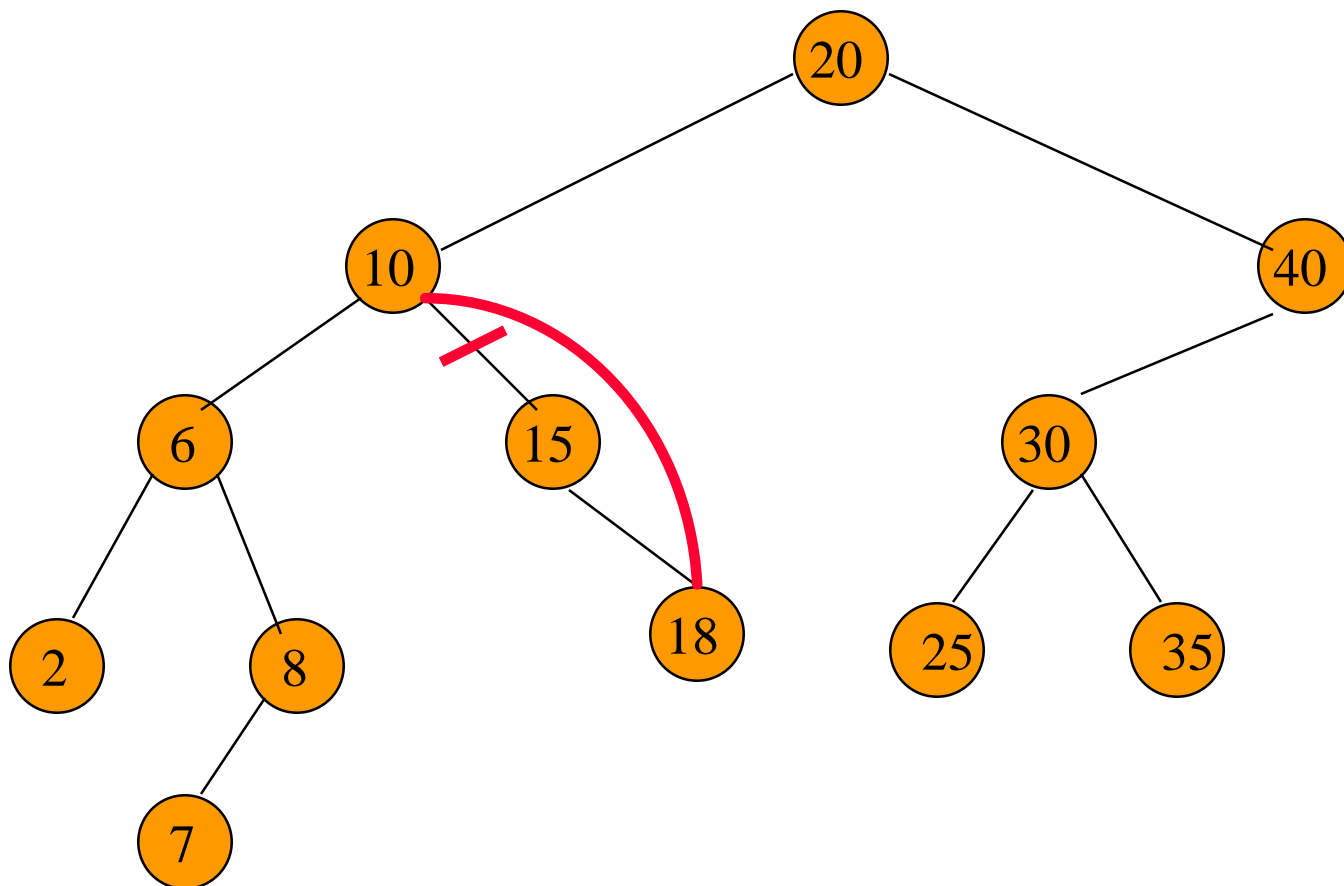
element.key = 35 вавчийг устгах

Зэрэглэл 1 –тэй зангилааг устгах



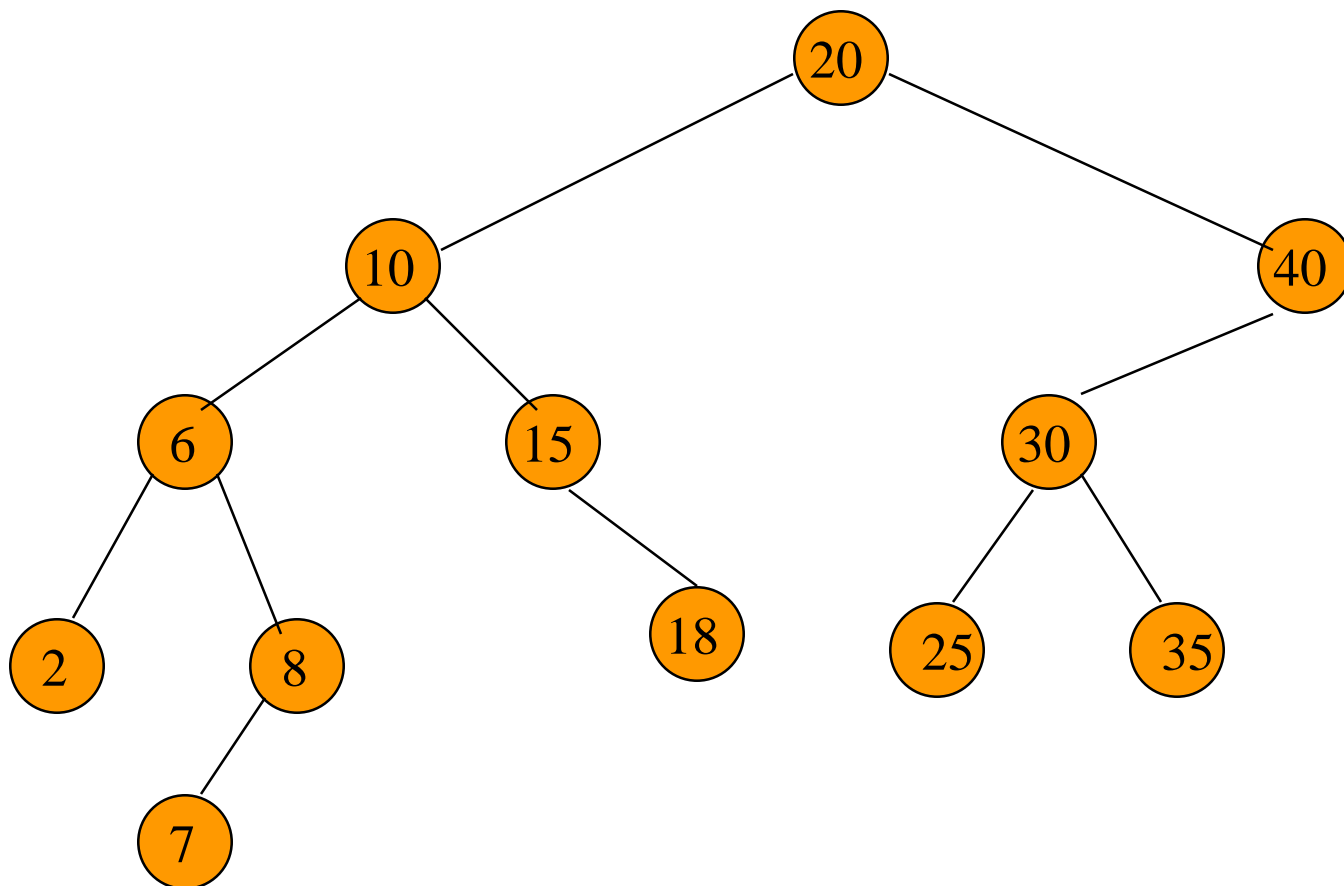
Зэрэглэл 1 –тэй $\text{node.key} = 40$ зангилааг устгах

Зэрэглэл 1 –тэй зангилааг устгах



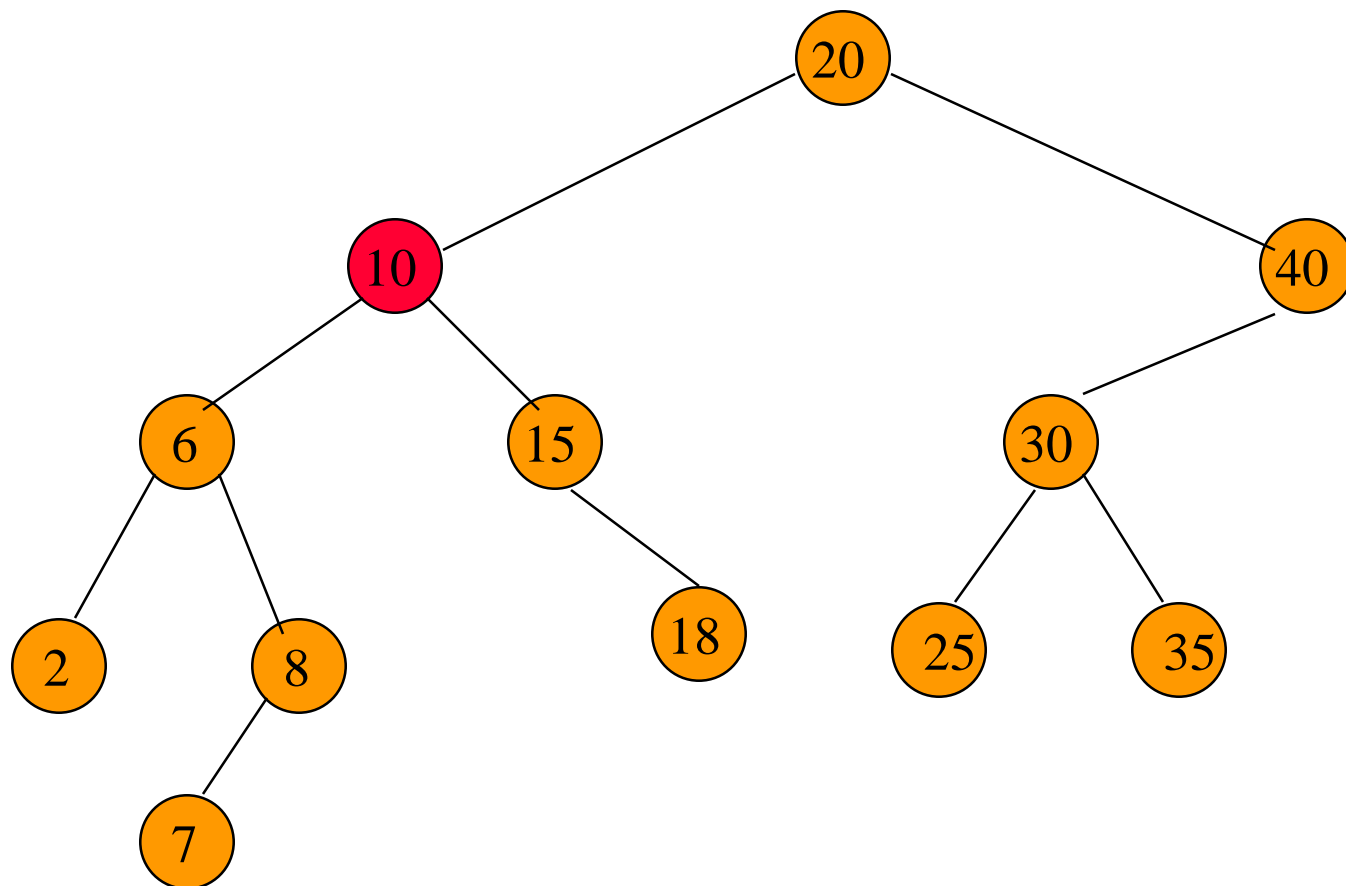
Зэрэглэл 1 –тэй node.key = 15 зангилааг устгах

Зэрэглэл 2 –тэй зангилааг устгах



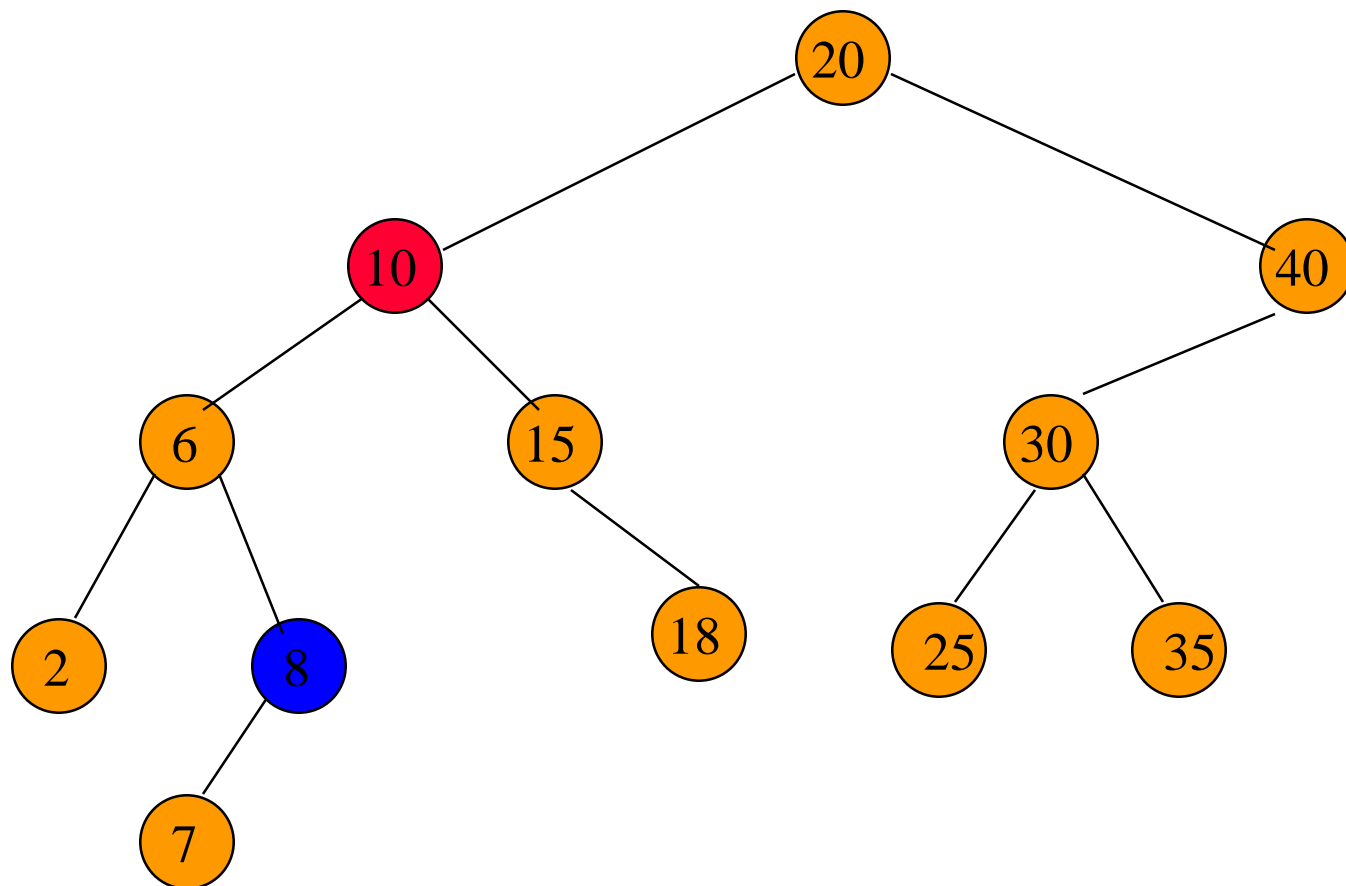
Зэрэглэл 2 –тэй node.key = 10 зангилааг устгах

