

Эрэмбэтэй(Priority) дараалал



Хоёр төрлийн эрэмбэтэй дараалал байдаг:

- Минимум эрэмбэтэй дараалал.
- Максимум эрэмбэтэй дараалал.

Минимум эрэмбэтэй дараалал

- Элементүүдийн цуглуулга.
- Элемент бүр эрэмбэ буюу түлхүүртэй.
- Хийгдэх үйлдлүүд:
 - isEmpty
 - size
 - add/put (эрэмбэтэй дараалал элемент оруулах)
 - get (min эрэмбэтэй элементийг авах)
 - remove (min эрэмбэтэй элементийг устгах)

Максимум эрэмбэтэй дараалал

- Элементүүдийн цуглуулга.
- Элемент бүр эрэмбэ буюу түлхүүртэй.
- Хийгдэх үйлдлүүд:
 - isEmpty
 - size
 - add/put (эрэмбэтэй дараалал элемент оруулах)
 - get (max эрэмбэтэй элементийг авах)
 - remove (max эрэмбэтэй элементийг устгах)

Үйлдлийг гүйцэтгэх хугацаа

Сайн хэрэгжүүлэлт нь heap-
овоолго , leftist tree-зүүний мод.

isEmpty, size, get $\Rightarrow O(1)$

put , remove $\Rightarrow O(\log n)$, үүнд n –
эрэмбэтэй дарааллын хэмжээ

Хэрэглээ

Эрэмбэлэлт

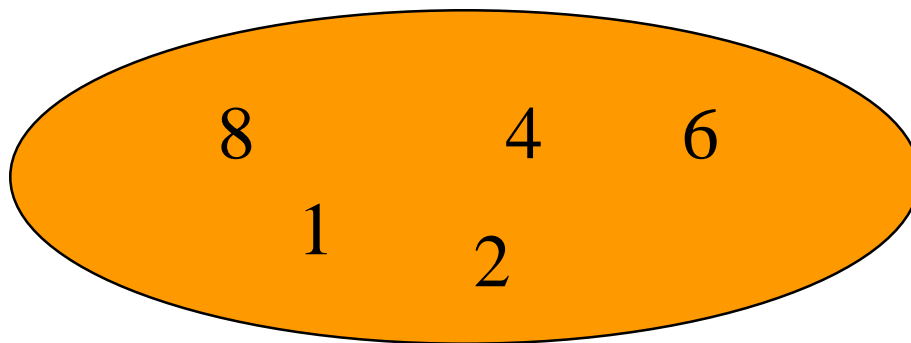
- Элементийн түлхүүрийг эрэмбэ болгон ашиглах
- Эрэмбэлэх элементүүдийг эрэмбэтэй дараалалд хийх
- Эрэмбийн дараалаар элементийг гаргаж авах
 - Хэрвээ **min** эрэмбэтэй дараалал бол, элементүүдийг эрэмбийн(түлхүүрийн) өсөх дарааллаар гаргана
 - Хэрвээ **max** эрэмбэтэй дараалал бол, элементүүдийг эрэмбийн(түлхүүрийн) буурах дарааллаар гаргана

Эрэмбэлэлтийн жишээ

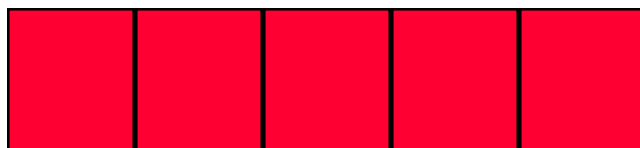
6, 8, 2, 4, 1 гэсэн түлхүүртэй таван элементийг тах эрэмбэтэй дарааллын аргаар эрэмбэлэе.

- Таван элементийг тах эрэмбэтэй дараалалд хийнэ.
- `remove` тах үйлдлийг таван удаа гүйцэтгэхдээ устгагдсан элементүүдийг эрэмбэлэсэн массивт варуунаас зүүн тийш хийнэ.

Мах эрэмбэтэй дараалалд хийсний дараа

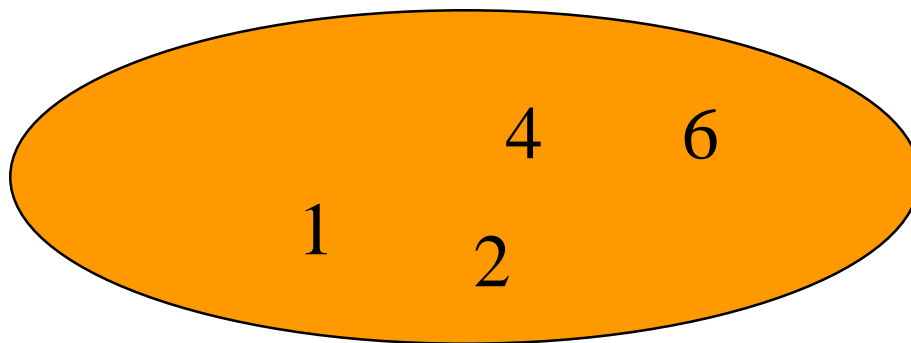


Мах
эрэмбэтэй
дараалал

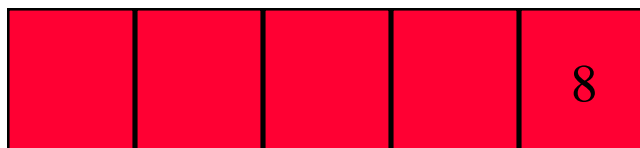


Эрэмбэлэгдсэн массив

Эхний Remove Max үйлдлийн дараа

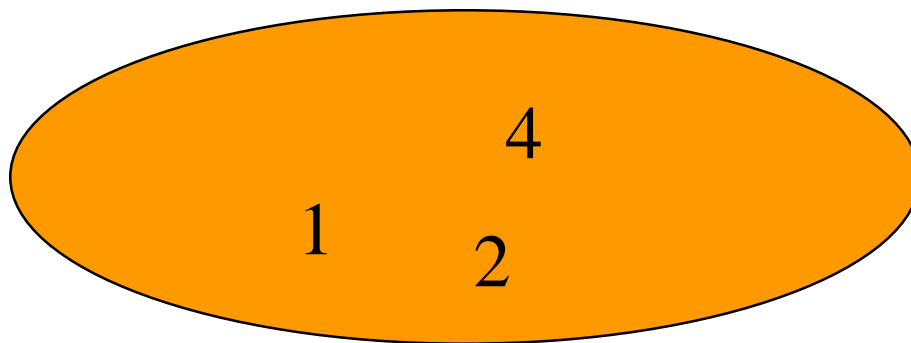


Max
эрэмбэтэй
дараалал

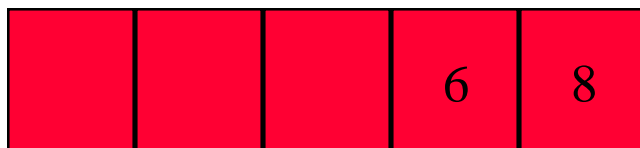


Эрэмбэлэгдсэн массив

Хоёр дахь Remove Max үйлдлийн дараа

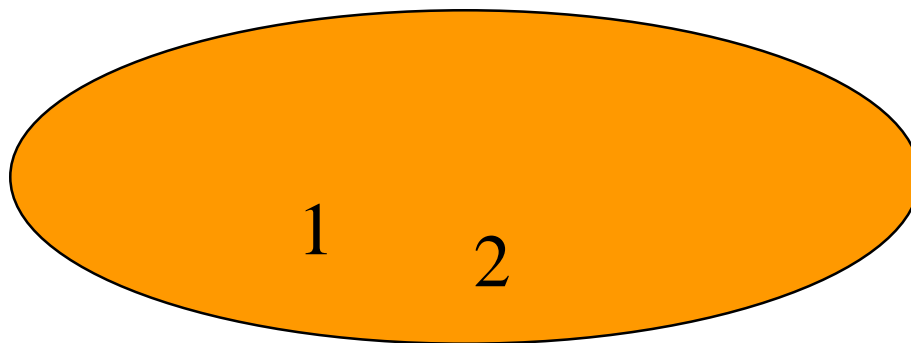


Max
эрэмбэтэй
дараалал

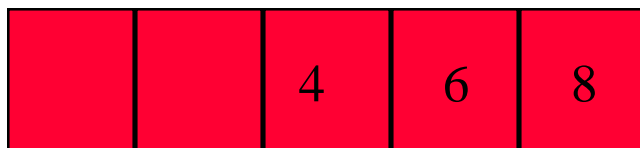


Эрэмбэлэгдсэн массив

Гурав дахь Remove Max үйлдлийн дараа

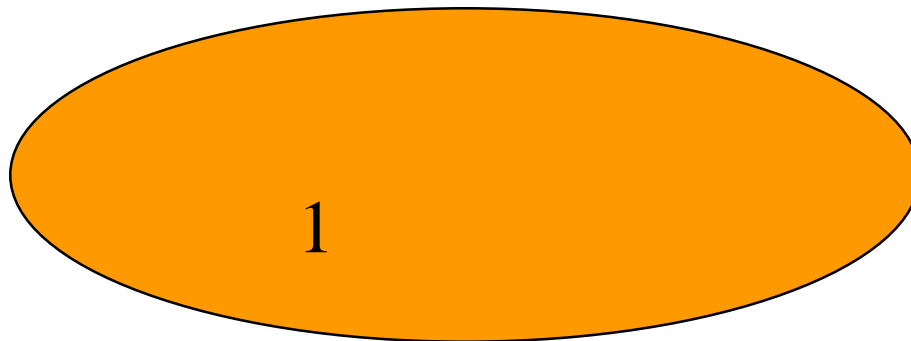


Max
эрэмбэтэй
дараалал

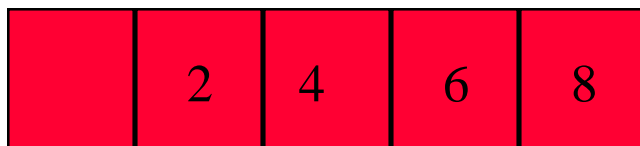


Эрэмбэлэгдсэн массив

Дөрөв дэх Remove Max үйлдлийн дараа

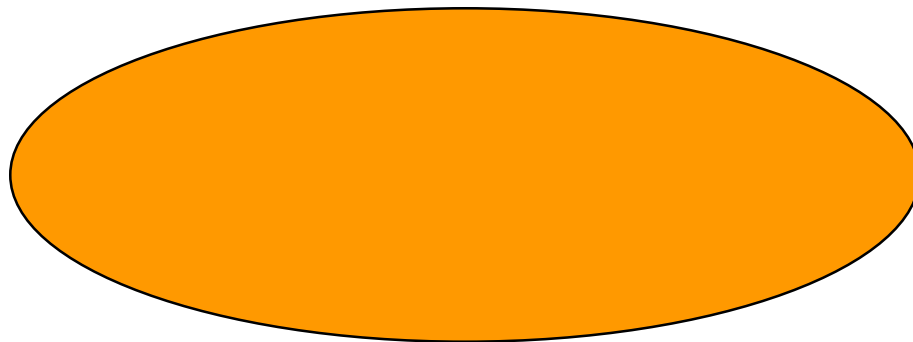


Мах
эрэмбэтэй
дараалал



Эрэмбэтэй массив

Тав дахь Remove Max үйлдлийн дараа



Max
эрэмбэтэй
дараалал

1	2	4	6	8
---	---	---	---	---

Эрэмбэлэгдсэн массив

Эрэмбэлэлтийн хугацаа

n элементийн эрэмбэлэлт.

- **n** put үйлдэл $\Rightarrow O(n \log n)$.
- **n** remove max үйлдэл $\Rightarrow O(n \log n)$.
- Нийт хугацаа $O(n \log n)$.
- 2-р бүлэгт үзсэн эрэмбэлэлтийн аргаар $O(n^2)$.

Овоолго(Heap)-ын Эрэмбэлэлт

Овоолгыг хэрэгжүүлэхдээ тах эрэмбэтэй дарааллыг ашиглана.

Эхний put үйлдлүүдийг овоолгыг идэвхижүүлэх алхмаар орлуулахад $O(n)$ хугацаа шаардана.

Машины төлөвлөлт

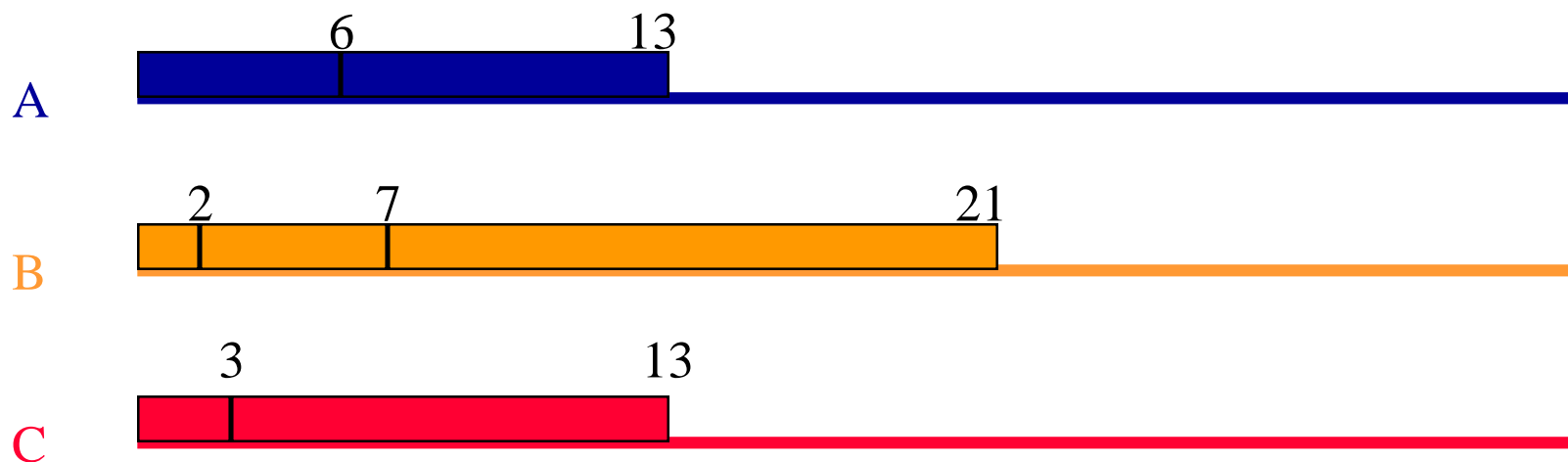
- m ижил төрлийн машин (өрөм, зүсэгч, г.м.)
- n ажил гүйцэтгэх хэрэгтэй
- Машиныудыг ажлаар ачаалахдаа сүүлийн ажлыг гүйцэтгэх хугацааг минимум болгох

Машины төлөвлөлтийн жишээ

3 машин, 7 ажил

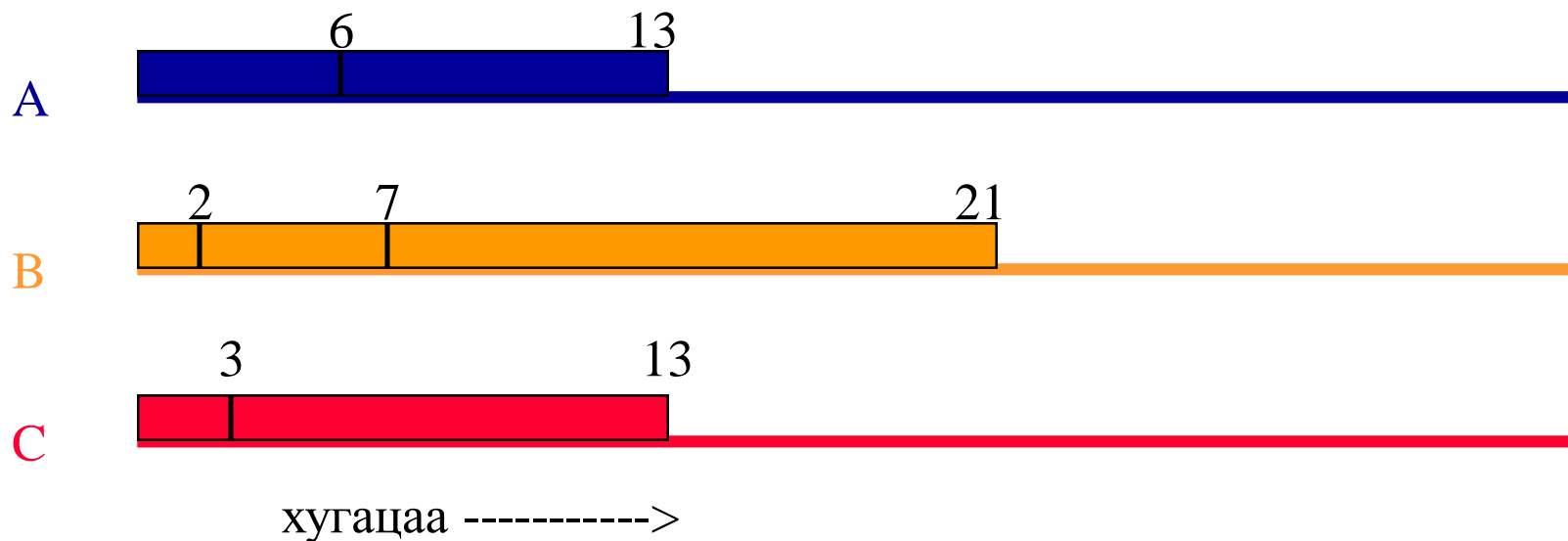
Ажлын хугацаа [6, 2, 3, 5, 10, 7, 14]

Боломжит төлөвлөлт:



хугацаа ----->

Машины төлөвлөлтийн жишээ



Дуусах хугацаа = 21

Зорилго: Дуусах хугацааг минимум болгох.

LPT Төлөвлөлт

Longest Processing Time - эхлээд.

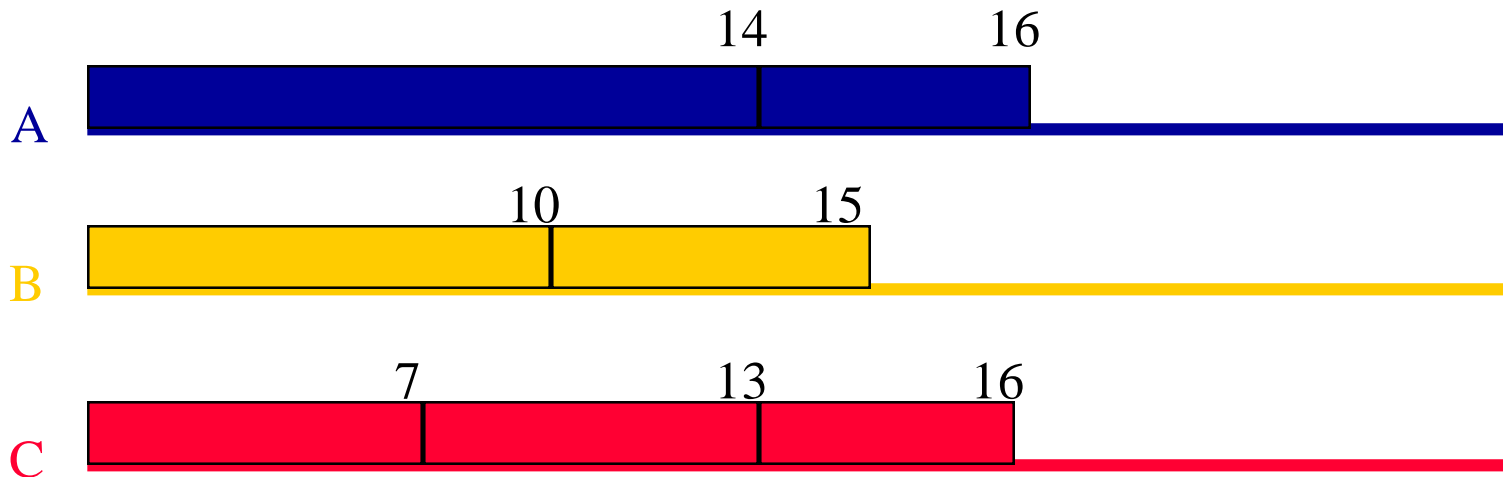
Ажлууд дараах байдлаар эрэмбэлэгджээ

14, 10, 7, 6, 5, 3, 2

Ажил бүрийг түрүүлж дуусгасан машин
дээр хуваарилна.

LPT төлөвлөлт

[14, 10, 7, 6, 5, 3, 2]



Дуусах хугацаа 16!

LPT төлөвлөлт

- LPT дүрэм баталгаат минимум дуусах хугацааг өгч чадахгүй.
- $(\text{LPT дуусах хугацаа})/(\text{Min дуусах хугацаа}) \leq 4/3 - 1/(3m)$ үүнд m - машины тоо.
- Ерөнхийдөө LPT дуусах хугацаа, min дуусах хугацаад ойрхон очдог .
- Min дуусах хагацаатай төлөвлөлтийг NP (nondeterministic polynomial)-hard өгдөг.

LPT төлөвлөлтийн хугацаа

- Ажлуудыг хугацаа буурах дарааллаар эрэмбэлнэ.
 - $O(n \log n)$ хугацаа (n ажлын тоо)
- Ажлуудыг энэ дарааллаар хуваарилна.
 - Ажлыг түрүүлж бэлэн болсон машинд хуваарилна
 - m (m машины тоо) дуусах хугацаанаас minimum-г олох ёстой
 - Энгийн стратеги хэрэглэхэд $O(m)$ хугацаа зарна
 - Бүх n ажлыг төлөвлөхөд $O(mn)$ хугацаа орно.

Min эрэмбэтэй дарааллыг ашиглах

- Min эрэмбэтэй дараалалд **m** машины дуусах хугацаа байна.
- Эхлээд бүх дуусах хугацаа **0**.
- Ажлыг төлөвлөхийн тулд эрэмбэтэй дарааллаас минимум дуусах хугацаатай машиныг устгана.
- Сонгосон машины дуусах хугацааг өөрчлөөд буцаагаад эрэмбэтэй дараалалд хийнэ.

Min эрэмбэтэй дарааллыг ашиглах

- m put үйлдлээр эрэмбэтэй дарааллыг идэвхижүүлнэ
- 1 remove min , 1 put үйлдлүүдийг хуваарилагдсан ажил бүрт хийнэ
- put , remove min үйлдэл бүрт $O(\log m)$ хугацаа орно
- Төлөвлөх хугацаа $O(n \log m)$
- Нийт хугацаа

$$O(n \log n + n \log m) = O(n \log (mn))$$

Хоффманы код

Хаягдалгүй шахахад тустай.

LZW аргатай хослуулж болно.

Номноос унш.

Товч тайлбар:

{a, x, u, z}

a-0, x-10, u-110, z-111 (давтамжаар хувьсах урттай код)

аахиахz (7 байт) -> 0010110010111 (13 бит)

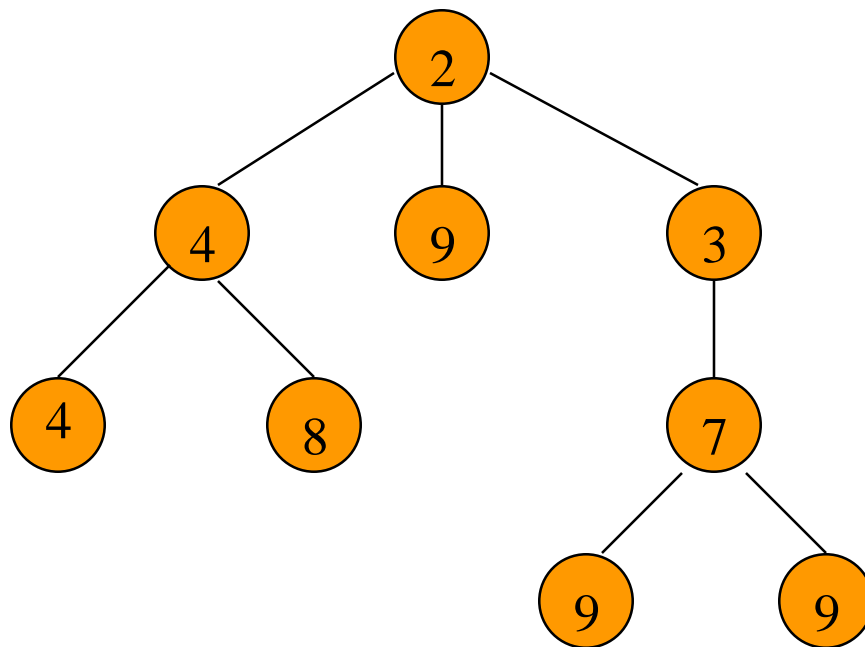
Min модны тодорхойлолт

Модны зангилаа бүр утгатай.

Аливаа зангилааг үндэс гэж үзвэл утга нь
дэд модондоо минимум байна.

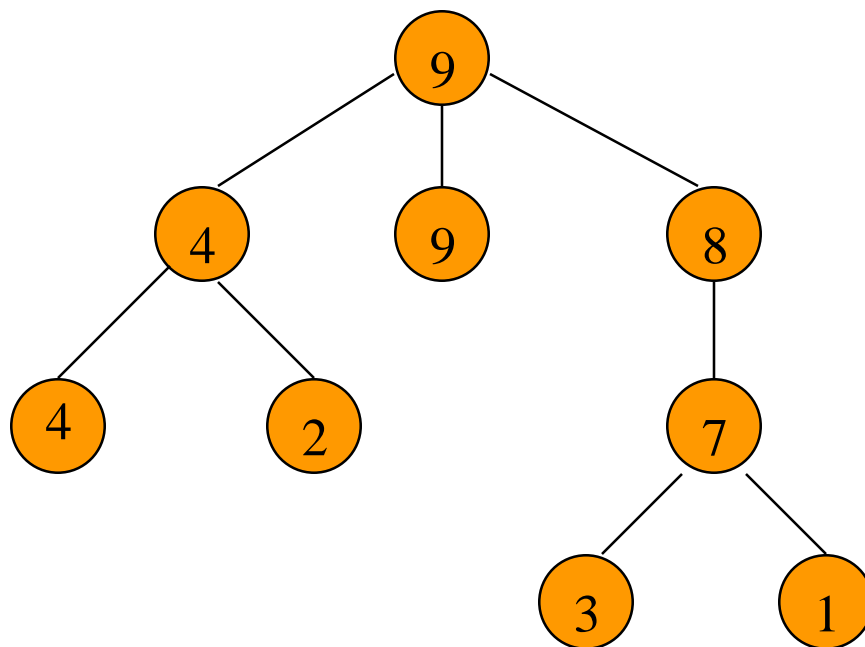
Өөрөөр хэлбэл бага утгатай хойч байхгүй.

Min модны жишээ



Үндэс минимум элементтэй.

Мах модны жишээ

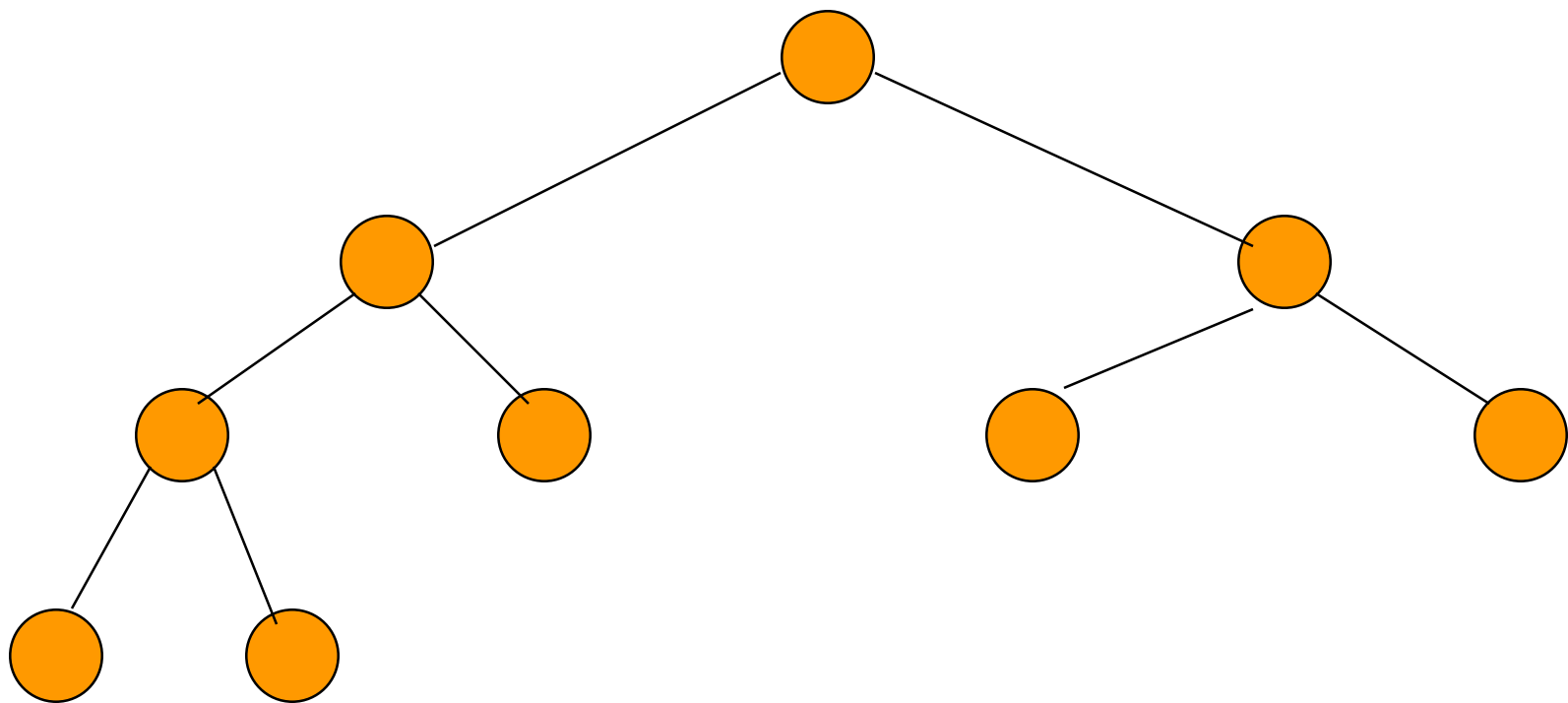


Үндэс максимум элементтэй.