Зэрэглэл 2 –тэй зангилааг устгах



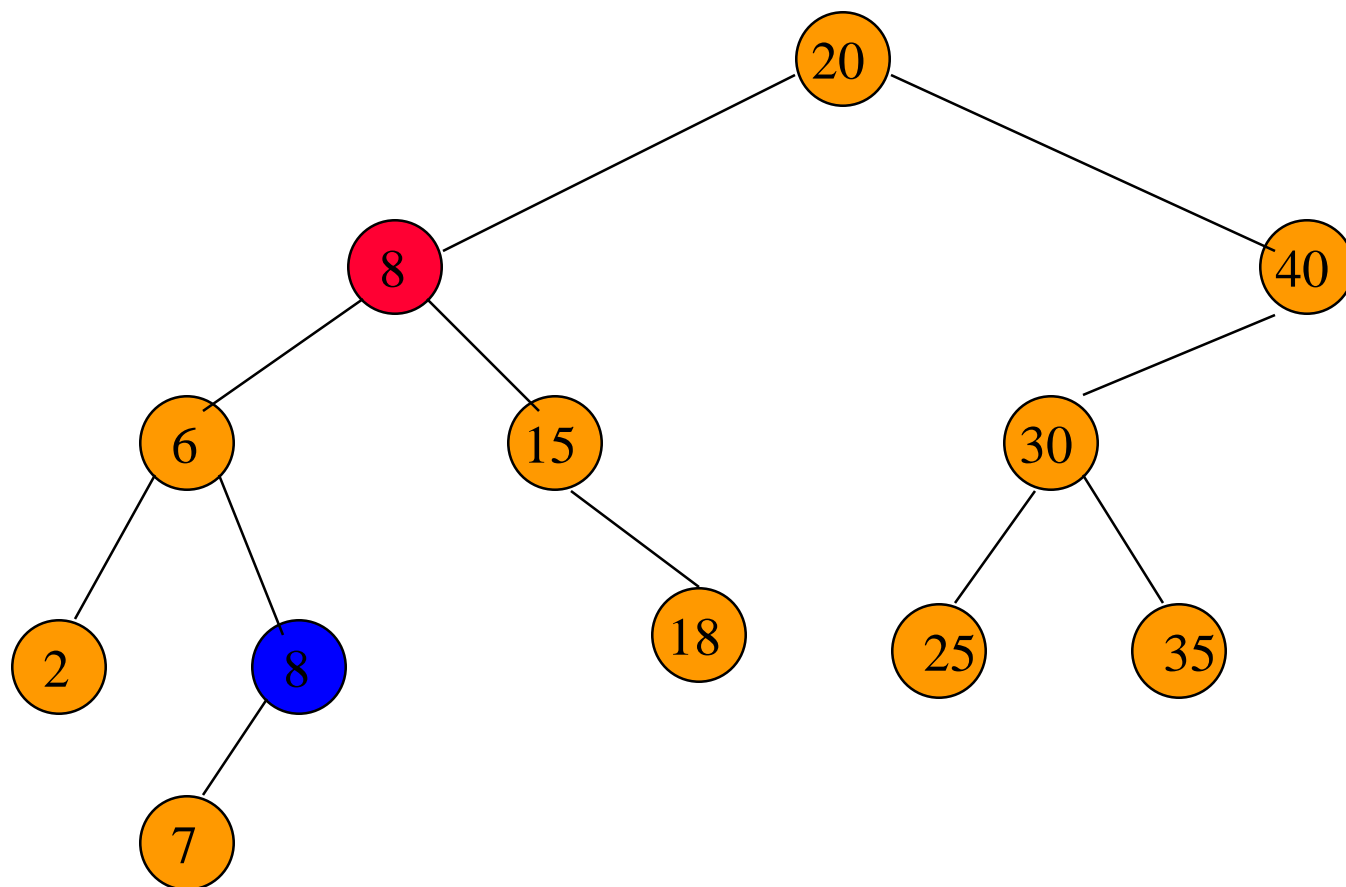
Зүүн дэд модны том түлхүүртэйгээр солих
(эсхүл баруун дэд модны бага түлхүүртэйгээр¹⁸).

Зэрэглэл 2 –тэй зангилааг устгах



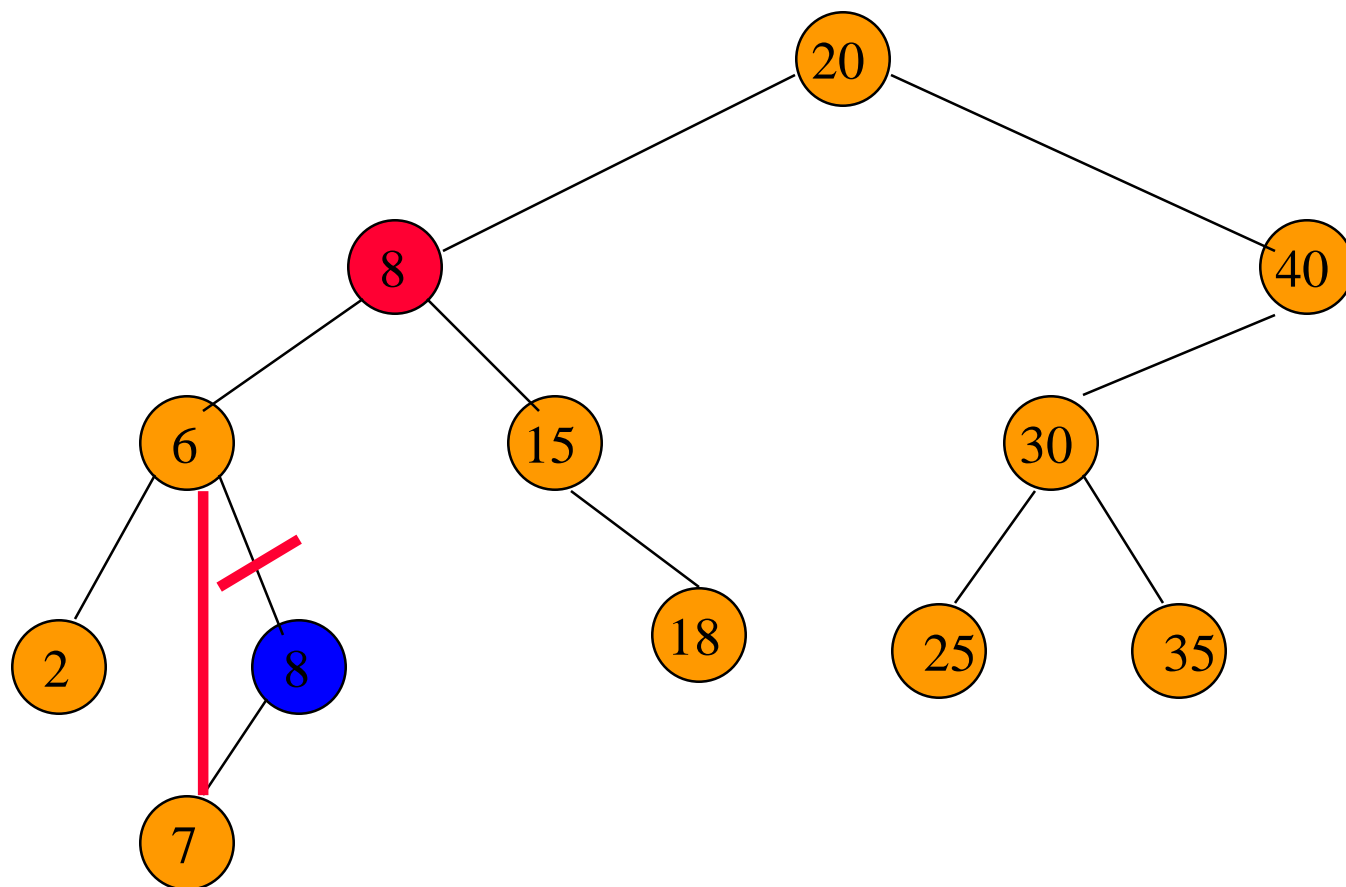
Зүүн дэд модны том түлхүүртэйгээр солих
(эсхүл баруун дэд модны бага түлхүүртэйгээр²⁸).

Зэрэглэл 2 –тэй зангилааг устгах



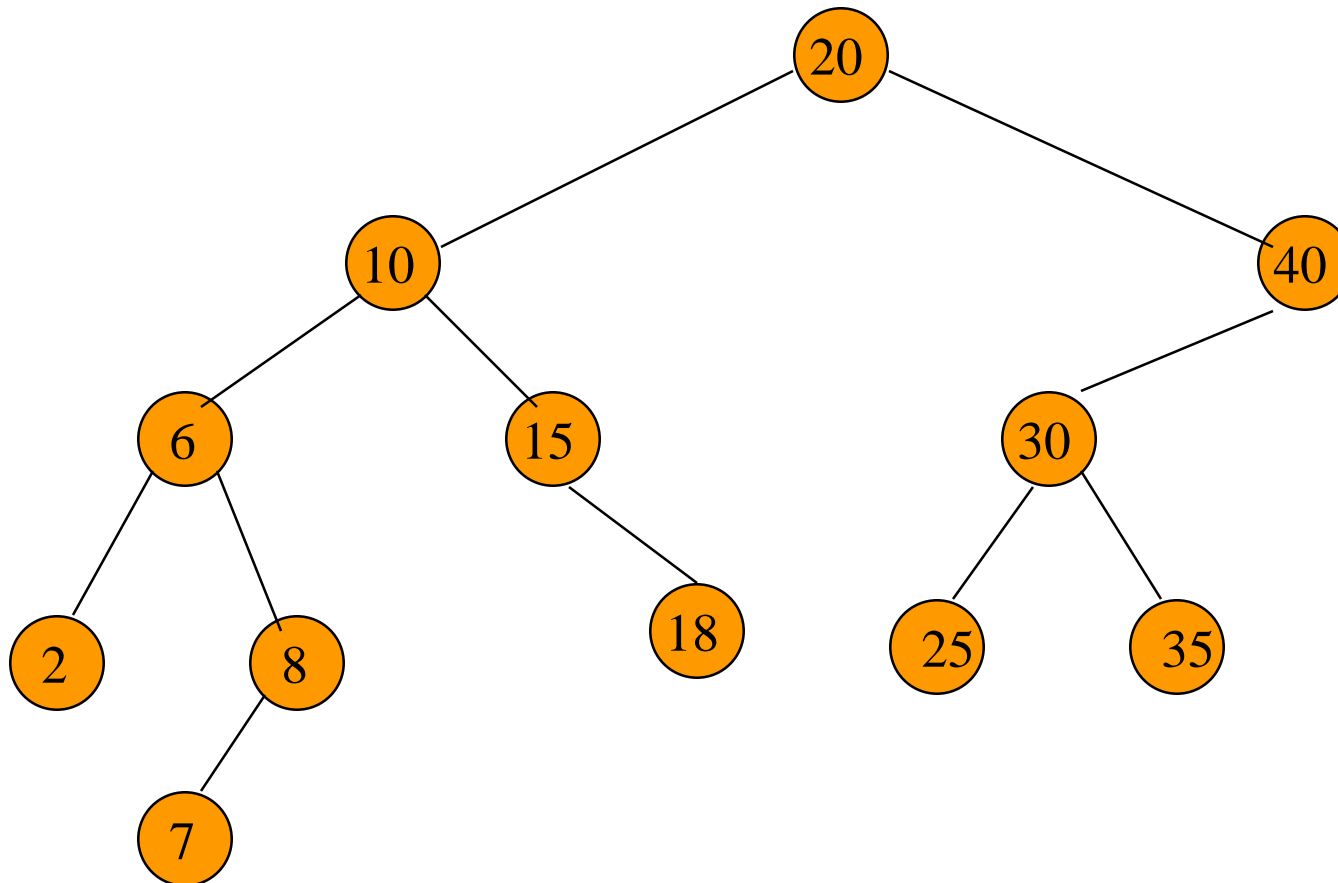
Зүүн дэд модны том түлхүүртэйгээр солих
(эсхүл баруун дэд модны бага түлхүүртэйгээр²).

Зэрэглэл 2 –тэй зангилааг устгах



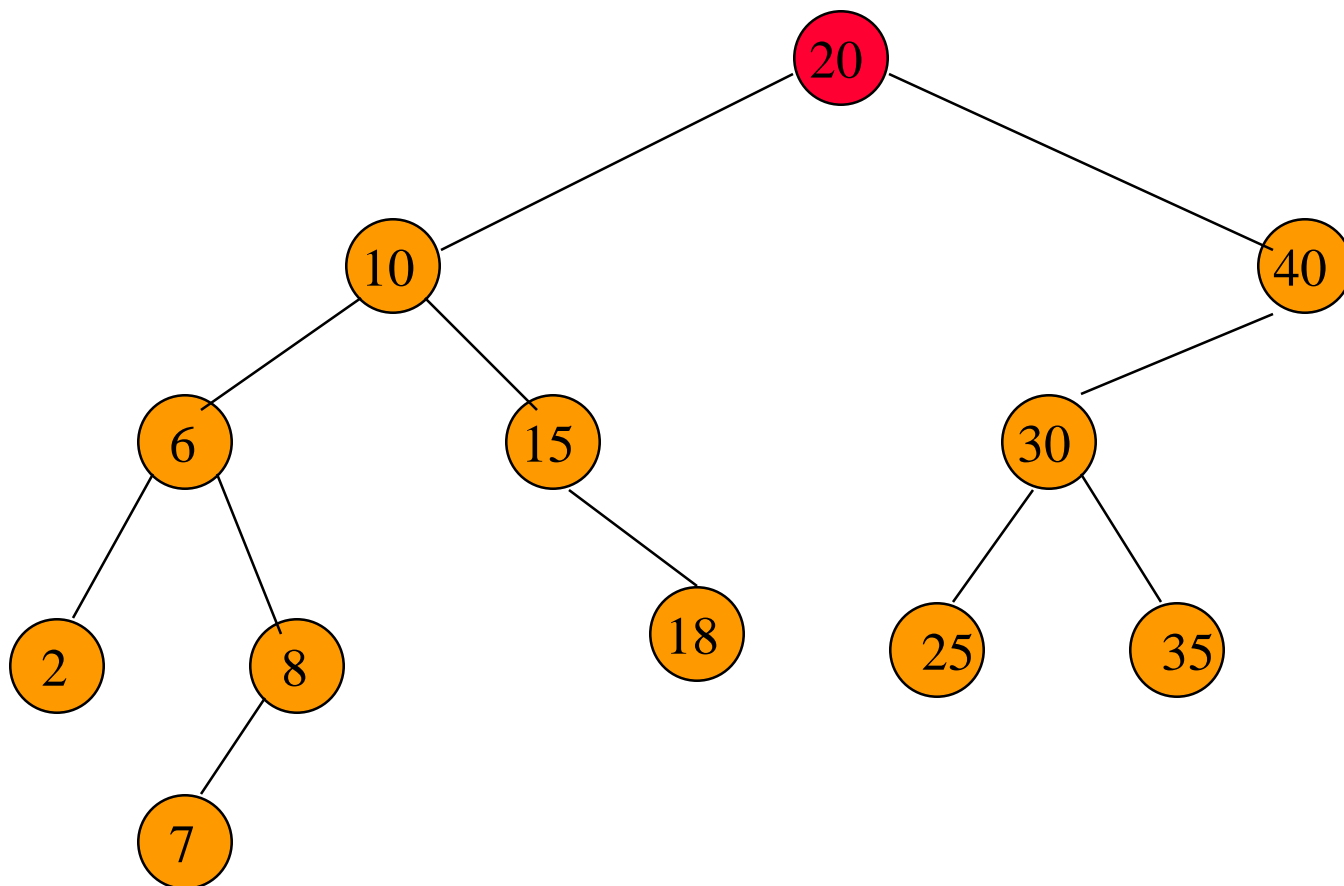
Том түлхүүр навч эсхүл зэрэглэл 1-тэй зангилаанд байх ёстой.

Зэрэглэл 2 –тэй зангилааг устгах өөр жишээ



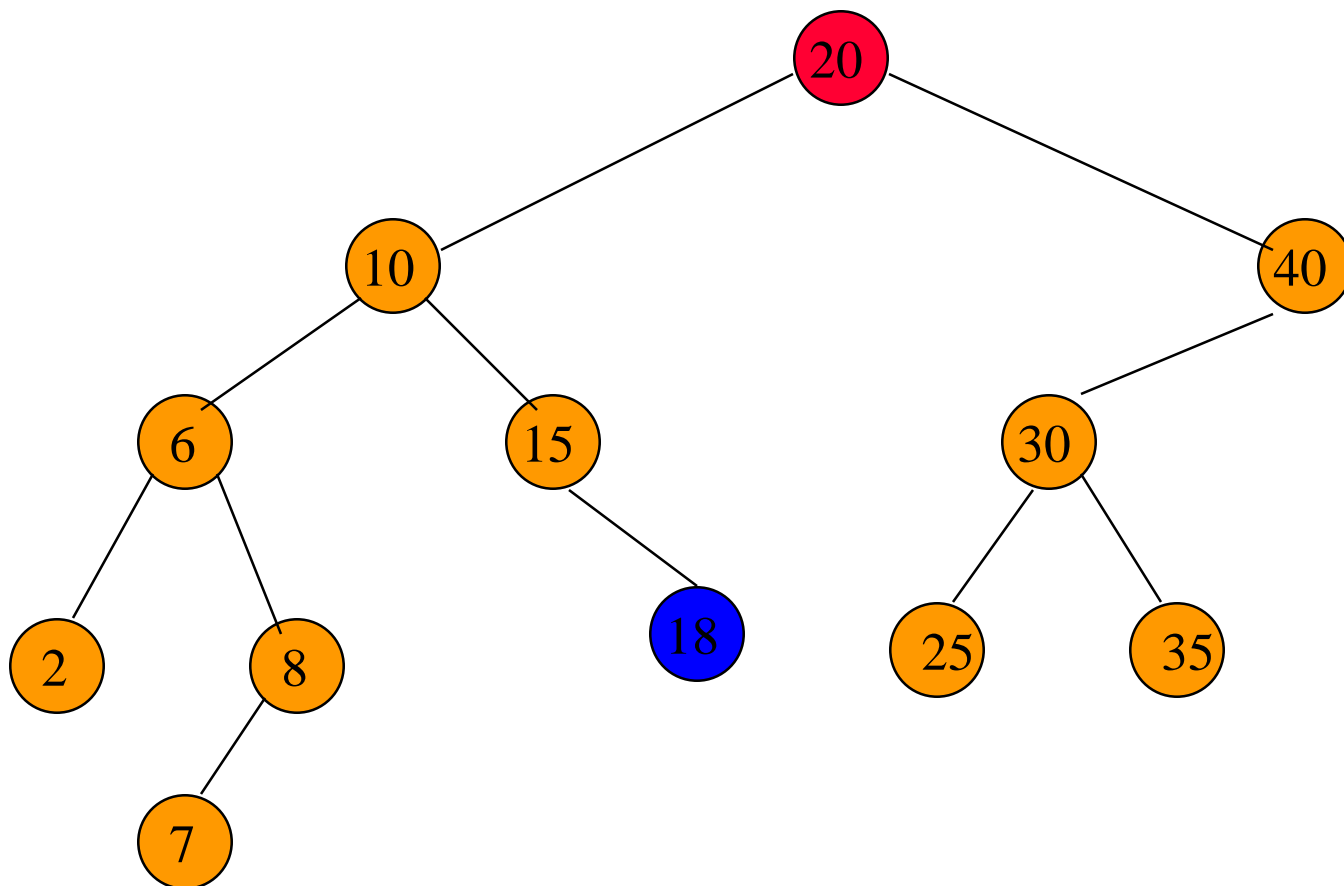
Зэрэглэл 2 –тэй $\text{node.key} = 20$ зангилааг устгах

Зэрэглэл 2 –тэй зангилааг устгах



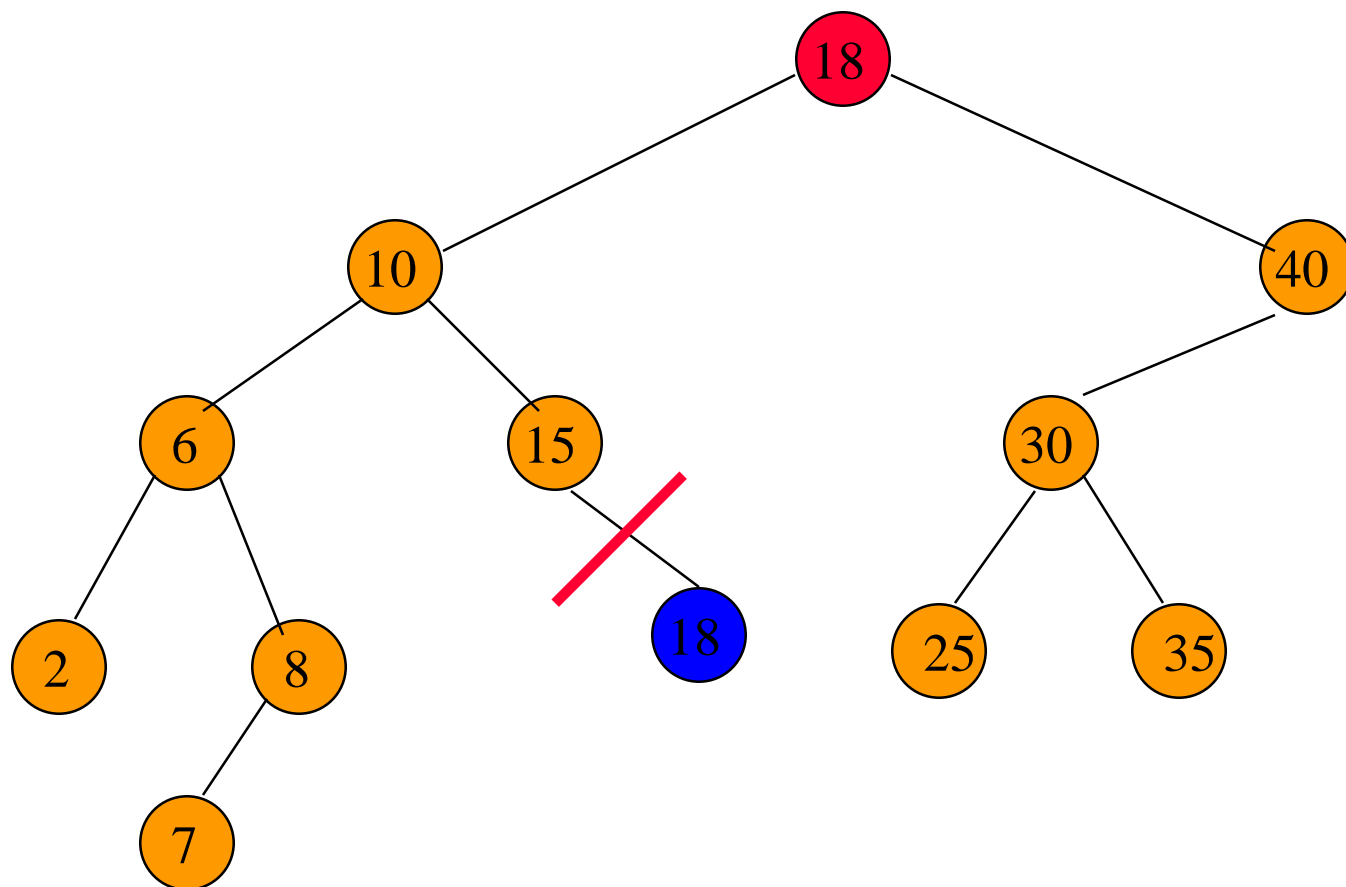
Зүүн дэд модны том түлхүүрээр солих.

Зэрэглэл 2 –тэй зангилааг устгах



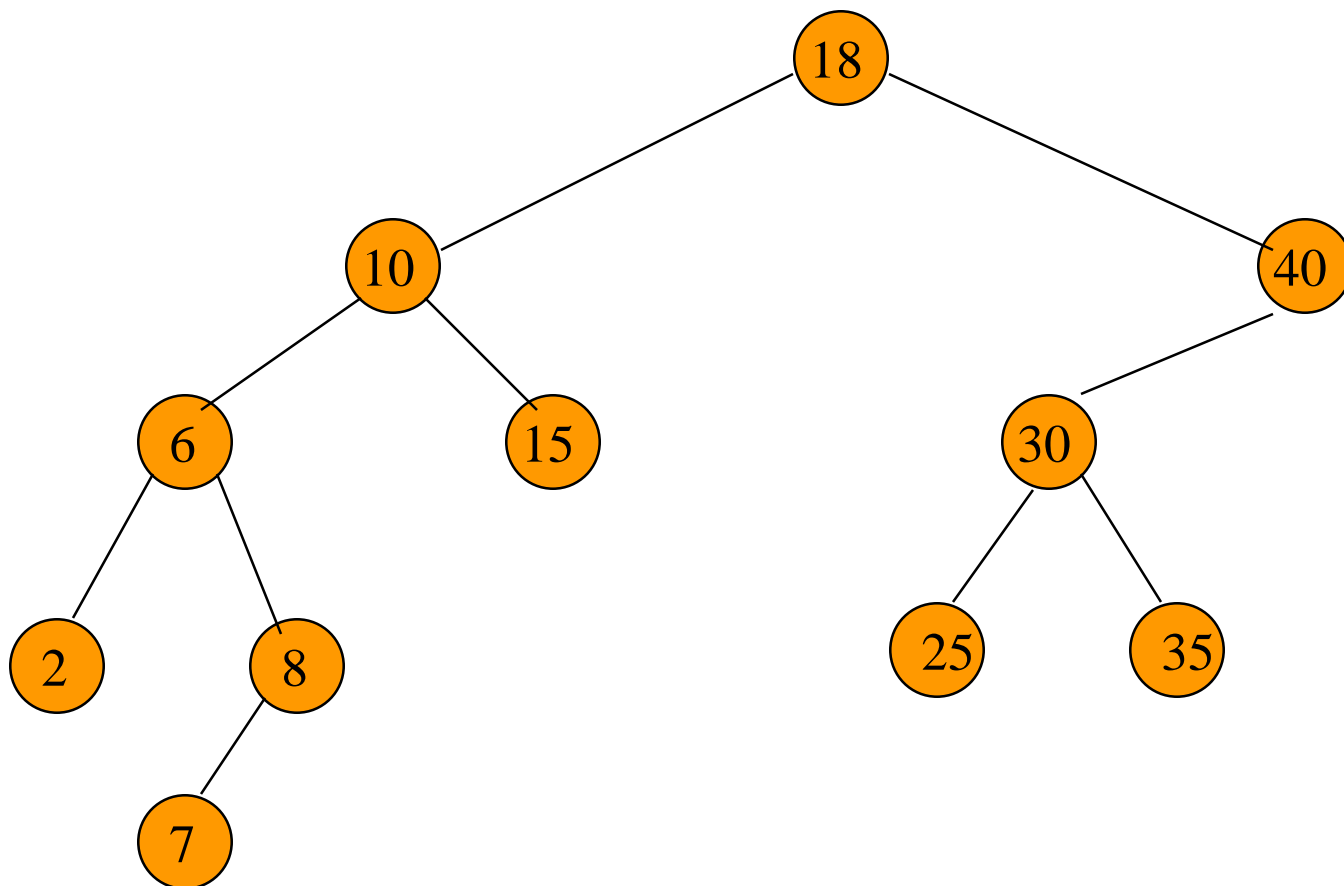
Зүүн дэд модны том түлхүүрээр солих.