Min овоолгын тодорхойлолт

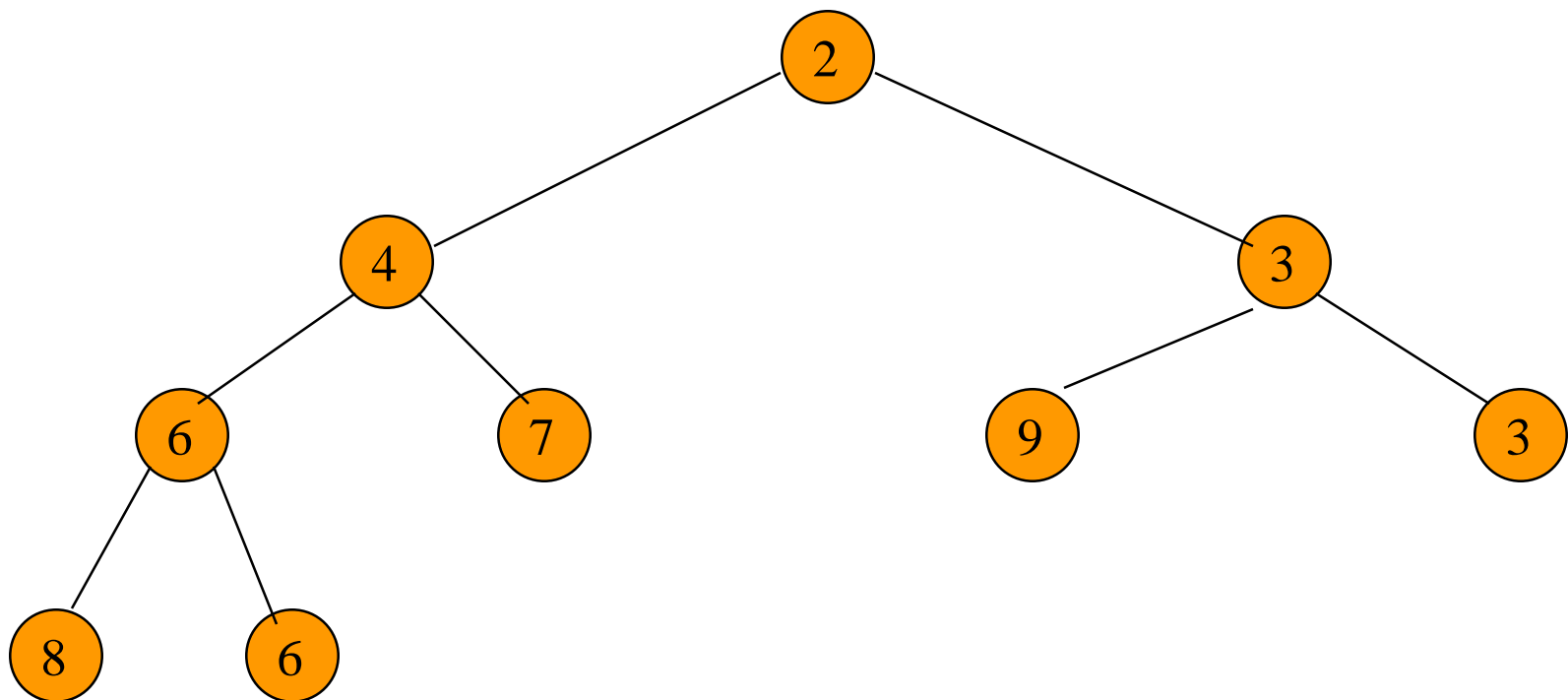
- Төгс хоёртын мод
- min мод

9 зангилаатай Min овоолго



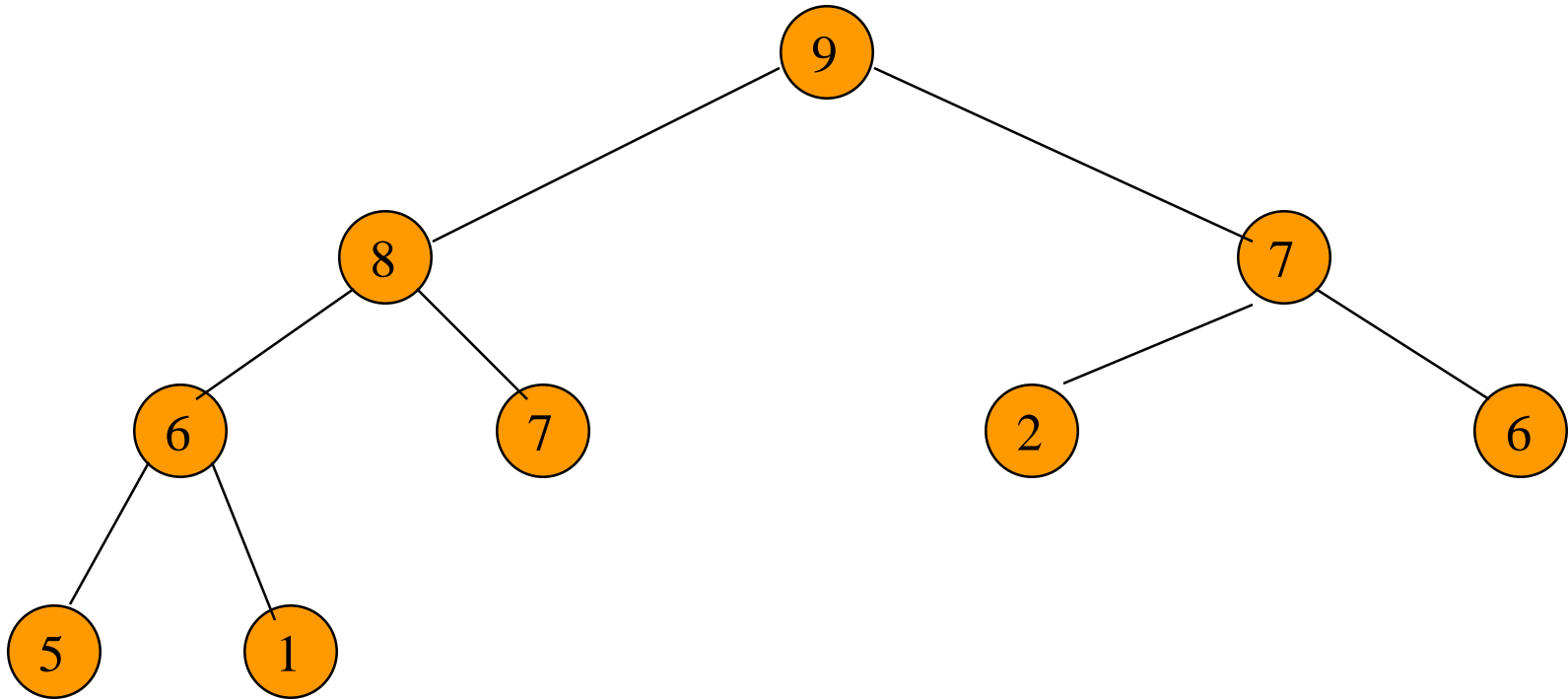
9 зангилаатай төгс хоёртын мод.

9 зангилаатай Min овоолго



9 зангилаатай төгс хоёртын мод
мөн min мод.

9 зангилаатай Мах овоолго

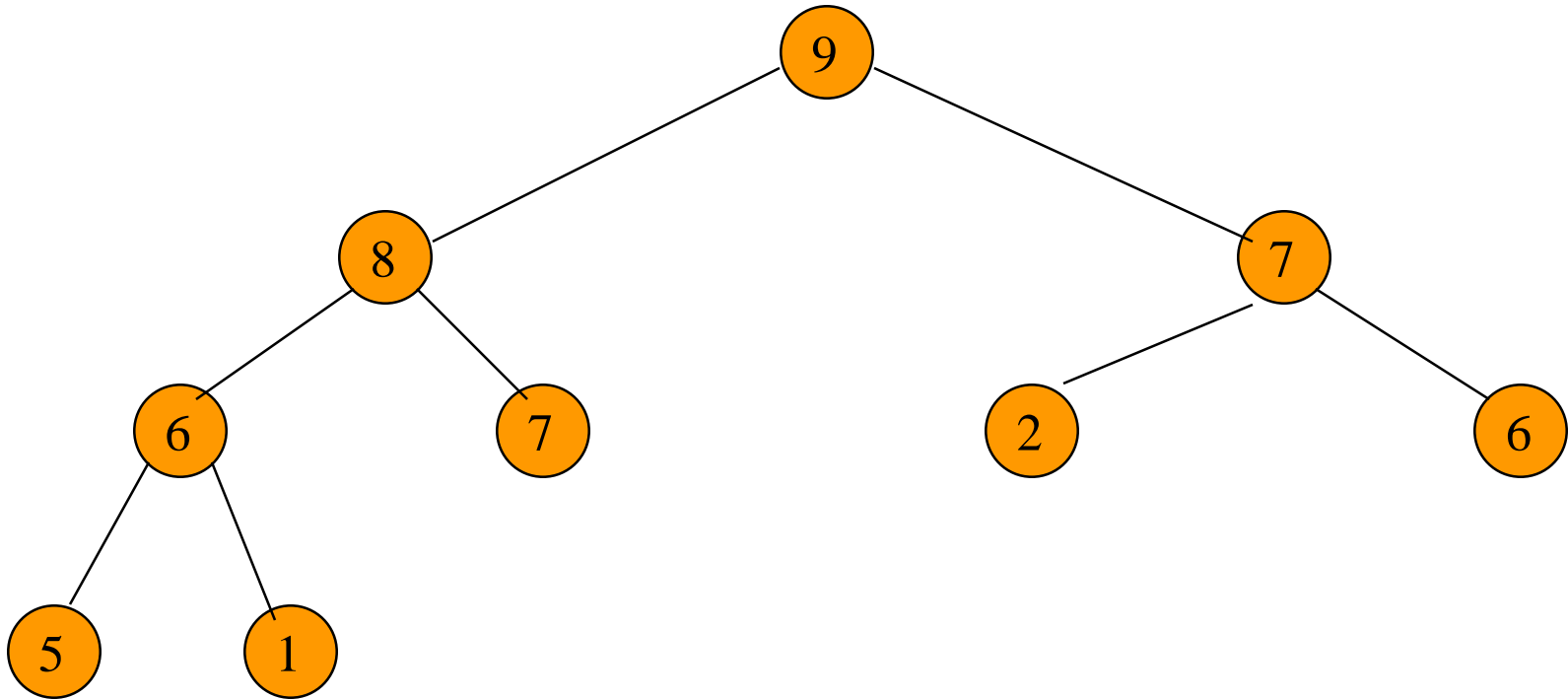


9 зангилаатай төгс хоёртын мод
мөн тах мод.

Овоолгын өндөр

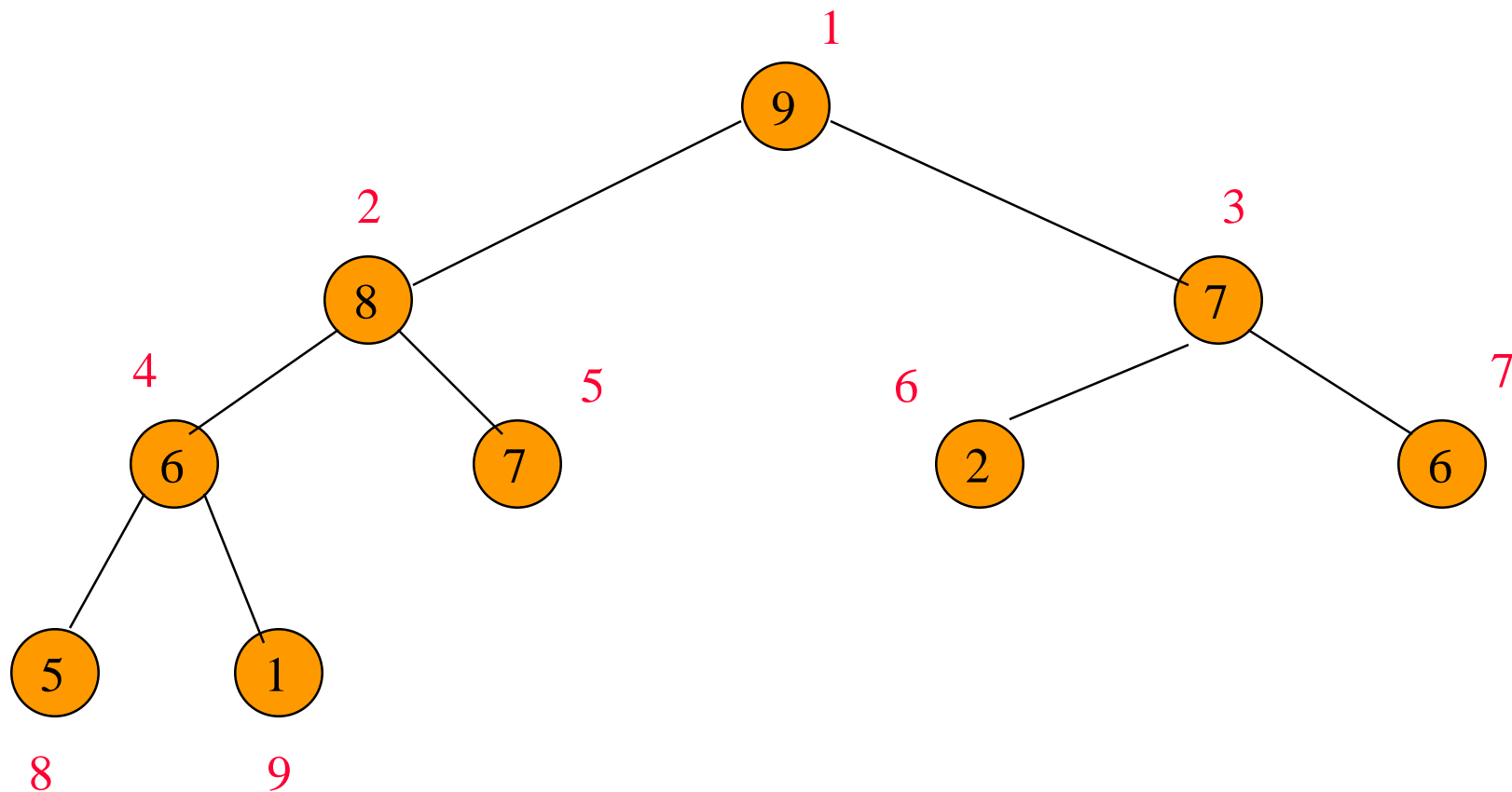
Нэгэнт овоолго нь төгс хоёртын
мод болохоор n зангилаатай
овоолгын өндөр $\log_2 (n+1)$.