Зэрэглэл 2 –тэй зангилааг устгах



Зүүн дэд модны том түлхүүрээр солих.

Зэрэглэл 2 –тэй зангилааг устгах

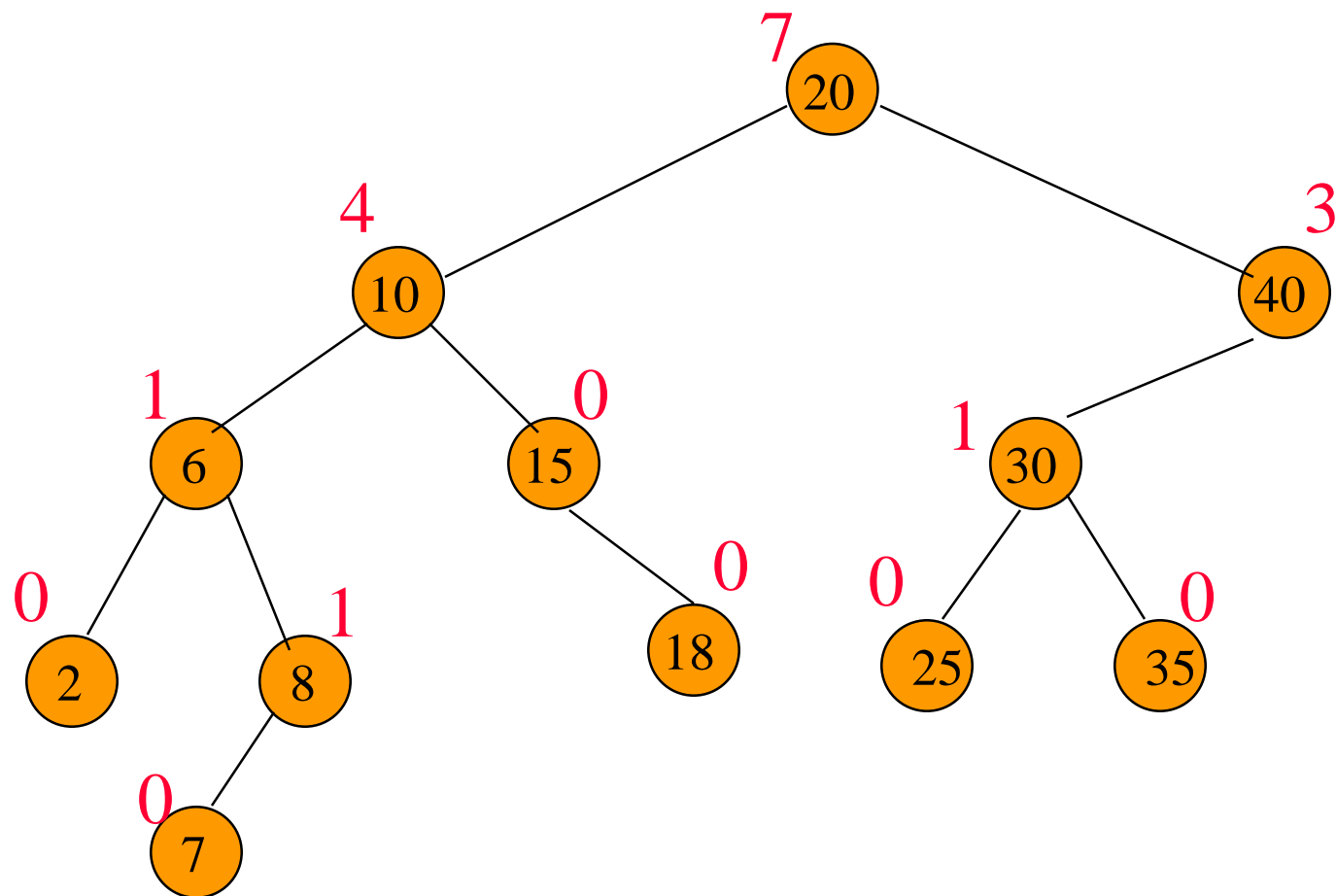


Хугацаа - $O(\text{өндөр})$.

Индексстэй хоёртын хайлтын мод

- Хоёртын хайлтын мод.
- Зангилаа бүхэн нэмэлт талбартай.
 - **leftSize** = зүүн дэд модны зангилааны тоо

Индексстэй хоёртын хайлтын модны жишээ



leftSize –ийн утга улаанаар

leftSize ба эрэмбэ

Эрэмбэ(Rank) нь элементийн inorder дарааллын байршил болно(inorder = түлхүүрийн өсөх дараалал).

[2,6,7,8,10,15,18,20,25,30,35,40]

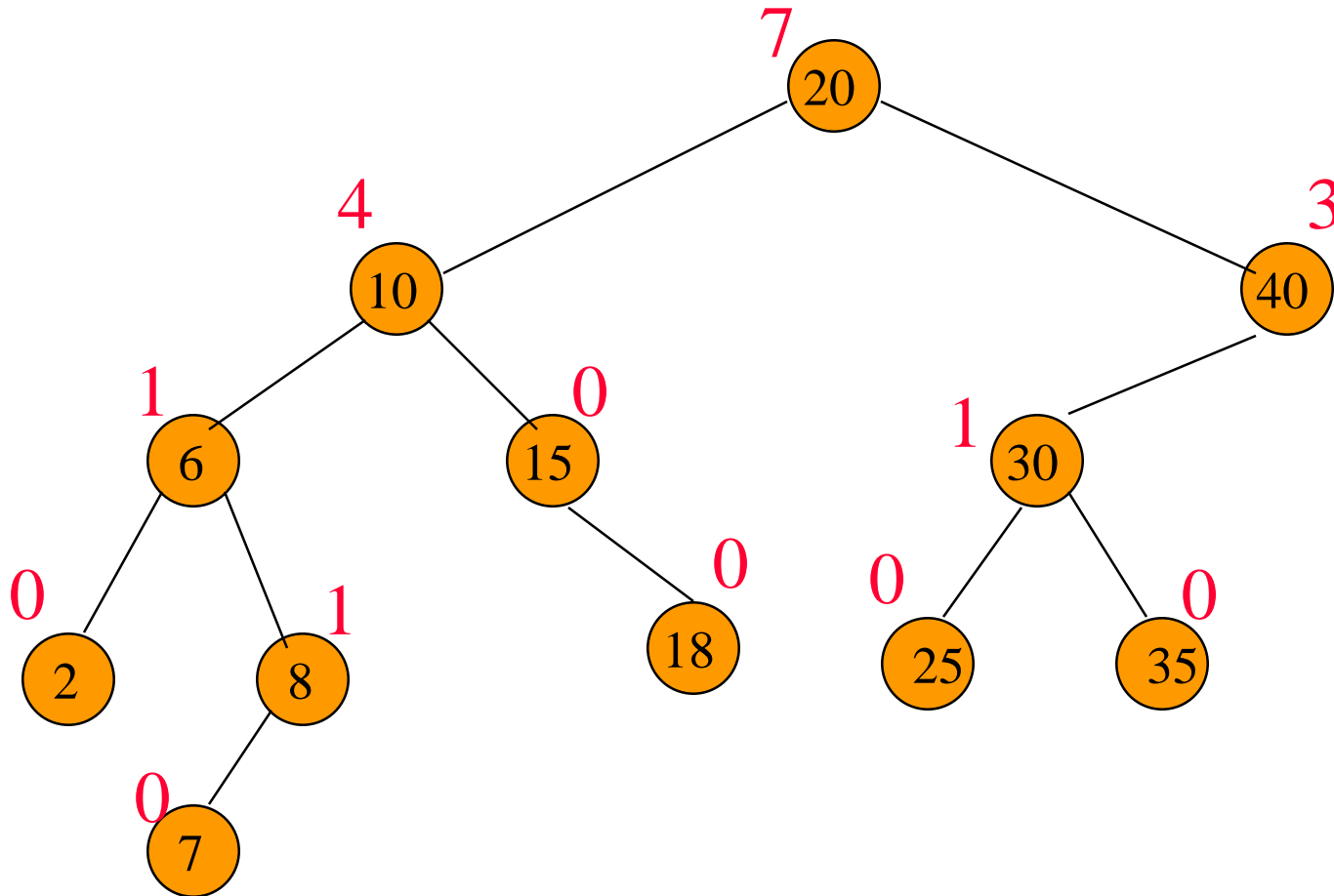
$$\text{rank}(2) = 0$$

$$\text{rank}(15) = 5$$

$$\text{rank}(20) = 7$$

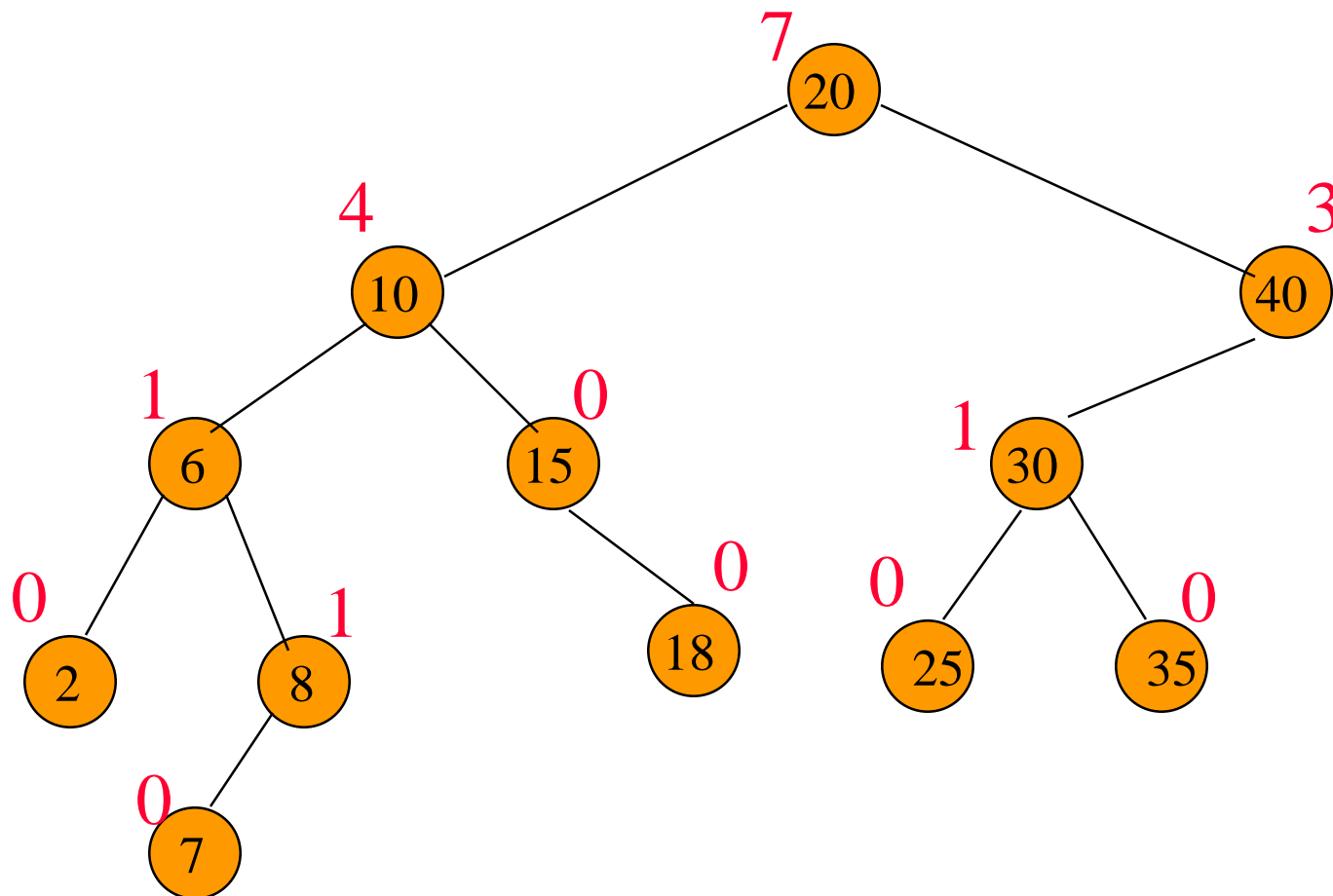
$\text{leftSize}(x) = \text{rank}(x) - x$ —ээс үндэстэй зүүн дэд модны элементүүдийн хувьд

leftSize ба Rank



Эрэмбэлэгдсэн жагсаалт = [2,6,7,8,10,15,18,20,25,30,35,40]

get(index) ба remove(index)