Овоолгыг массиваар оновчтой дүрсэлж болно

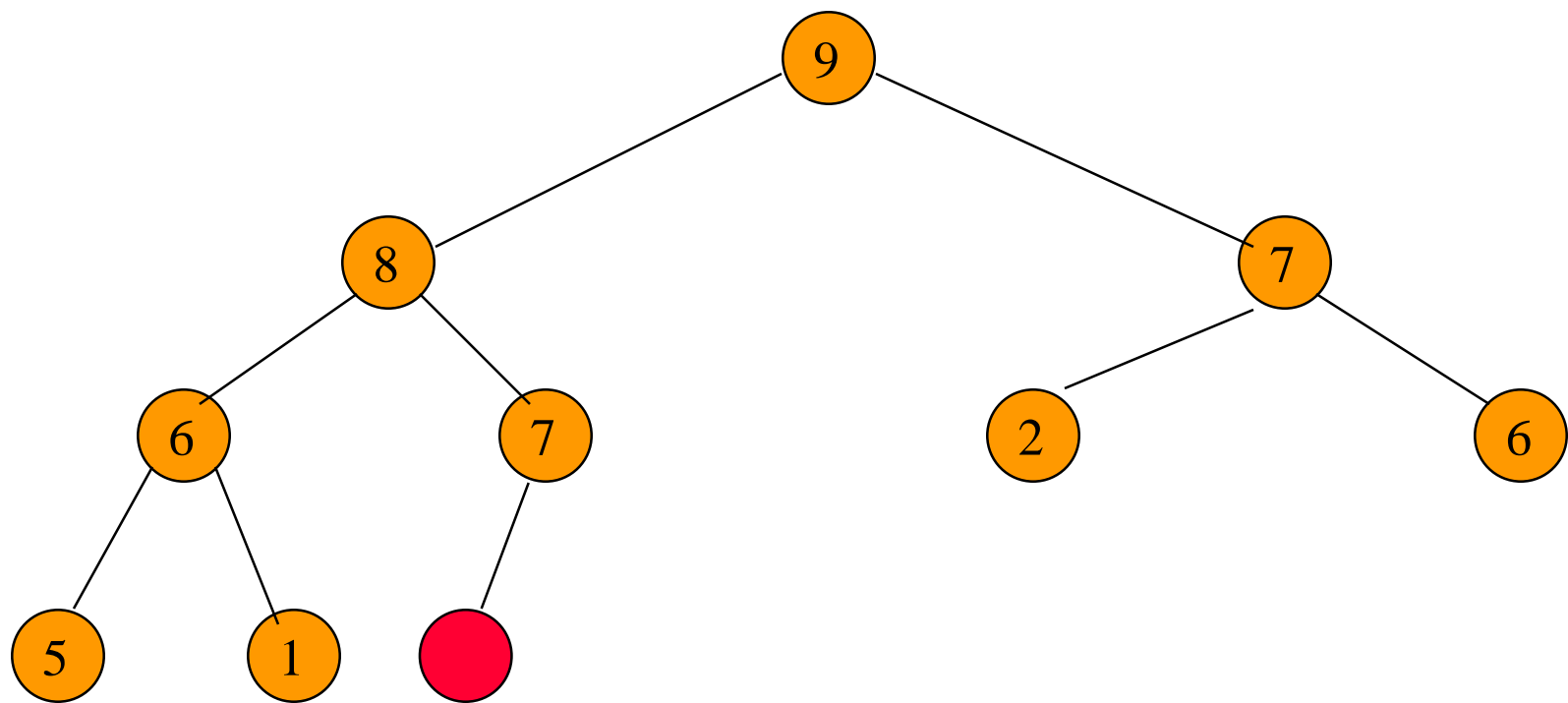


0 1 2 3 4 5 6 7 8 9 10

Овоолгоор дээш, доош явах

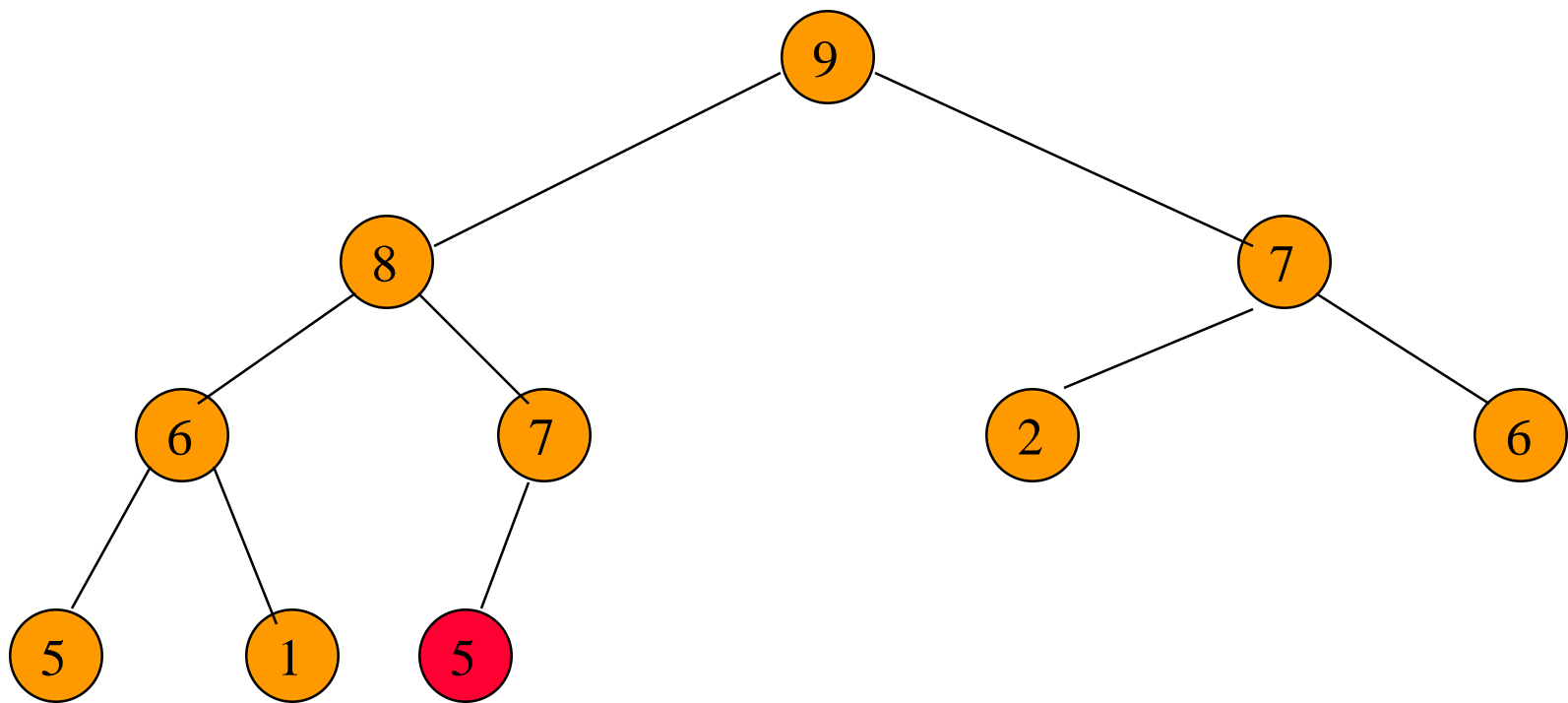


Мах овоолгод элемент хийх



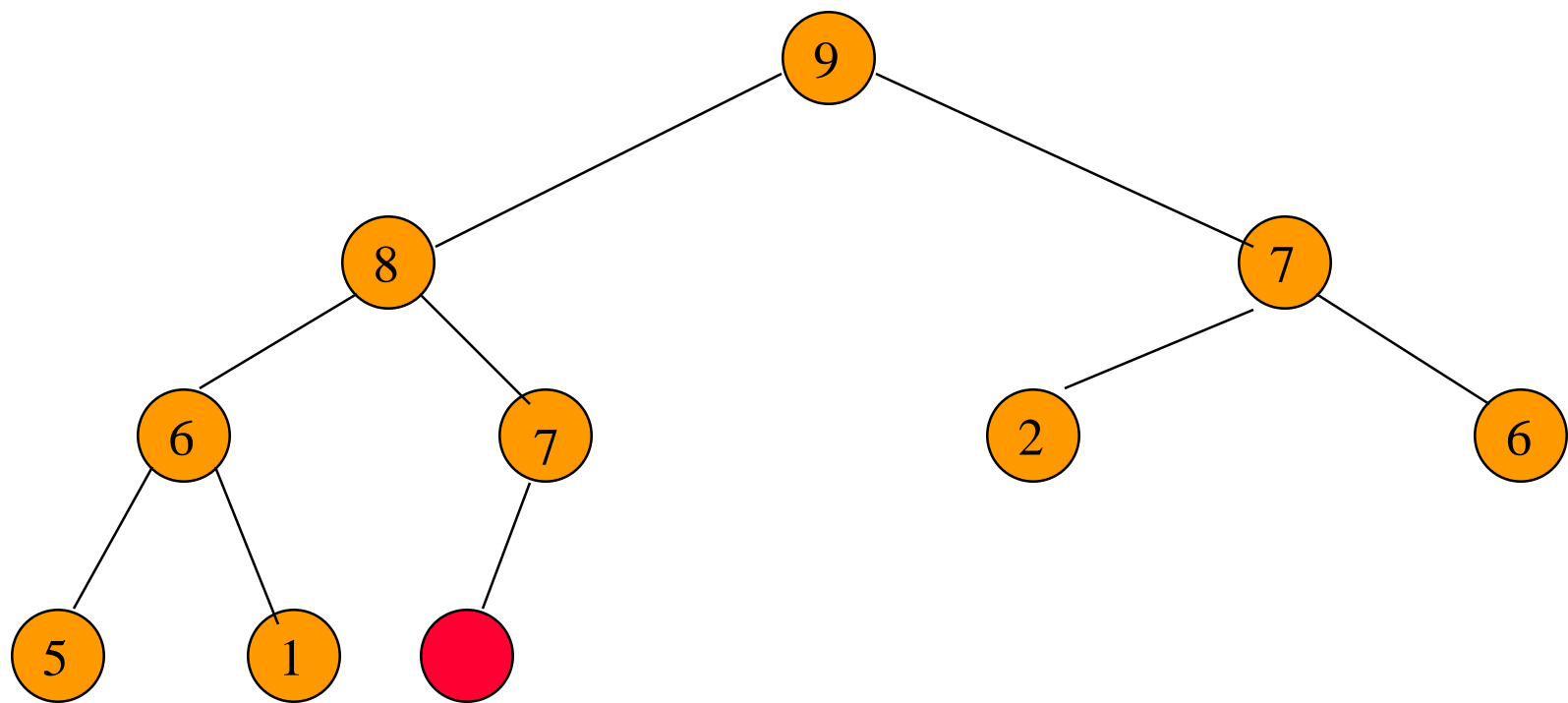
10 зангилаатай төгс хоёртын мод.

Мах овоолгод элемент хийх



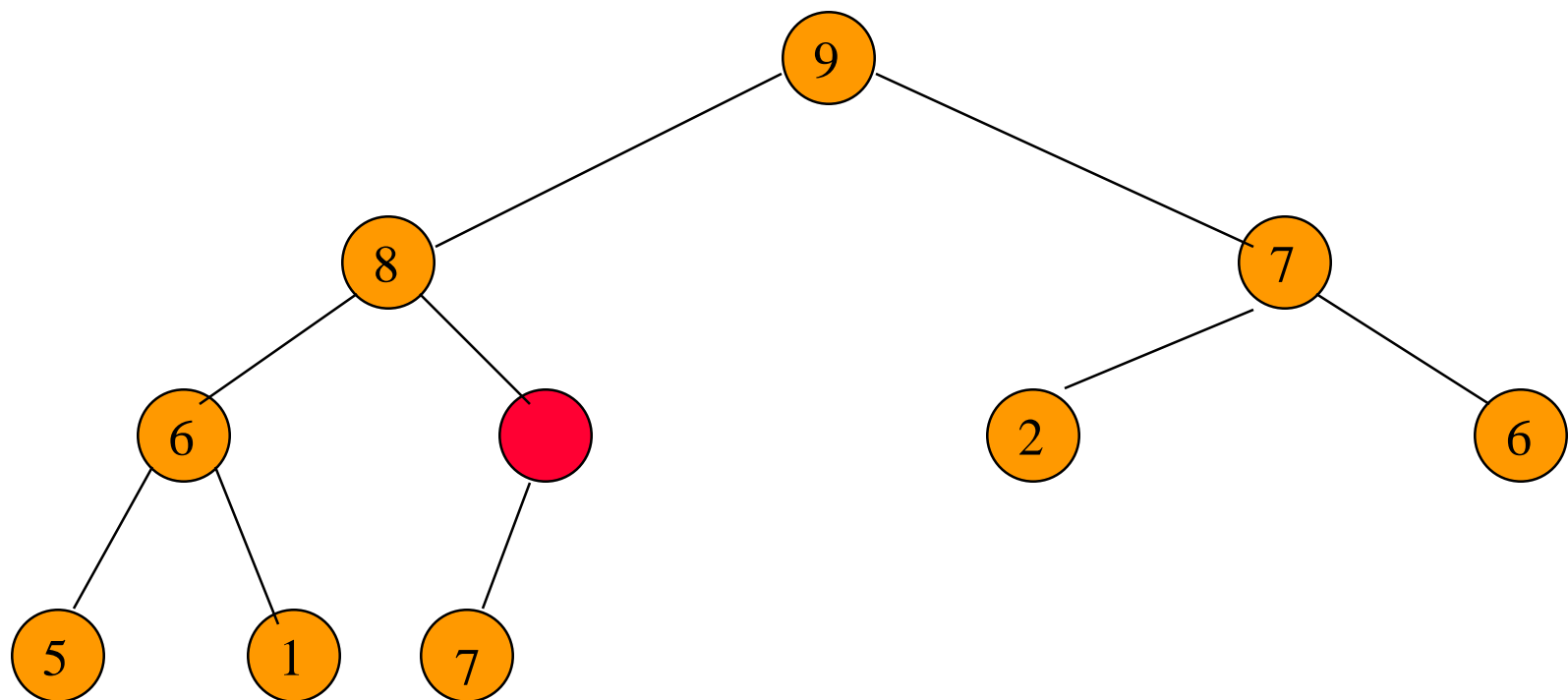
Шинэ элемент 5.