Эрэлбэлэгдсэн жагсаалт = [2,6,7,8,10,15,18,20,25,30,35,40]

get(index) ба remove(index)

- хэрвээ $index = x.leftSize$ бол харгалзах элемент нь $x.element$
- хэрвээ $index < x.leftSize$ бол харгалзах элемент нь x –ийн зүүн дэд модны $index$ дэх элемент
- хэрвээ $index > x.leftSize$ бол харгалзах элемент нь x –ийн баруун дэд модны $(index - x.leftSize - 1)$ дахь элемент

Хэрэглээ

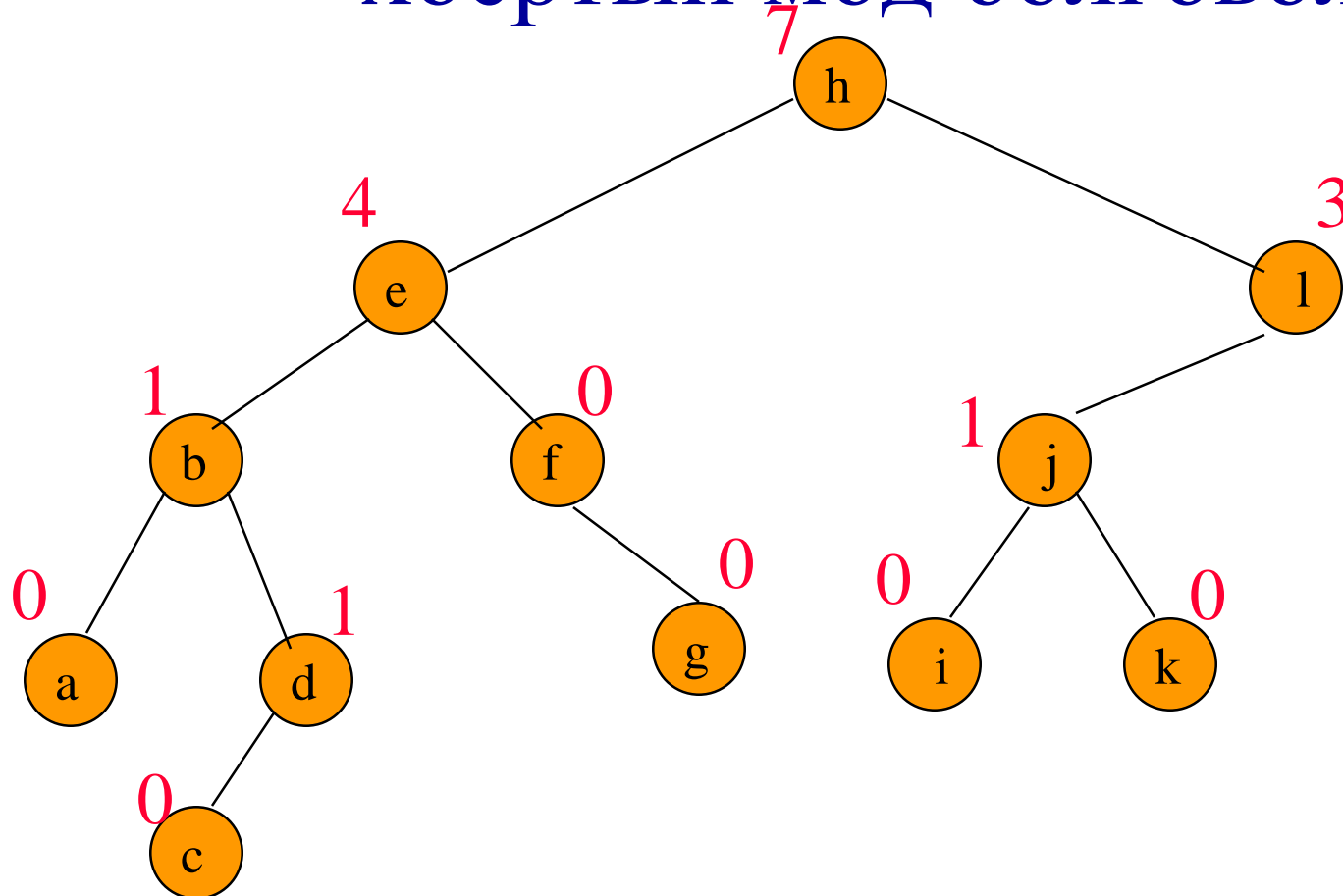
(Тэнцвэржүүлсэн модны хугацаа)

Ачааг Сайн дүүргэлтээр ачихад $O(n \log n)$ хугацаа орно.

Шугаман жагсаалтаар дүрслэхэд `get(index)`, `add(index, element)`, ба `remove(index)` үйлдлүүд $O(\log(\text{жагсаалтын хэмжээ}))$ хугацаанд хийгдэнэ (индекстэй хоёртын хайлтын биш мод ашиглахад).

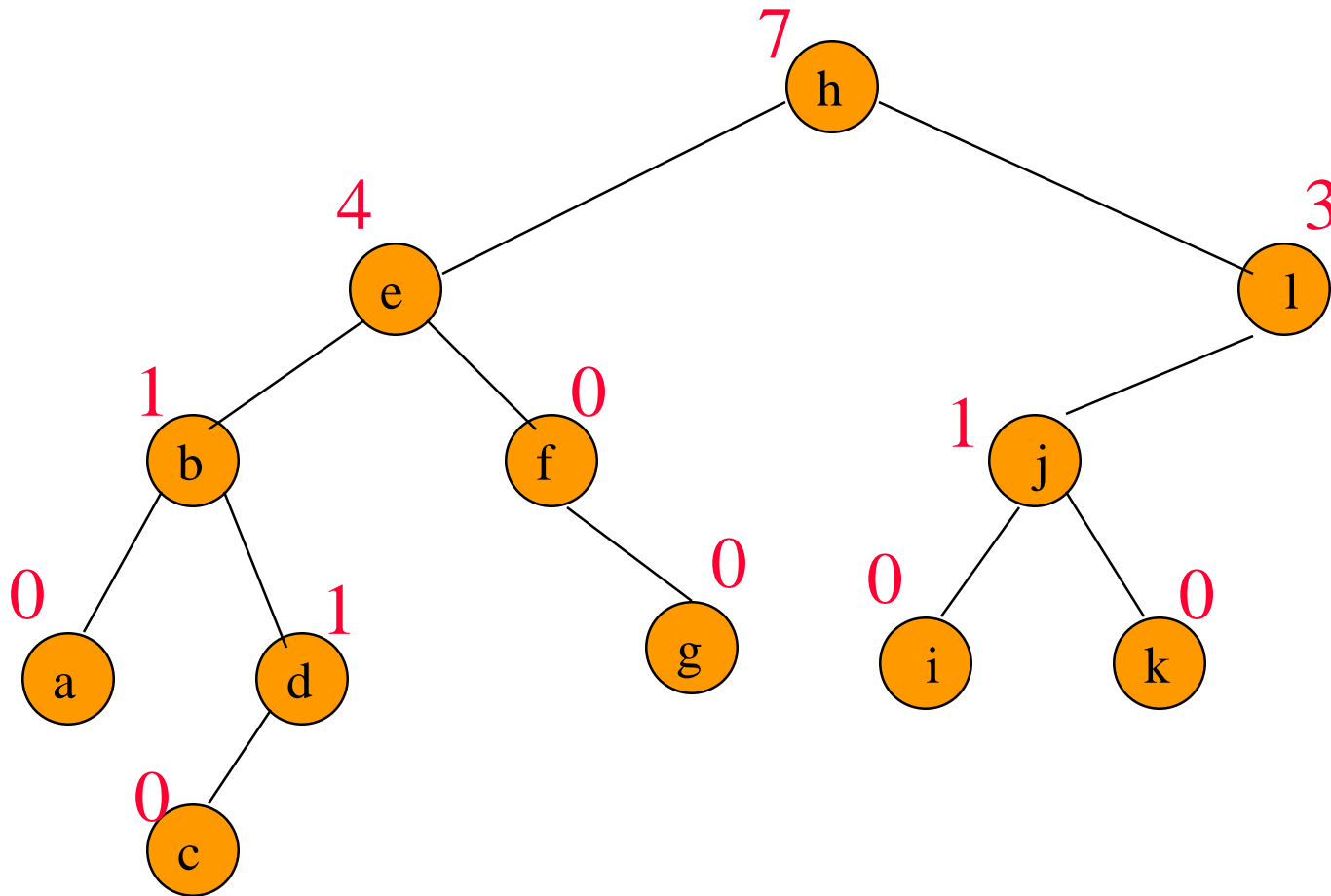
Эдгээрийн алинд ч хэш хүснэгт ашиглаж болохгүй.

Шугаман жагсаалтыг Индексстэй хоёртын мод болговол



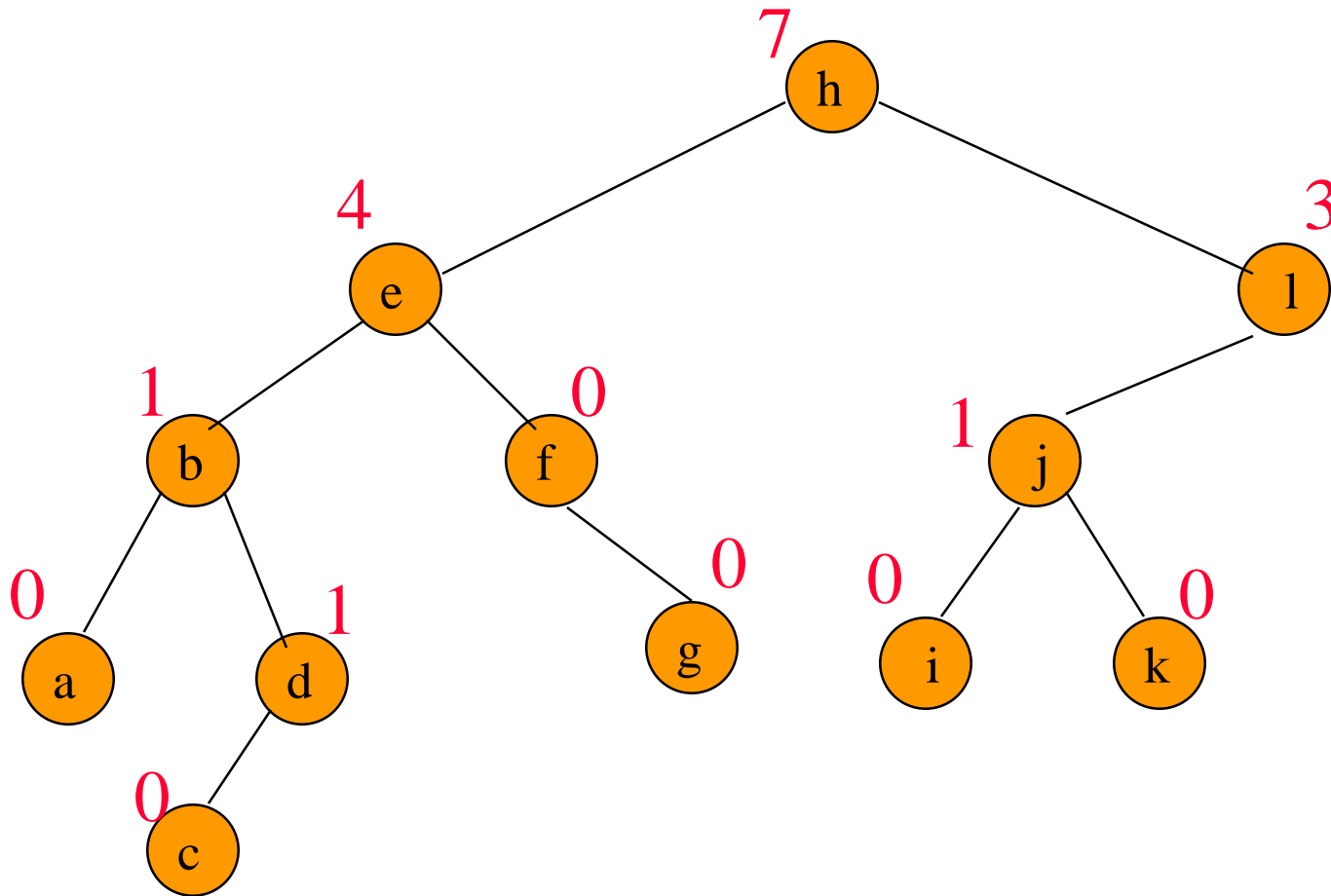
жагсаалт = [a,b,c,d,e,f,g,h,i,j,k,l]

add(5,'m')



жагсаалт = [a,b,c,d,e,f,g,h,i,j,k,l]

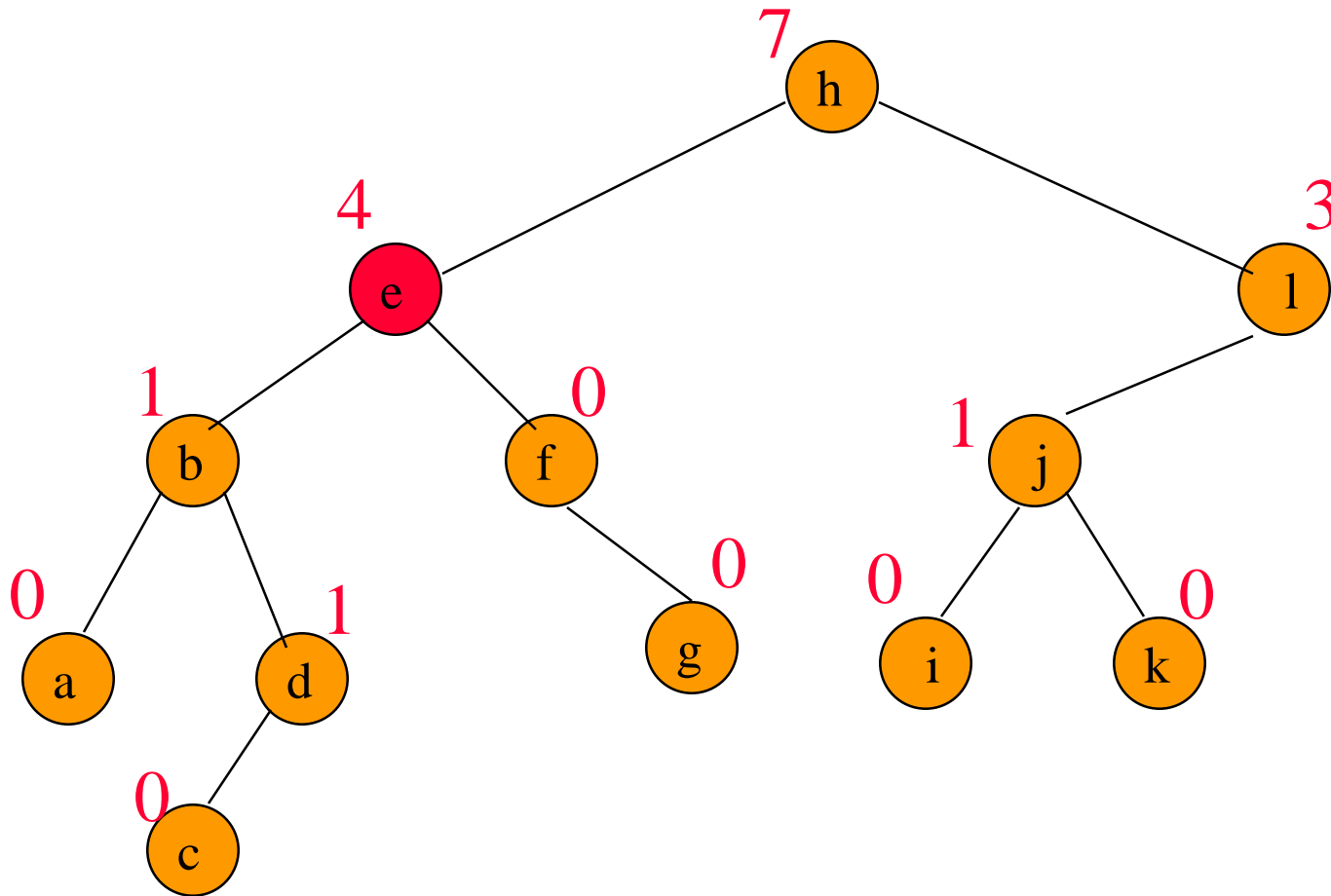
add(5,'m')



жагсаалт = [a,b,c,d,e, m,f,g,h,i,j,k,l]

элемент 4 (e) —г олох

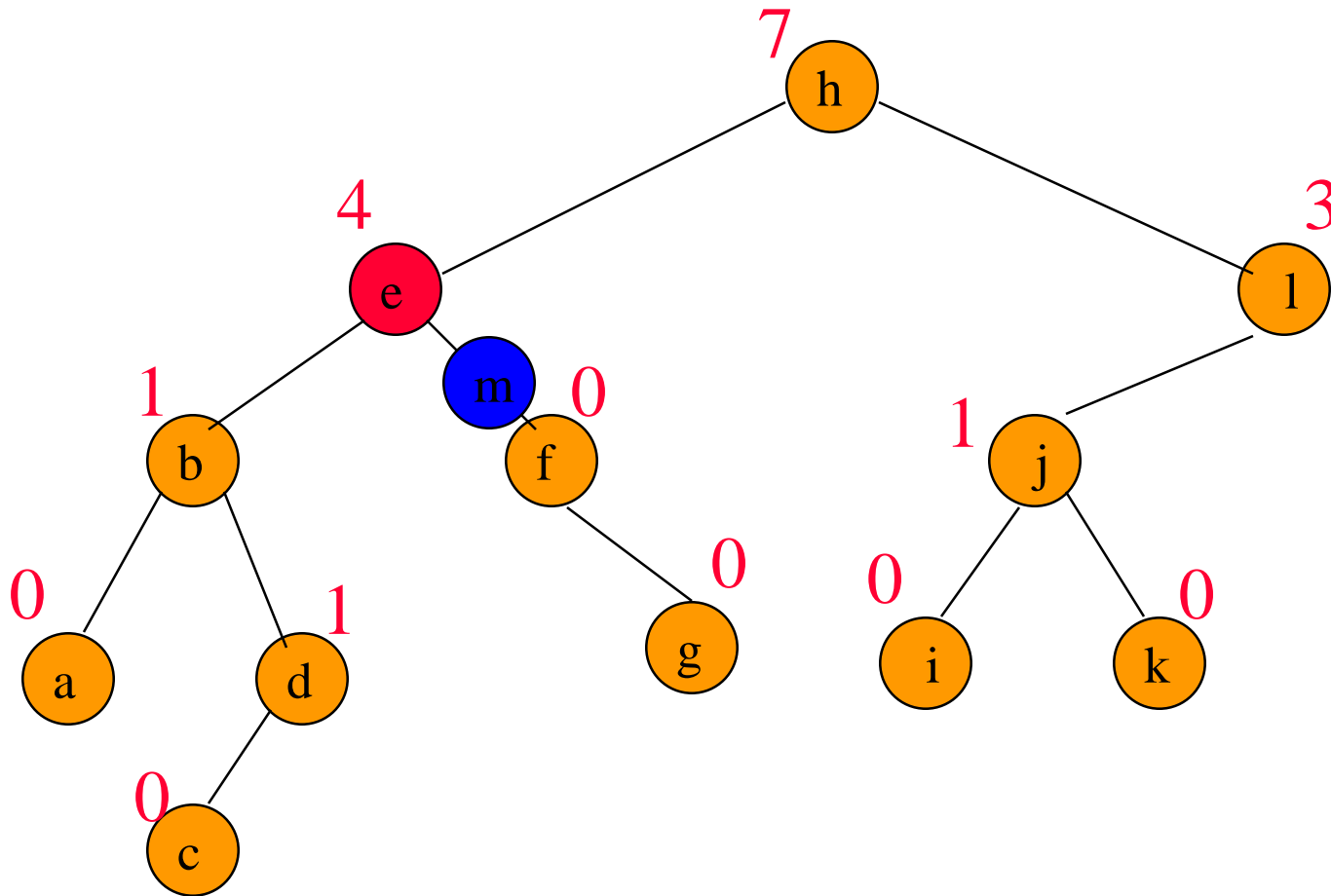
add(5,'m')



жагсаалт = [a,b,c,d,e, m,f,g,h,i,j,k,l]

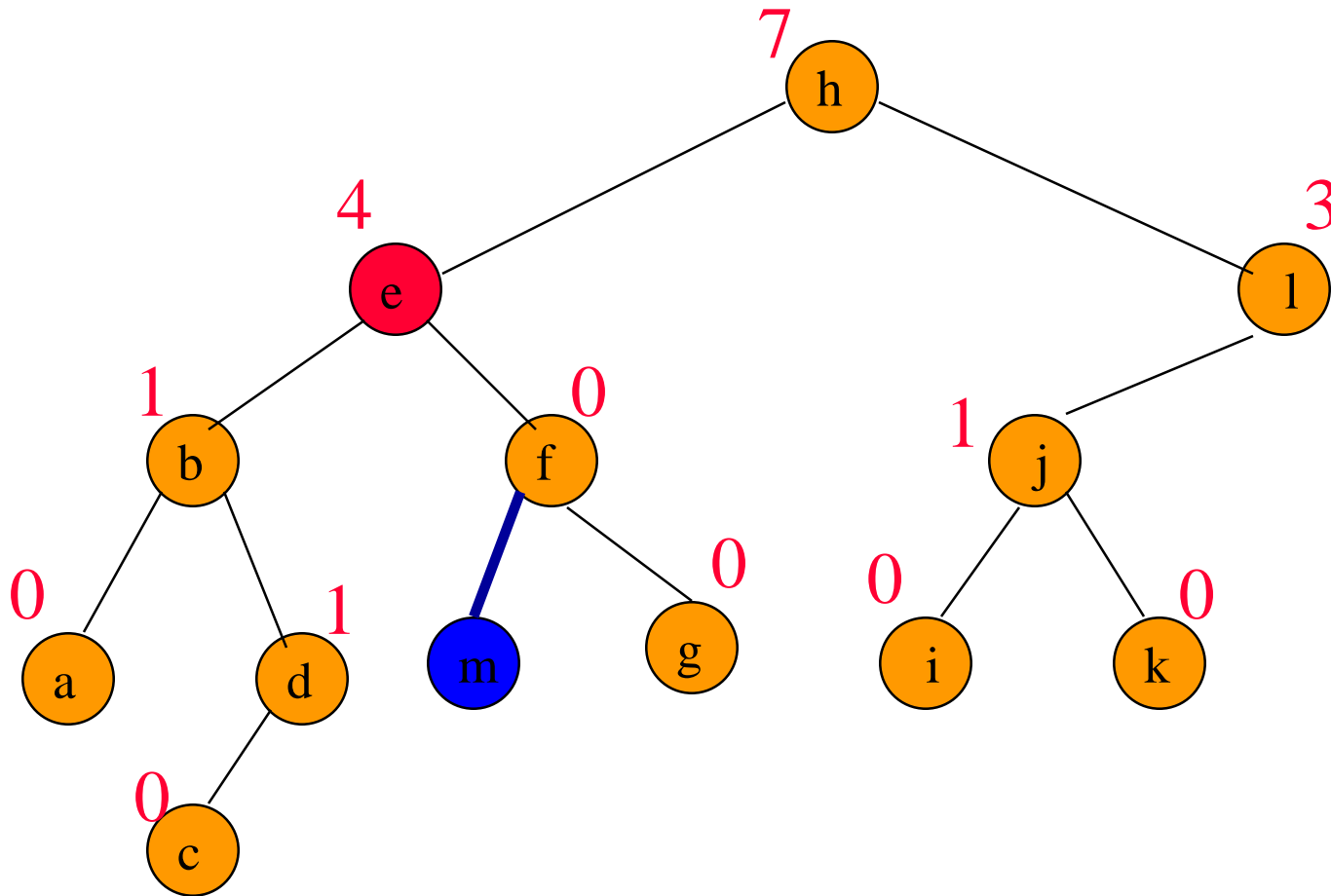
элемент 4 (e) —г олох

add(5, 'm')



m -г **e** -ийн баруун хүүгээр оруулах; **e** –ийн хуучин баруун дэд мод **m** –ийн баруун дэд мод болно

add(5, 'm')

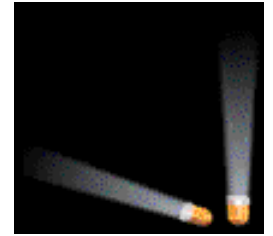
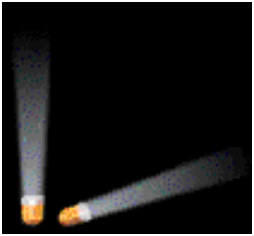


m –ийг **e** –ийн баруун дэд модны хамгийн зүүн элементээр нэмнэ

add(5, 'm')

- Өөр боломжууд бий.
- Үндсээс шинэ зангилаа хүртэлх замын **leftSize** утгуудыг шинэчлэнэ.
- Хугацаа **$O(\text{өндөр})$** .

Тэнцвэртэй хоёртын хайлтын мод



- Өндөр $O(\log n)$, үүнд n –модны элементийн тоо
- AVL (Adelson-Velsky and Landis) мод
- Улаан-хар мод
- get , put , ба $remove$ үйлдлүүд $O(\log n)$ хугацаанд хийгдэнэ

Тэнцвэртэй хоёртын хайлтын мод

- Индексстэй AVL мод
- Индексстэй улаан-хар мод
- Индексстэй үйлдлүүд

$O(\log n)$ хугацаанд хийгдэнэ

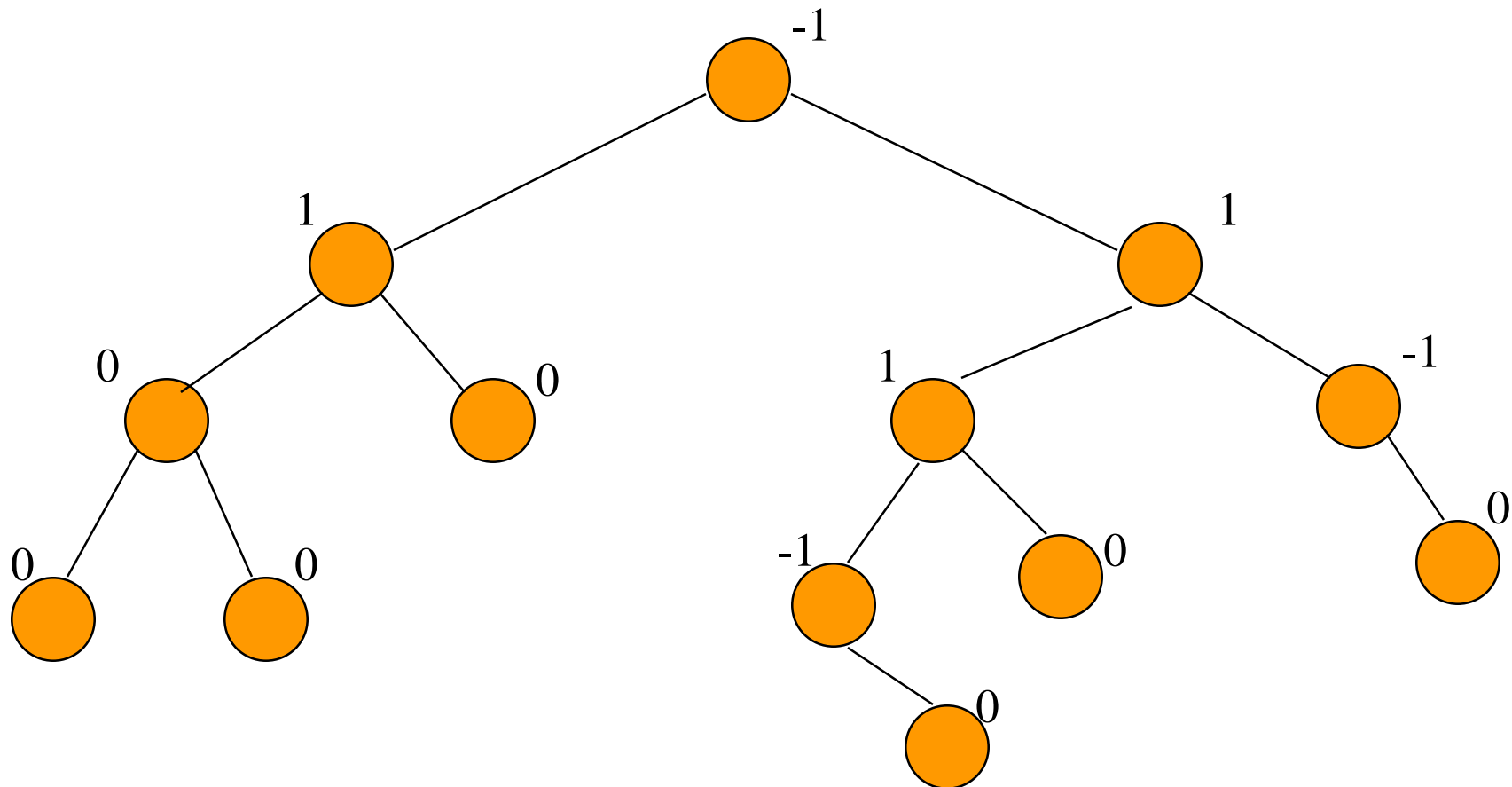
Тэнцвэртэй хайлтын мод

- Жингээр тэнцвэржсэн хоёртын хайлтын мод
- 2-3 ба 2-3-4 мод
- АА мод
- В-мод
- BBST
- Г.М.