Мах овоолгод элемент хийх



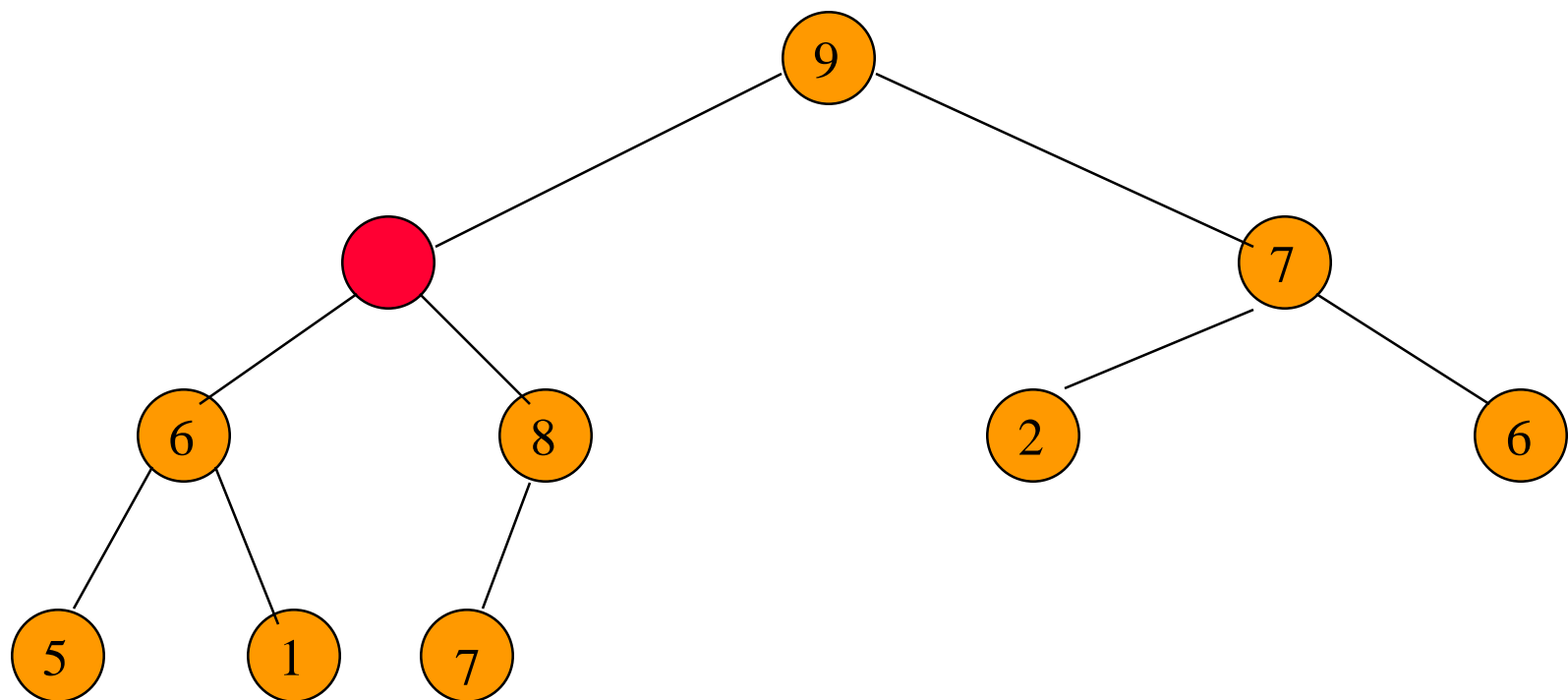
Шинэ элемент 20.

Мах овоолгод элемент хийх



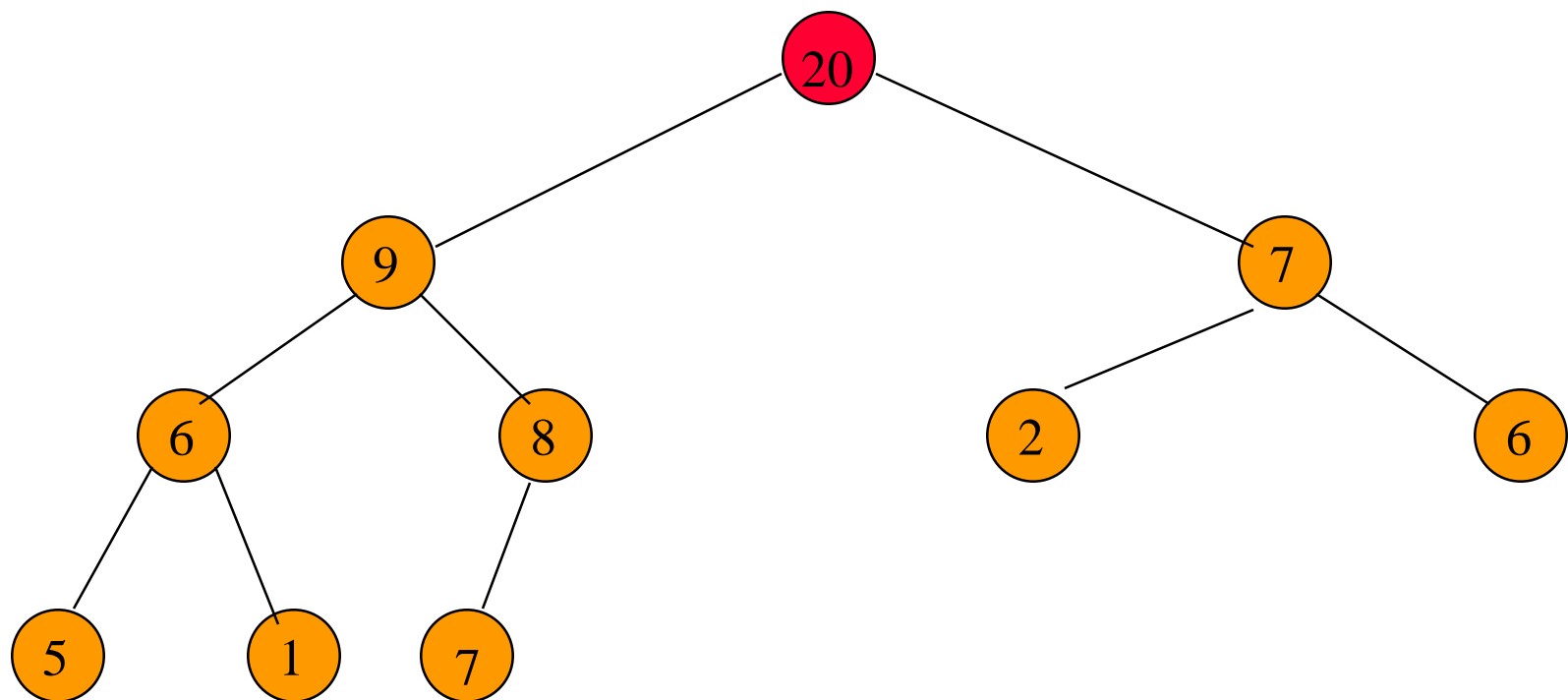
Шинэ элемент 20.

Мах овоолгод элемент хийх



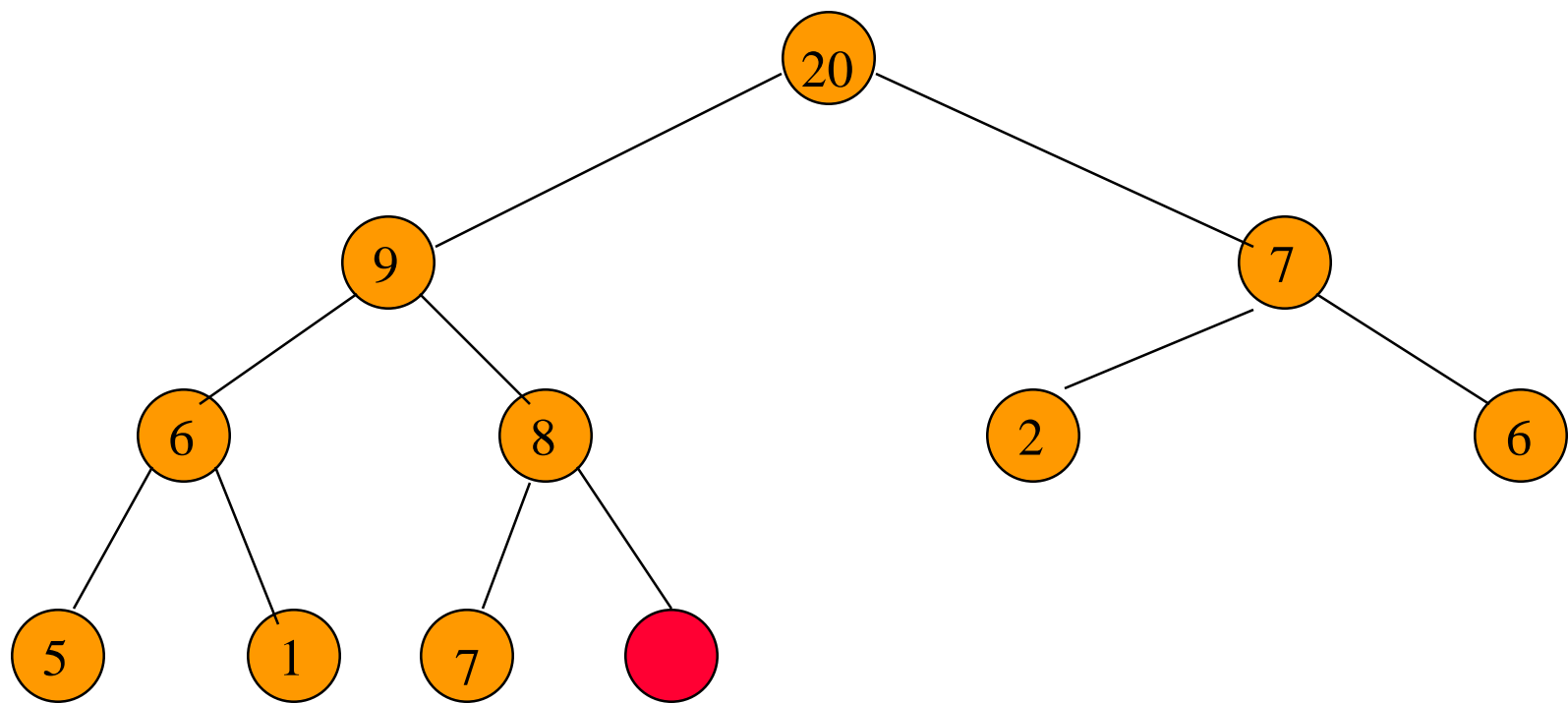
Шинэ элемент 20.

Мах овоолгод элемент хийх



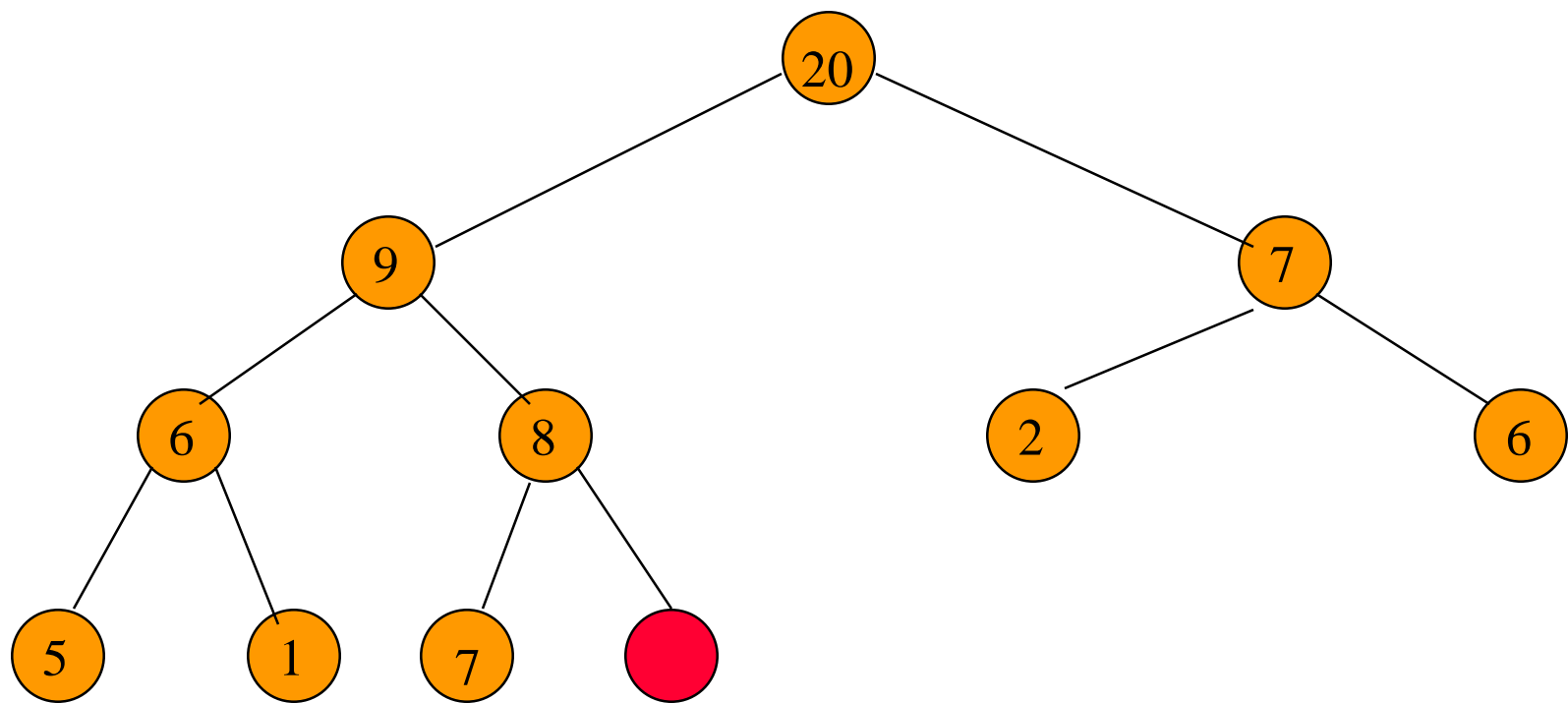
Шинэ элемент 20.

Мах овоолгод элемент хийх



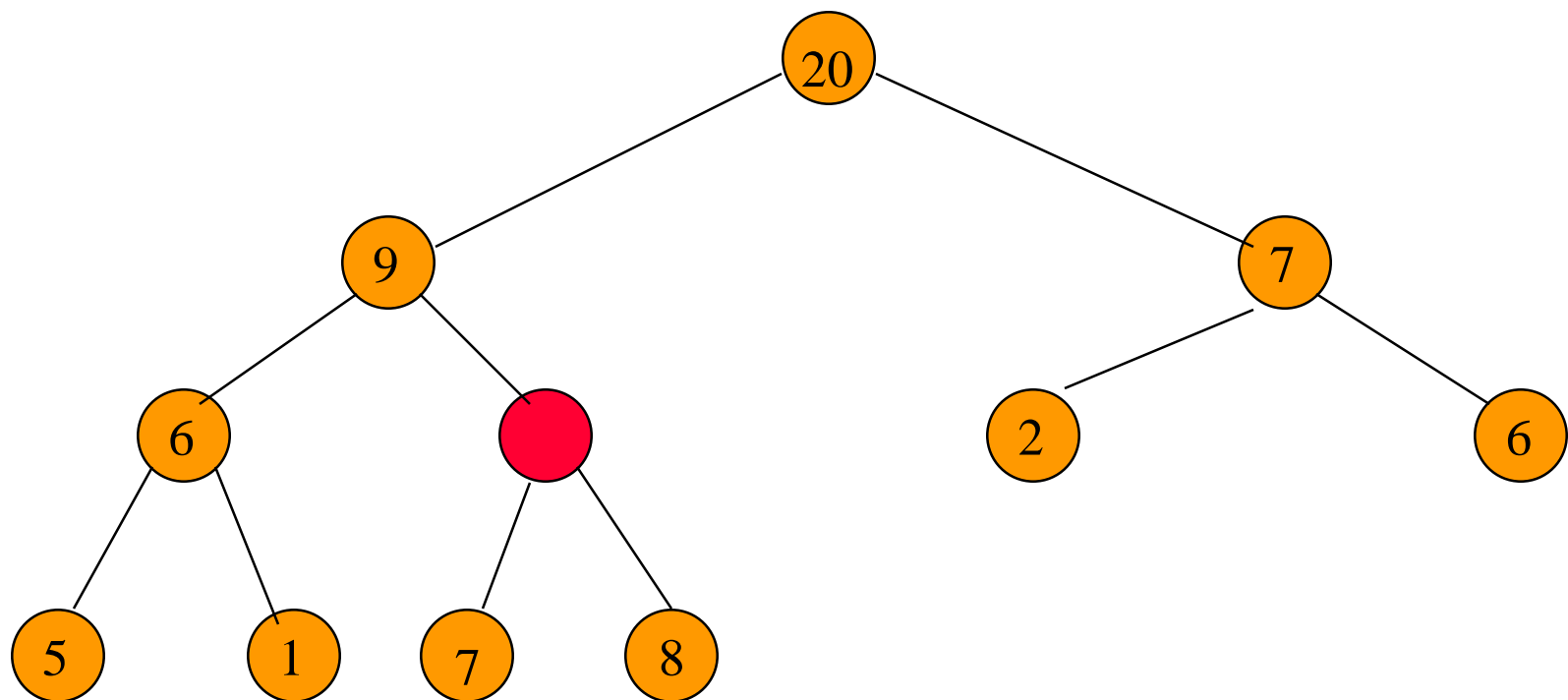
11 зангилаатай төгс хоёртын мод.

Мах овоолгод элемент хийх



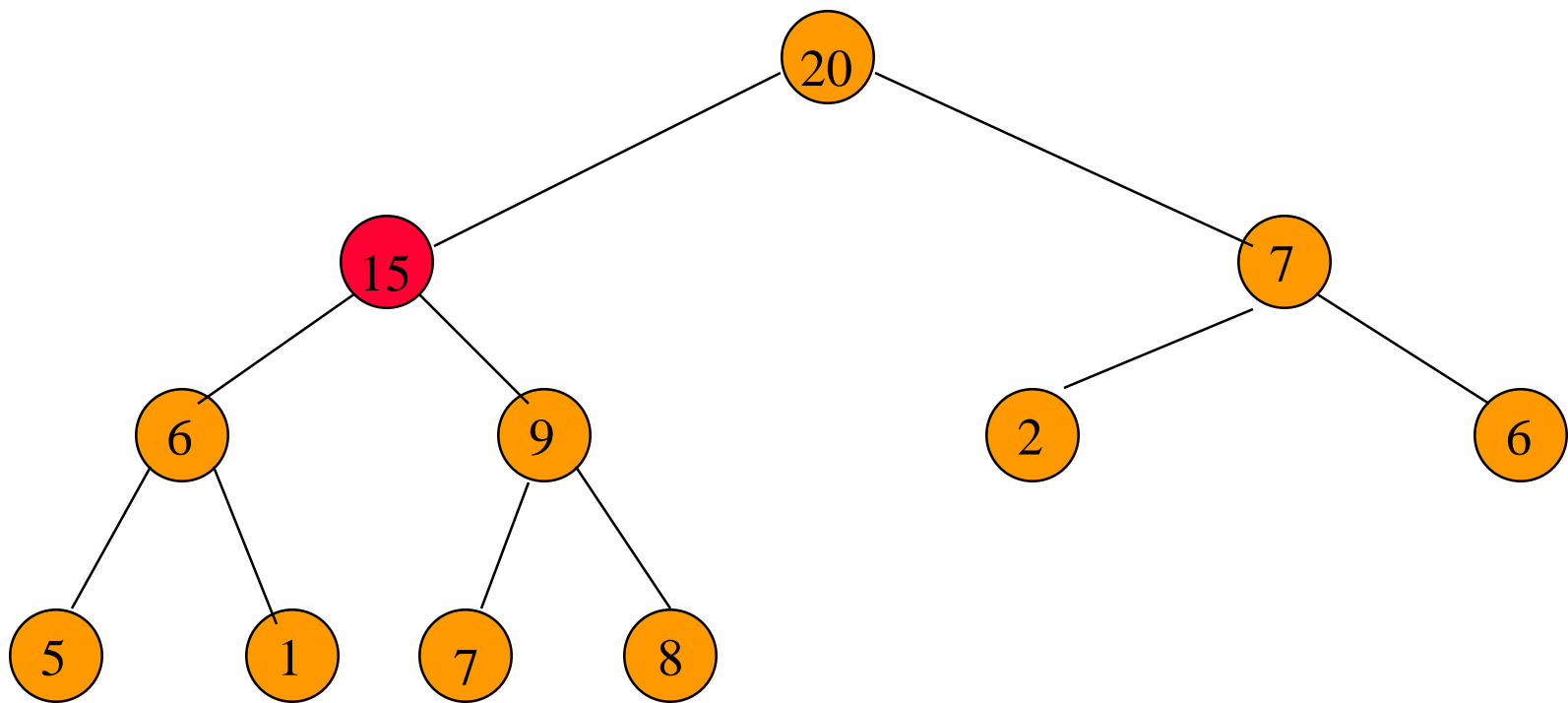
Шинэ элемент 15.

Мах овоолгод элемент хийх



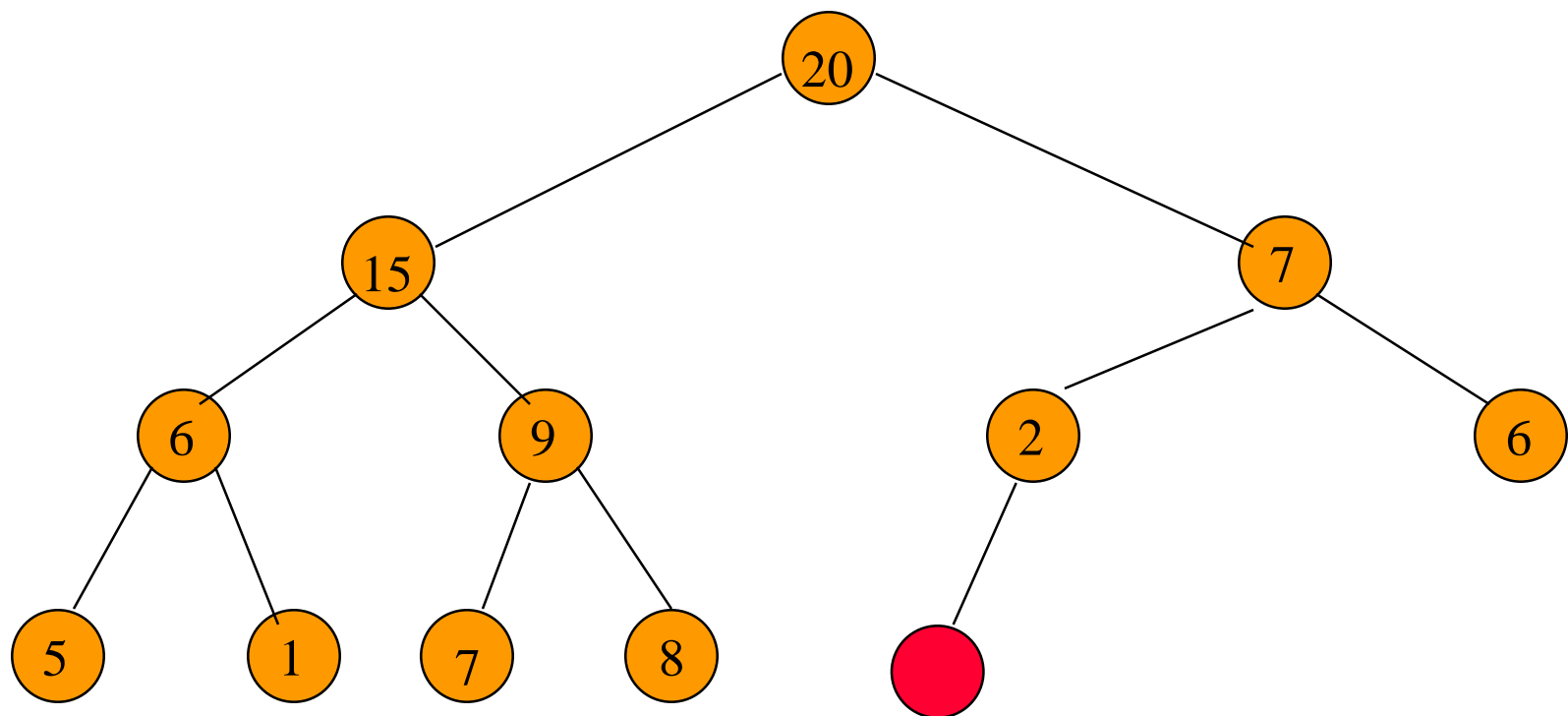
Шинэ элемент 15.

Мах овоолгод элемент хийх



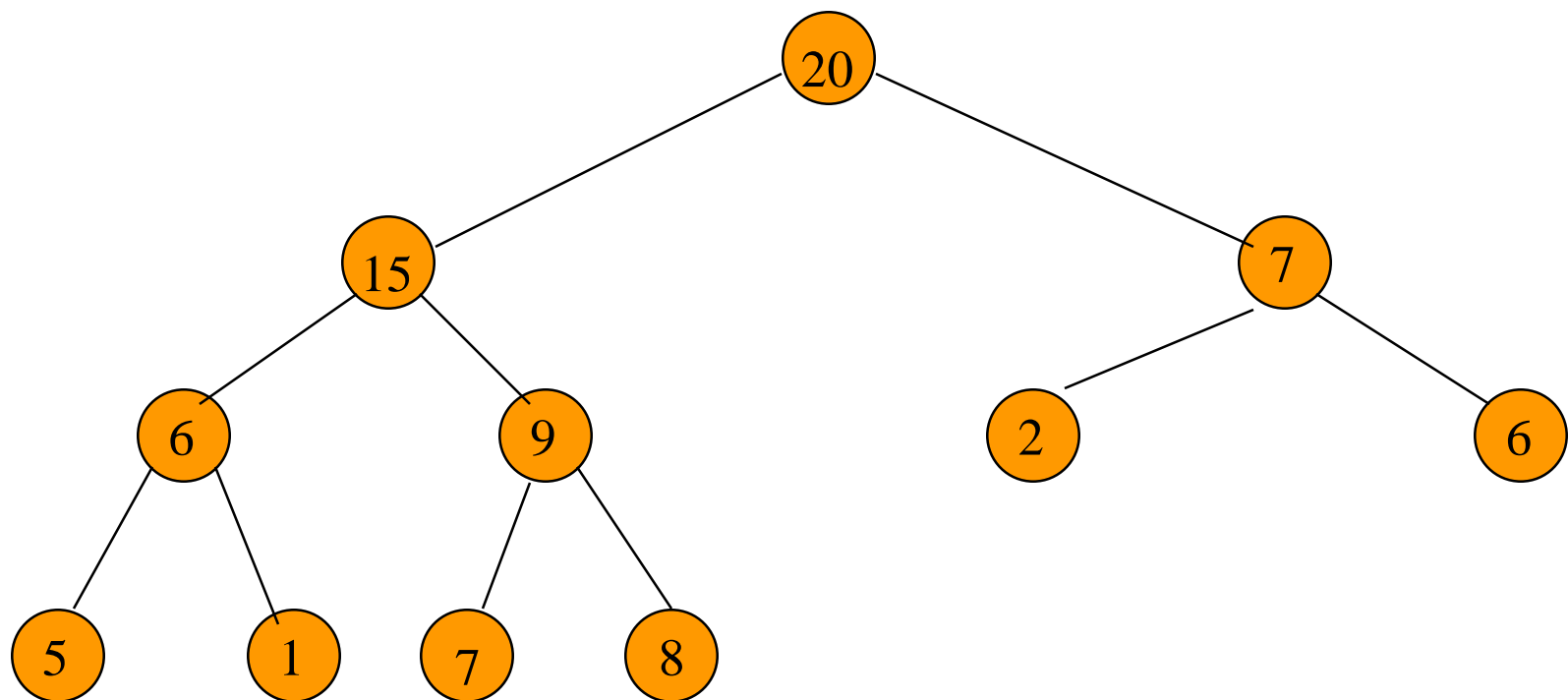
Шинэ элемент 15.

Put үйлдлийн хугацаа



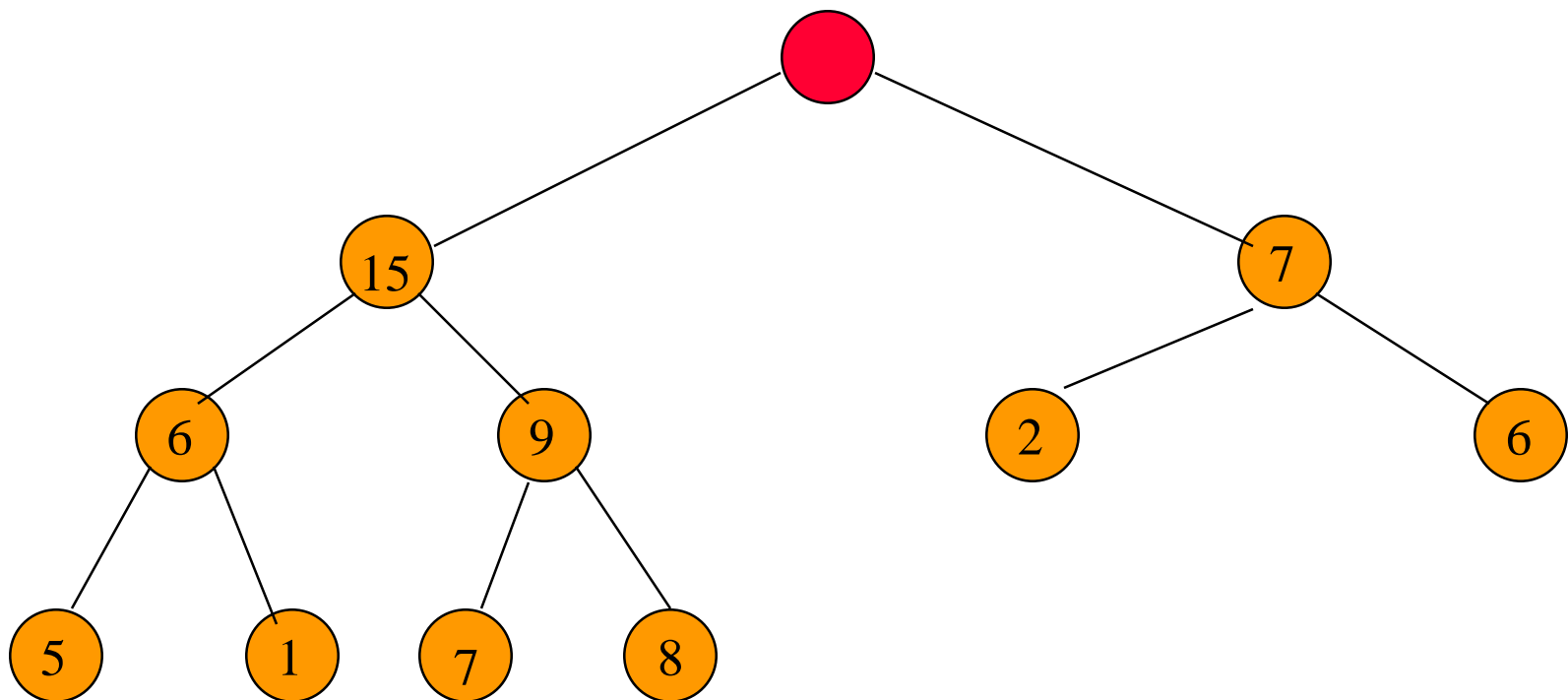
Хугацаа $O(\log n)$, үүнд n ОВООЛГЫН
ХЭМЖЭЭ.