AVL мод

- Хоёртын мод
- Аливаа зангилаа x -ийн хувьд тэнцвэржилтын коэффициентийг олно
 x –ийн тэнцвэржилтийн коэф = x –ийн зүүн дэд модны өндөр - x –ийн аруун дэд модны өндөр
- коэффициент -1 , 0 , эсхүл 1 байна

Тэнцвэржилтийн коэффициент



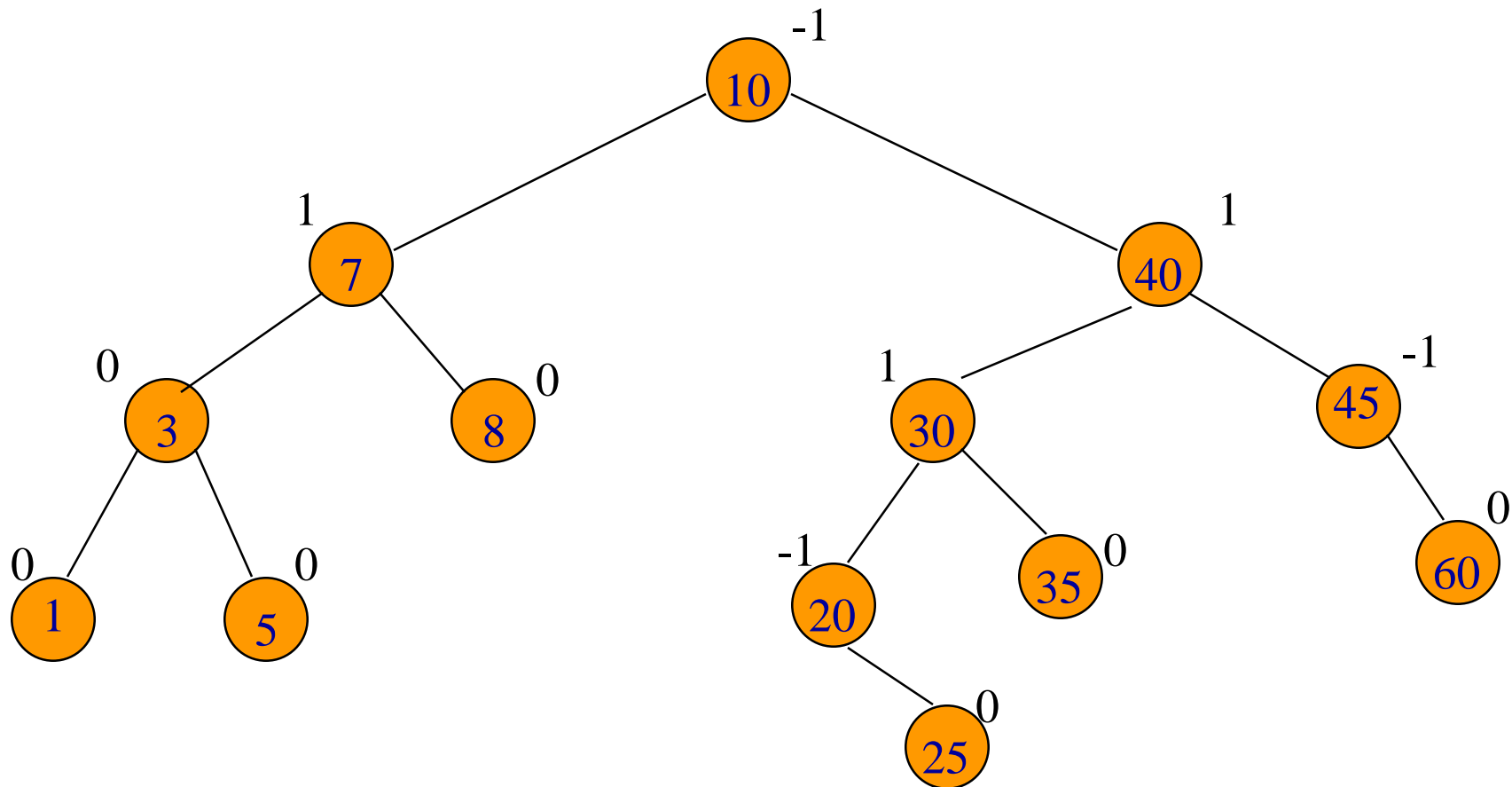
Энэ бол AVL мод.

Өндөр

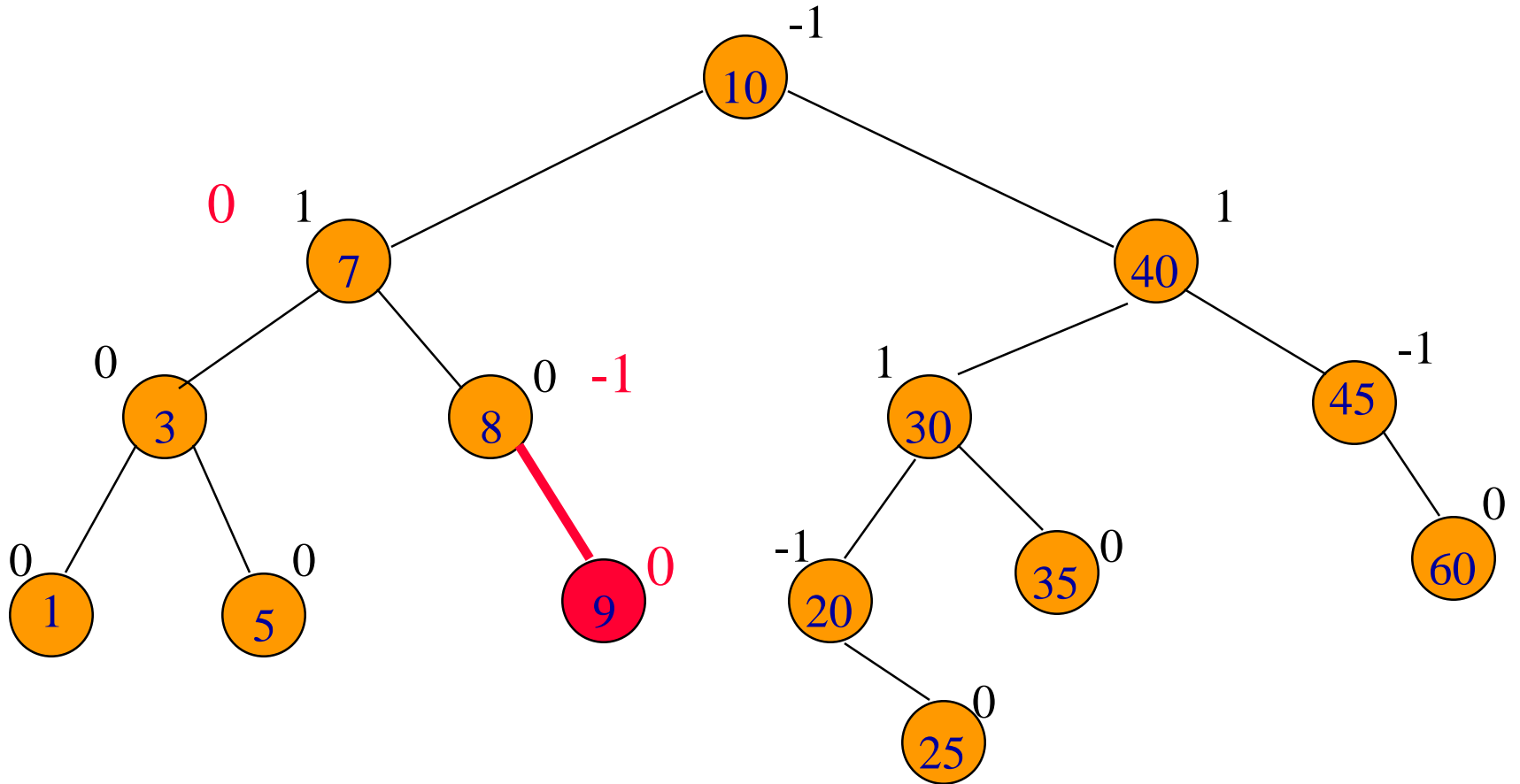
N зангилаатай AVL модны өндөр хамгийн ихдээ $1.44 \log_2 (n+2)$.

Аливаа n зангилаатай хоёртын модны өндөр дор хаяж $\log_2 (n+1)$.

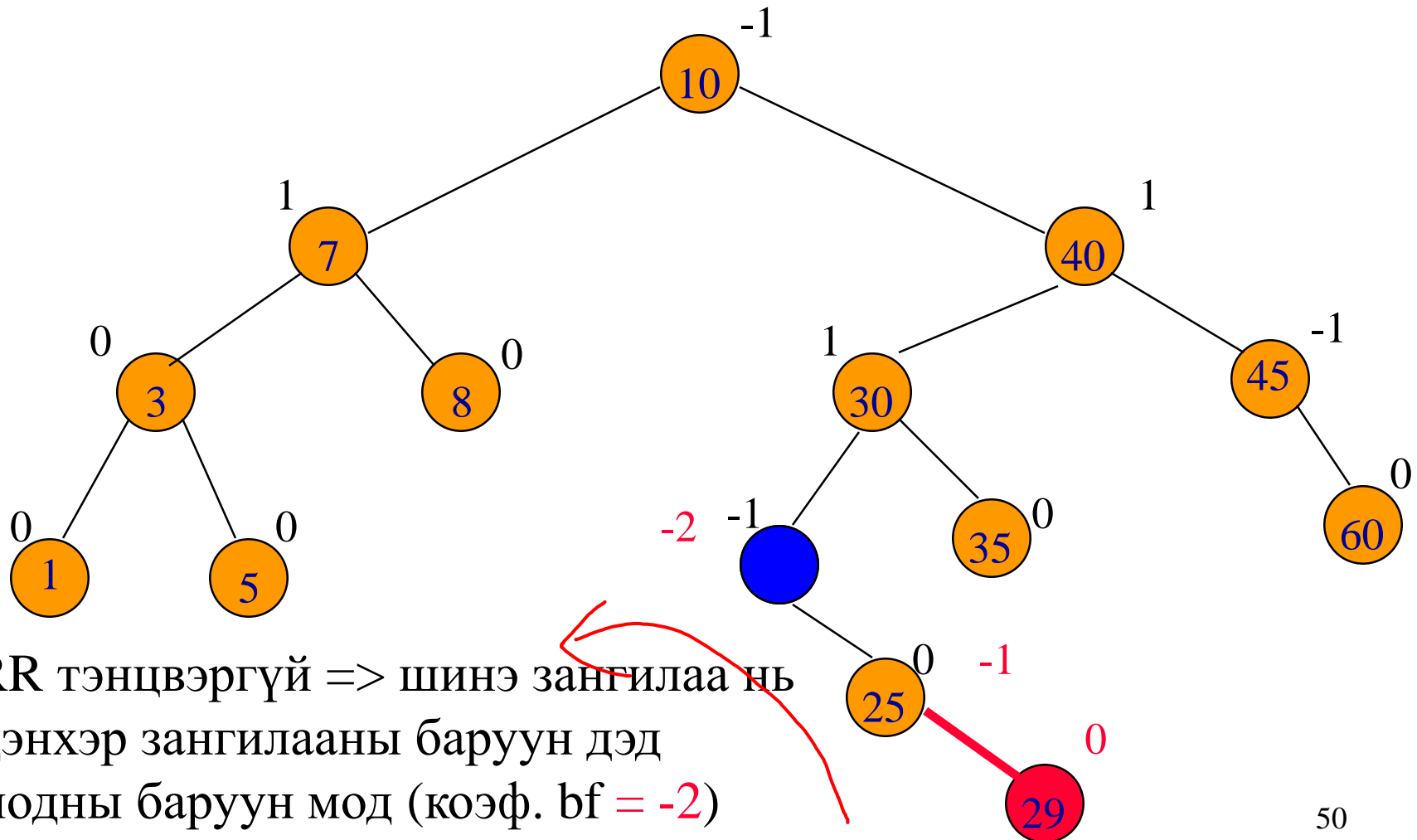
AVL хайлтын мод



put(9)