Мах элементийн устгах



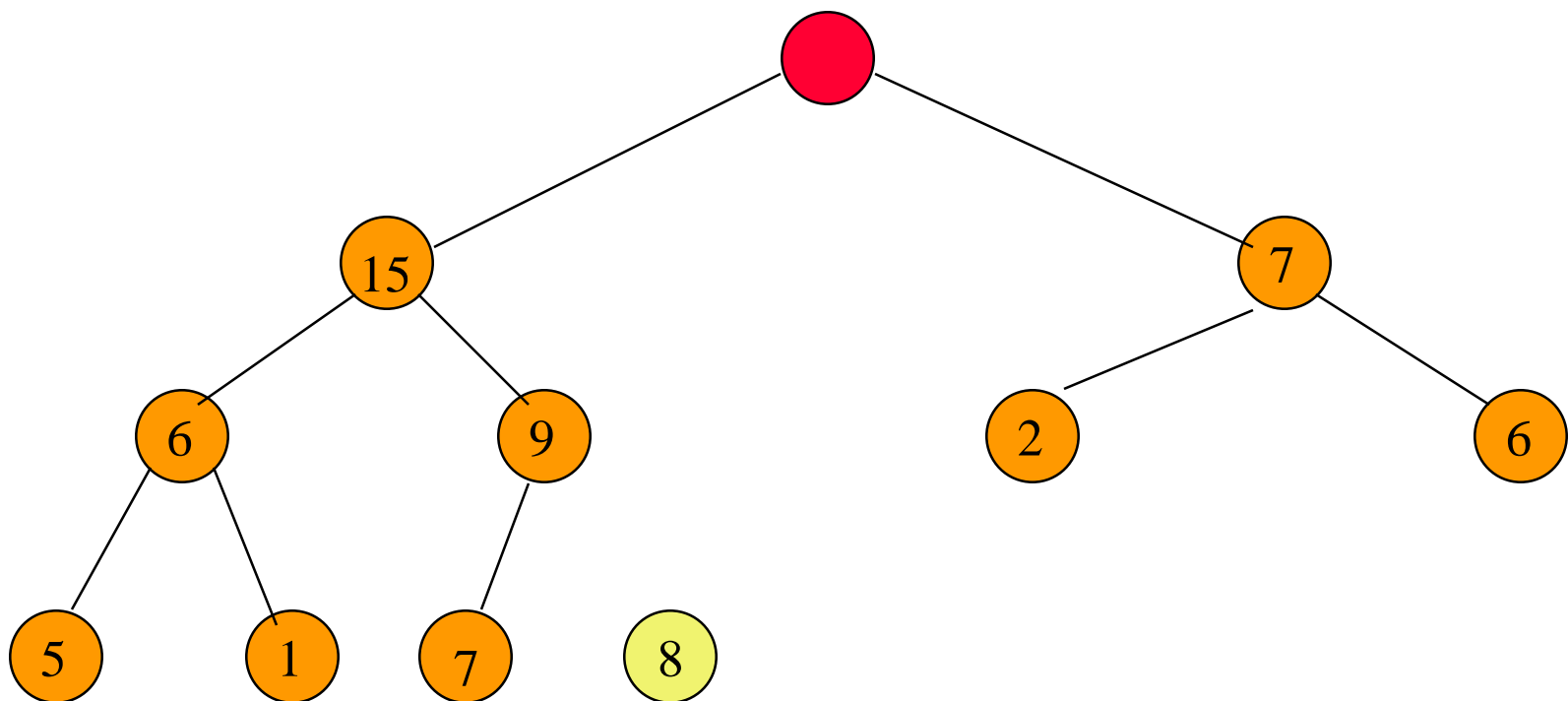
Мах элемент нь үндэс.

Мах элементийн устгах



тах элементийг устгасны дараа.

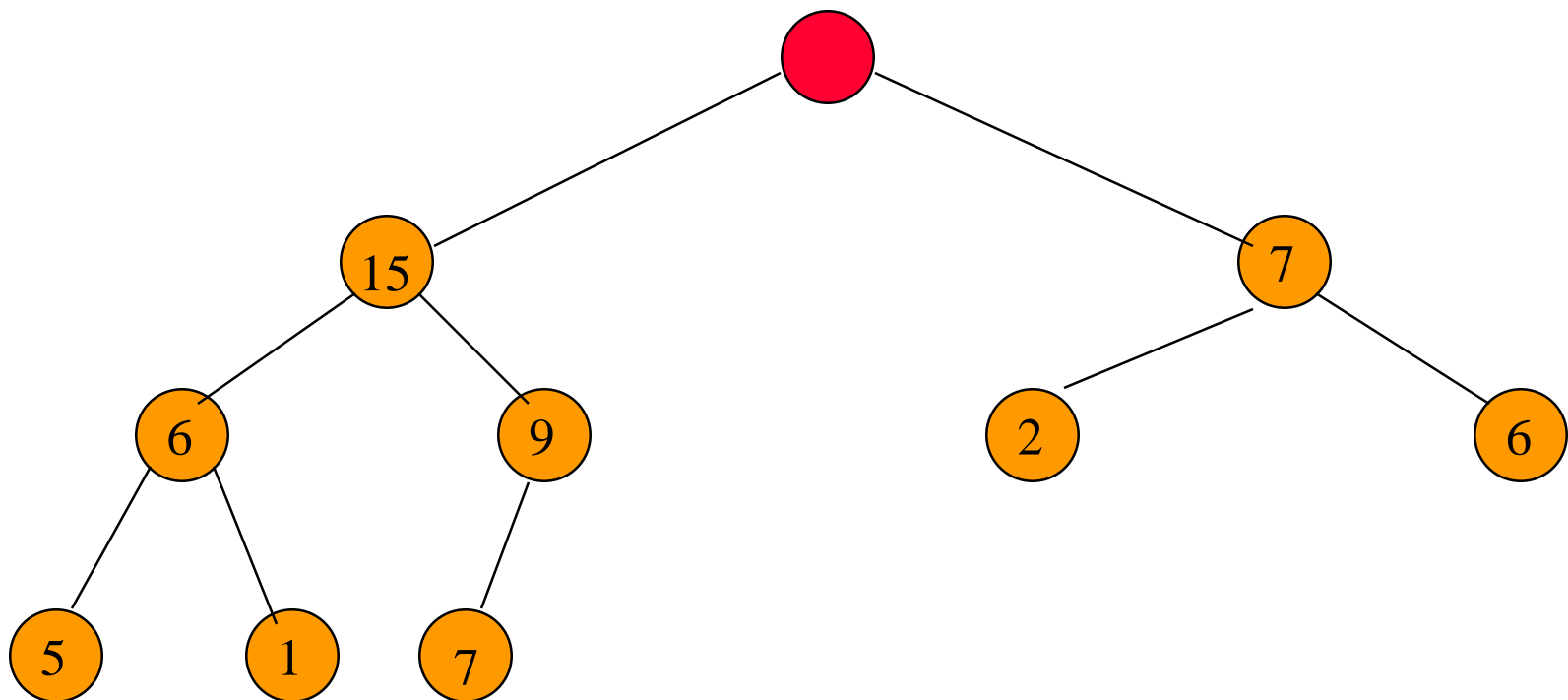
Мах элементийн устгах



10 зангилаатай овоолго.

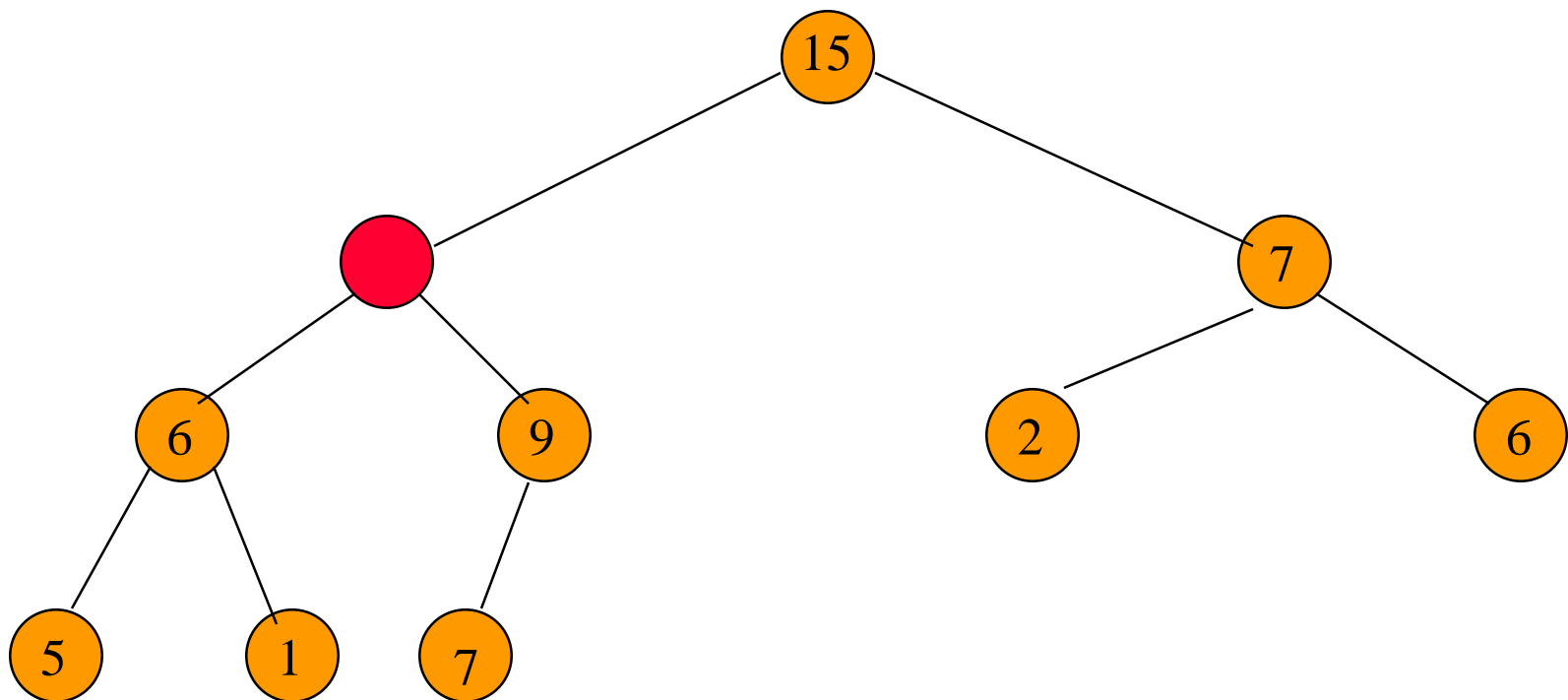
Овоолгод 8 –г дахин хийх.

Мах элементийн устгах



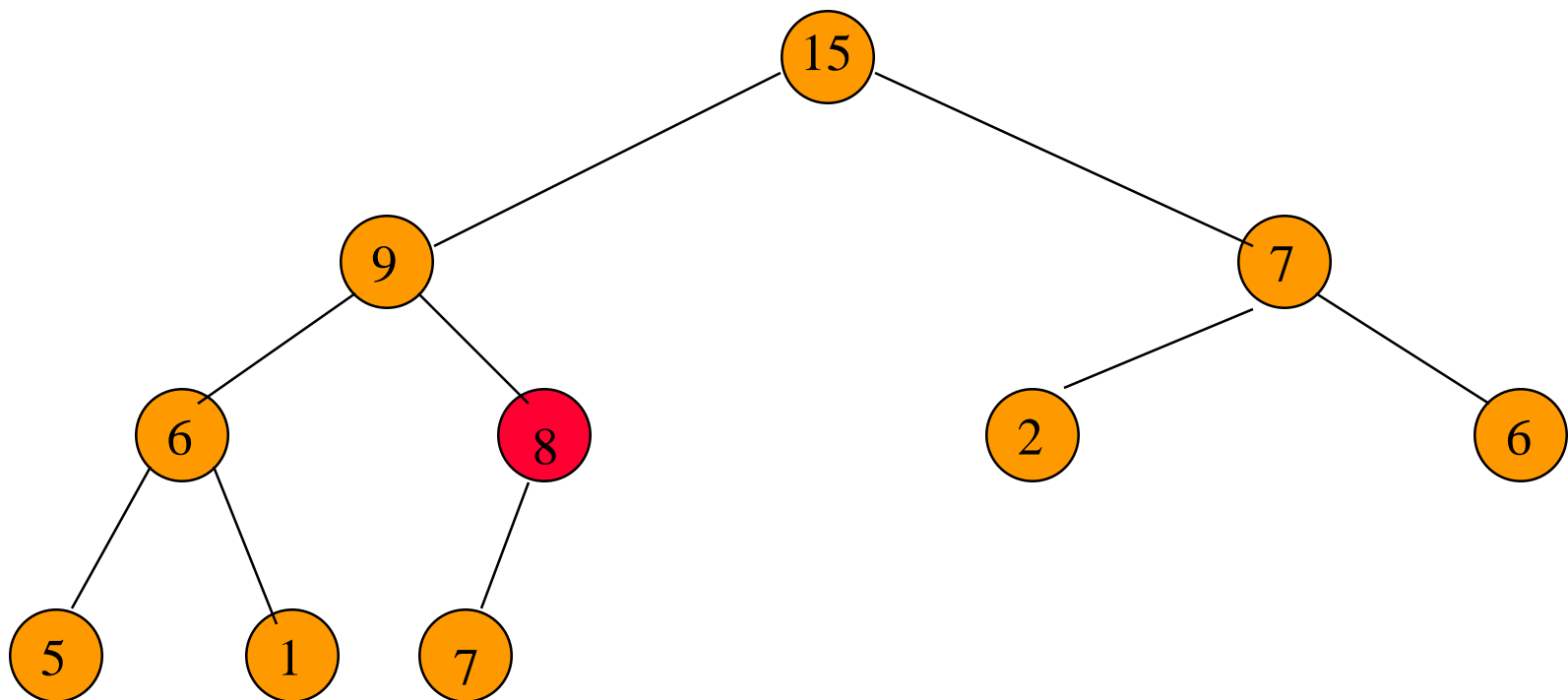
Овоолгод 8 –г дахин хийх.