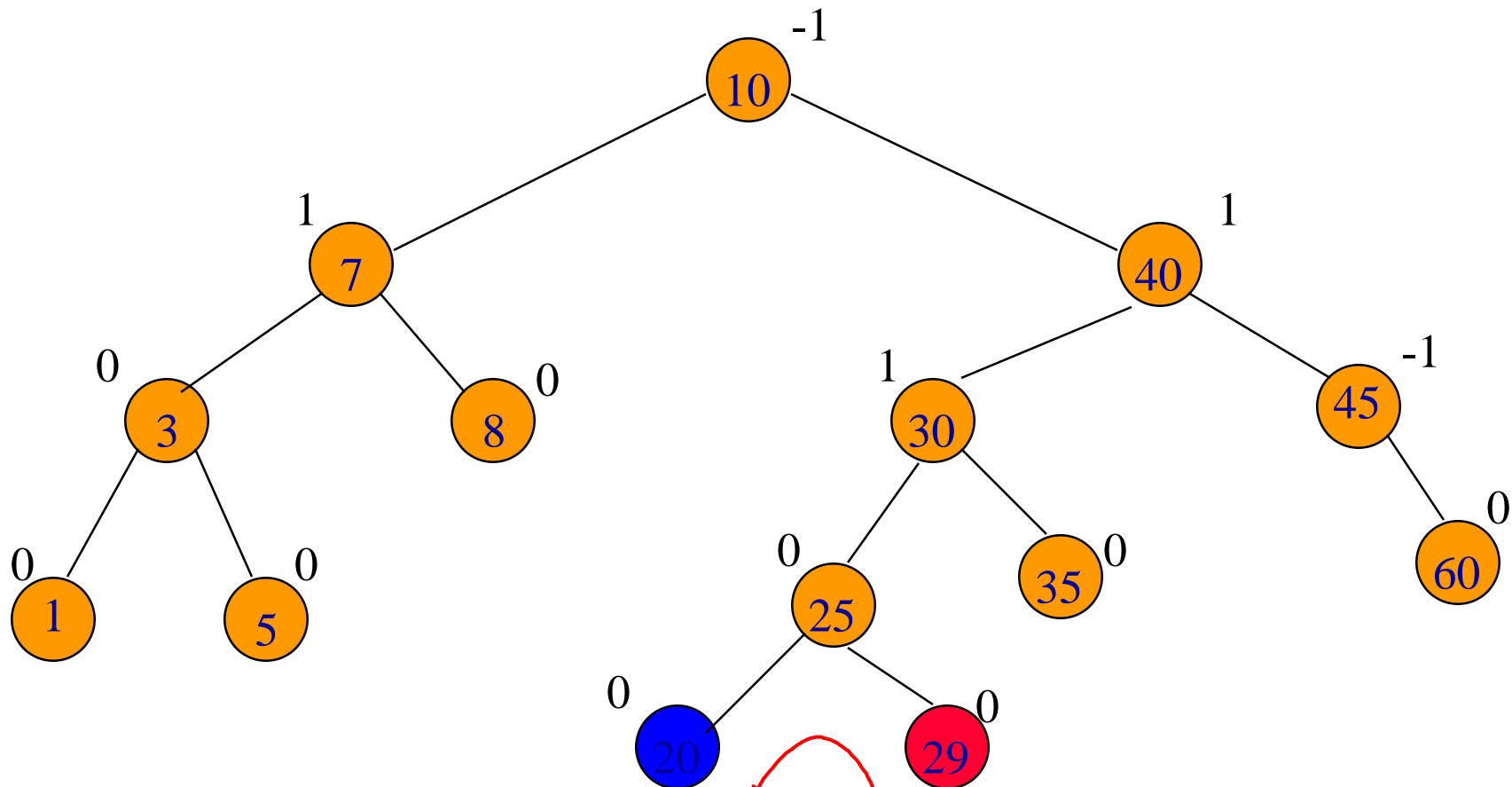


put(29)



RR тэнцвэргүй => шинэ зангилаа нь
цэнхэр зангилааны баруун дэд
модны баруун мод (коэф. bf = -2)

put(29)



RR эргүүлэлт.

AVL эргүүлэлт

- RR
- LL
- RL
- LR

<http://webpages.ull.es/users/jriera/Docencia/AVL/AVL%20tree%20applet.htm>

<http://www.cs.queensu.ca/home/jstewart/applets/bst/bst-rotation.html>

Эргүүлэлт

There are 4 cases in all, choosing which one is made by seeing the direction of the first 2 nodes from the unbalanced node to the newly inserted node and matching them to the top most row.

Root is the initial parent before a rotation and **Pivot** is the child to take the root's place.

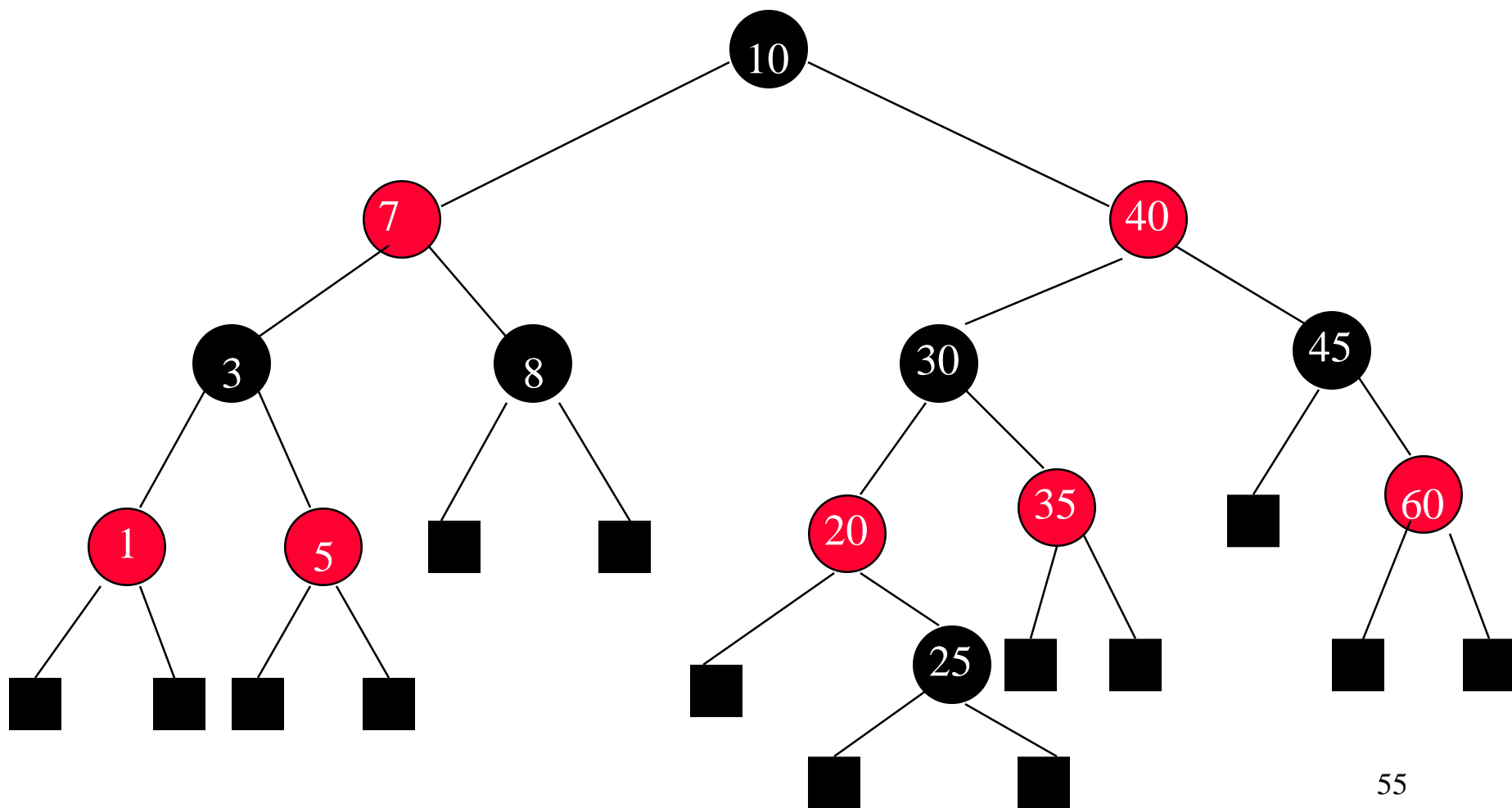
<p>Left Left Case</p> <p>Root 5 Pivot 3</p> <p>Right Rotation</p>	<p>Right Right Case</p> <p>Root 3 Pivot 5</p> <p>Left Rotation</p>	<p>Left Right Case</p> <p>Root 5 Pivot 4</p> <p>Left Rotation</p>	<p>Right Left Case</p> <p>Root 3 Pivot 4</p> <p>Right Rotation</p>
		<p>Root 5 Pivot 4</p> <p>Right Rotation</p>	<p>Root 3 Pivot 4</p> <p>Left Rotation</p>

Улаан хар мод

Будагтай загилааны тодорхойлолт

- Хоёртын хайлтын мод.
- Зангилаа бүрийг **улаан** , хар өнгөөр будна.
- Үндэс болон бүх гадны зангилаа хар өнгөтэй.
- Дараалсан хоёр **улаан** зангилаа орсон үндсээс гадны зангилаа хүрсэн зам байхгүй.
- Үндсээс гадны гадны зангилаа хүрсэн бүх замууд ижил тооны хар зангилаатай

Улаан хар модны жишээ

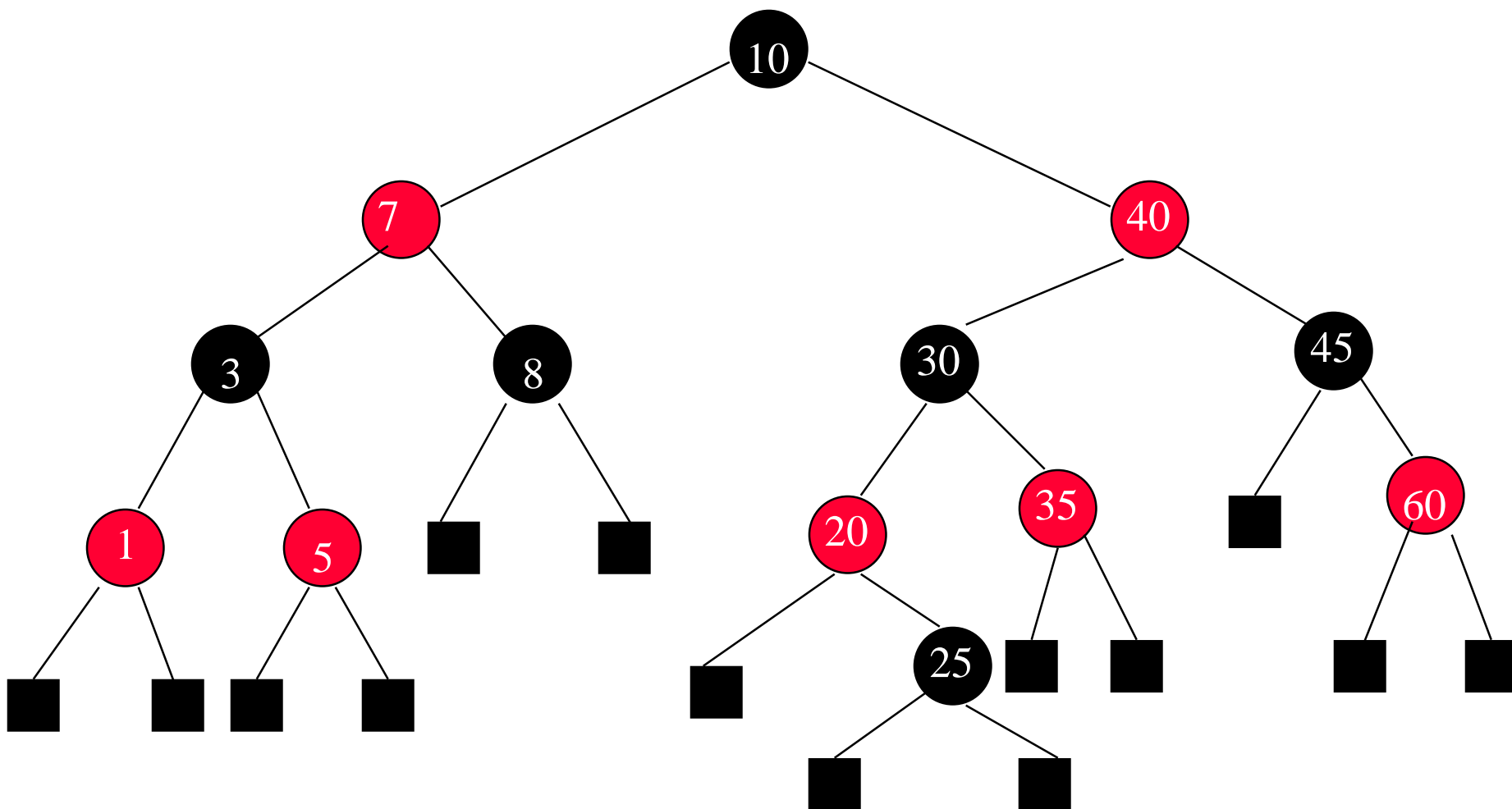


Улаан хар мод

Будагтай ирмэгийн тодорхойлолт

- Хоёртын хайлтын мод.
- Хүүгийн заагчийг **улаан** эсхүл хараар будна.
- Гадны зангилааг заагчийг хараар будна.
- Хоёр дараалсан **улаан** заагчтай үндсээс гадны зангилаа хүрсэн зам байхгүй.
- Үндсээс гадны зангилаа хүрсэн зам бүр ижил тооны хар заагчтай байна.

Улаан хар модны жишээ



Улаан хар мод

- n (дотоод) зангилаатай улаан хар модны өндөр $\log_2(n+1)$ ба $2\log_2(n+1)$ -ийн хооронд байна
- `java.util.TreeMap` => улаан хар мод