Мах элементийн устгах



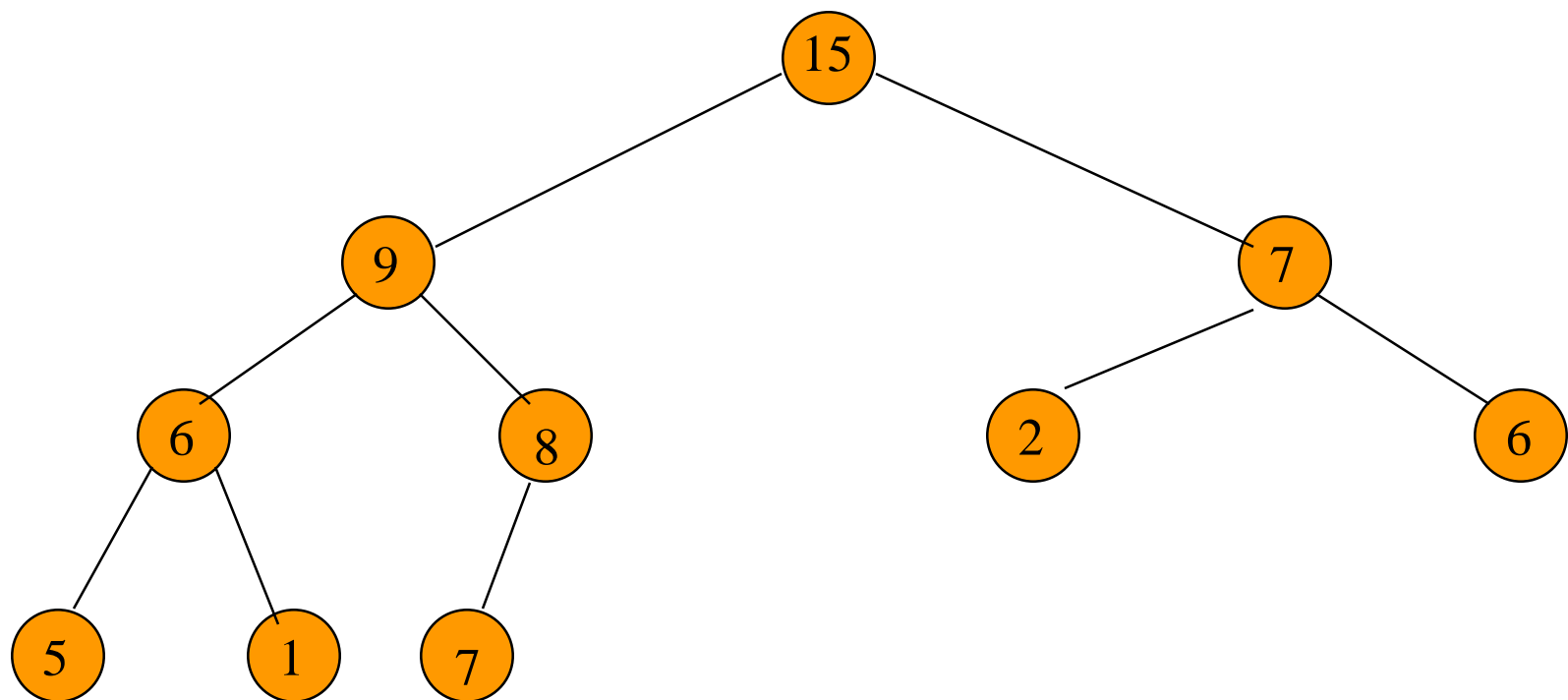
Овоолгод 8 –г дахин хийх.

Мах элементийн устгах



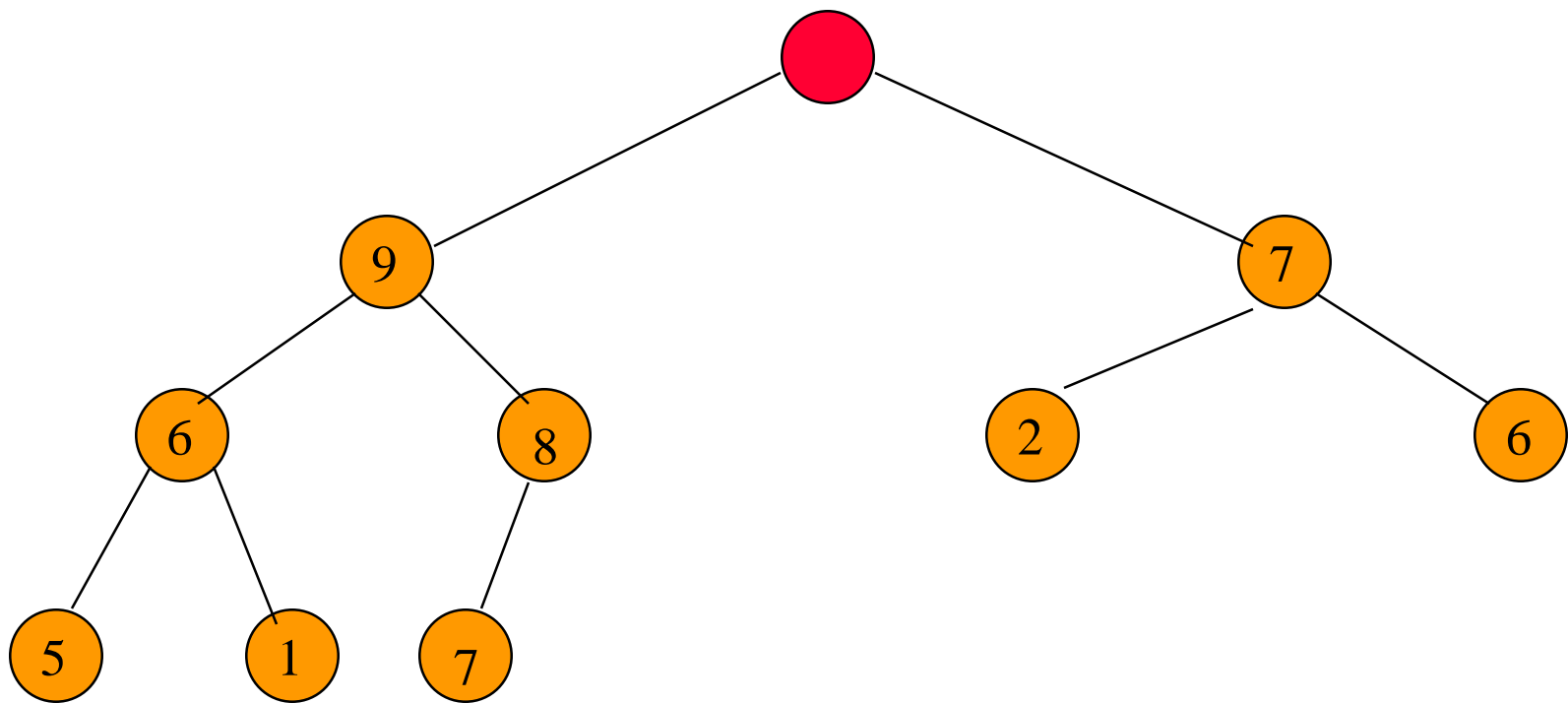
Овоолго 8 —г дахин хийх.

Мах элементийн устгах



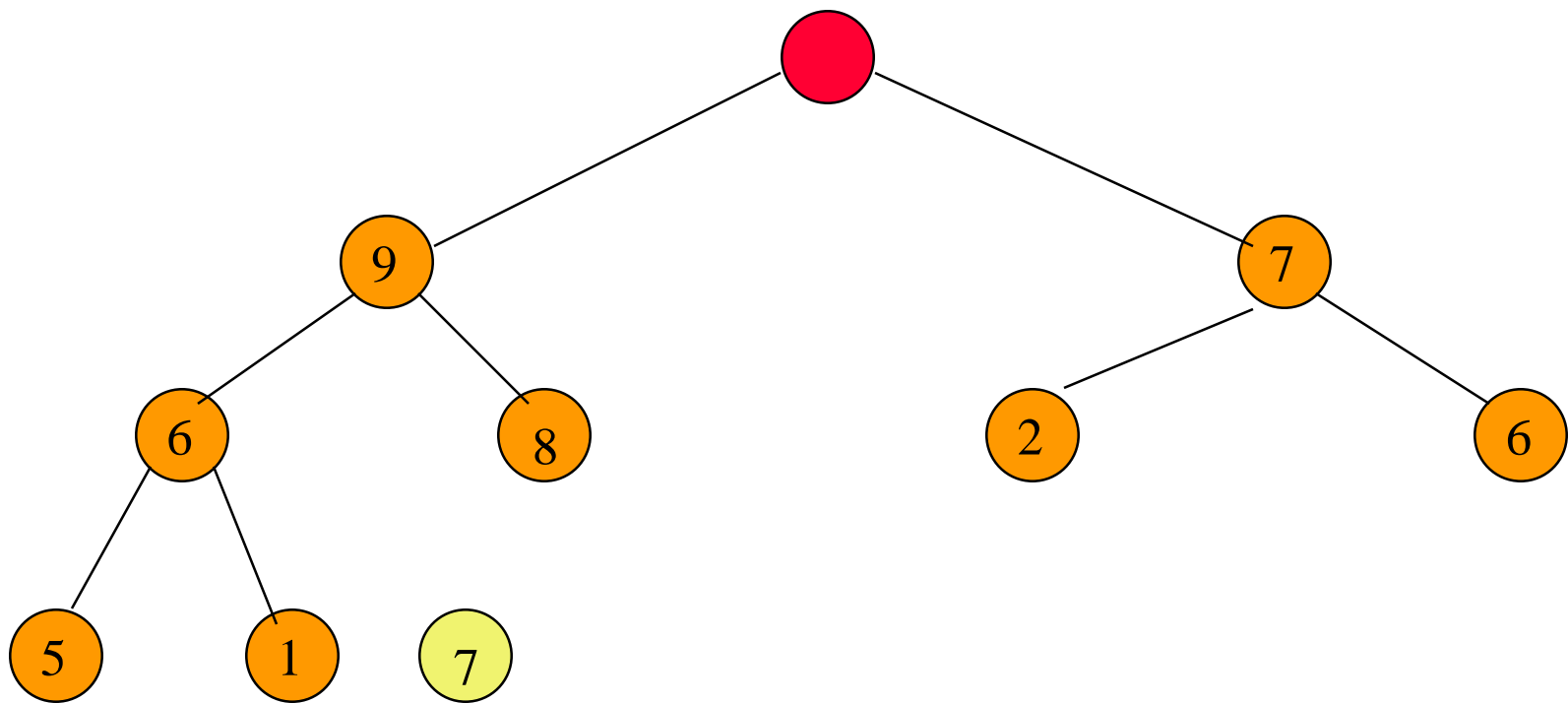
Мах элемент 15.

Мах элементийн устгах



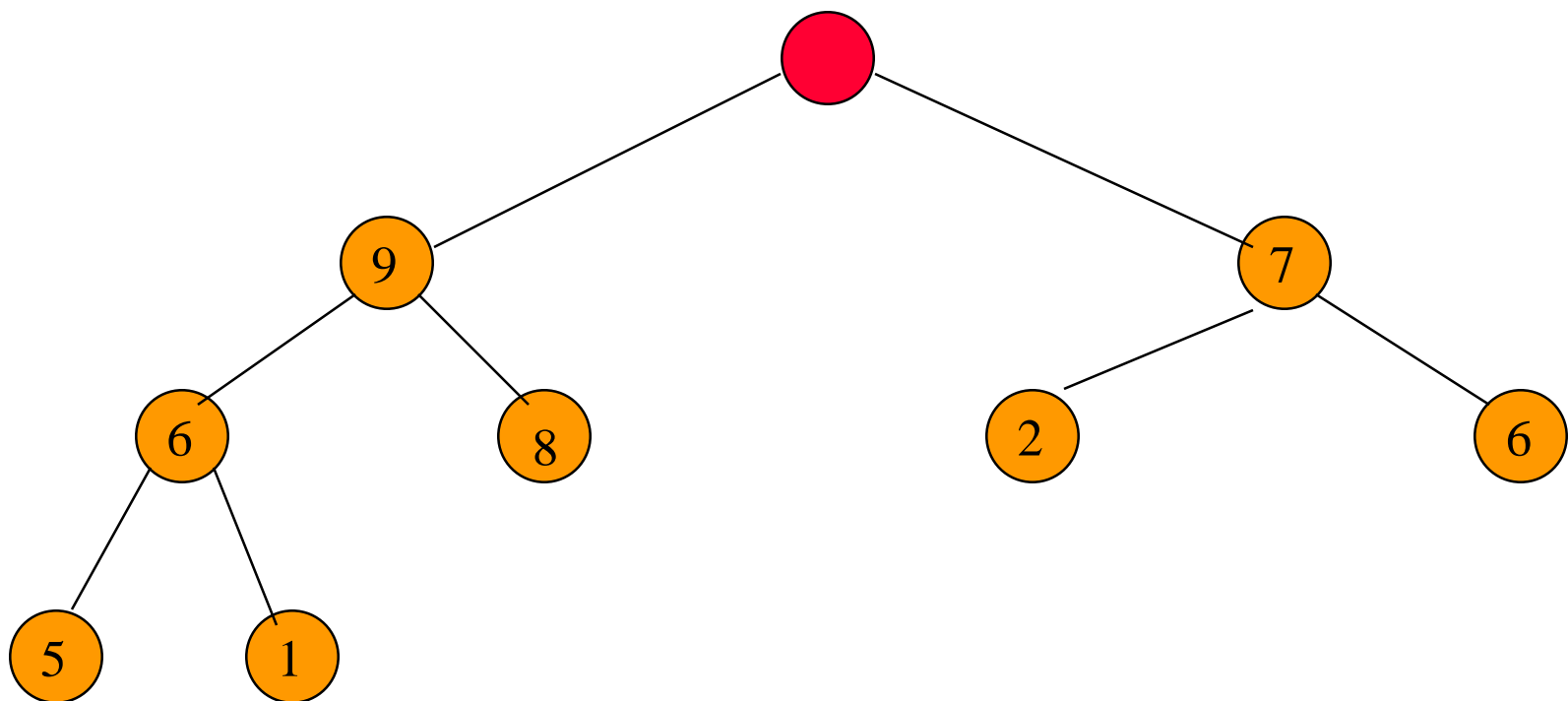
тах элементийг устгасны дараа.

Мах элементийн устгах



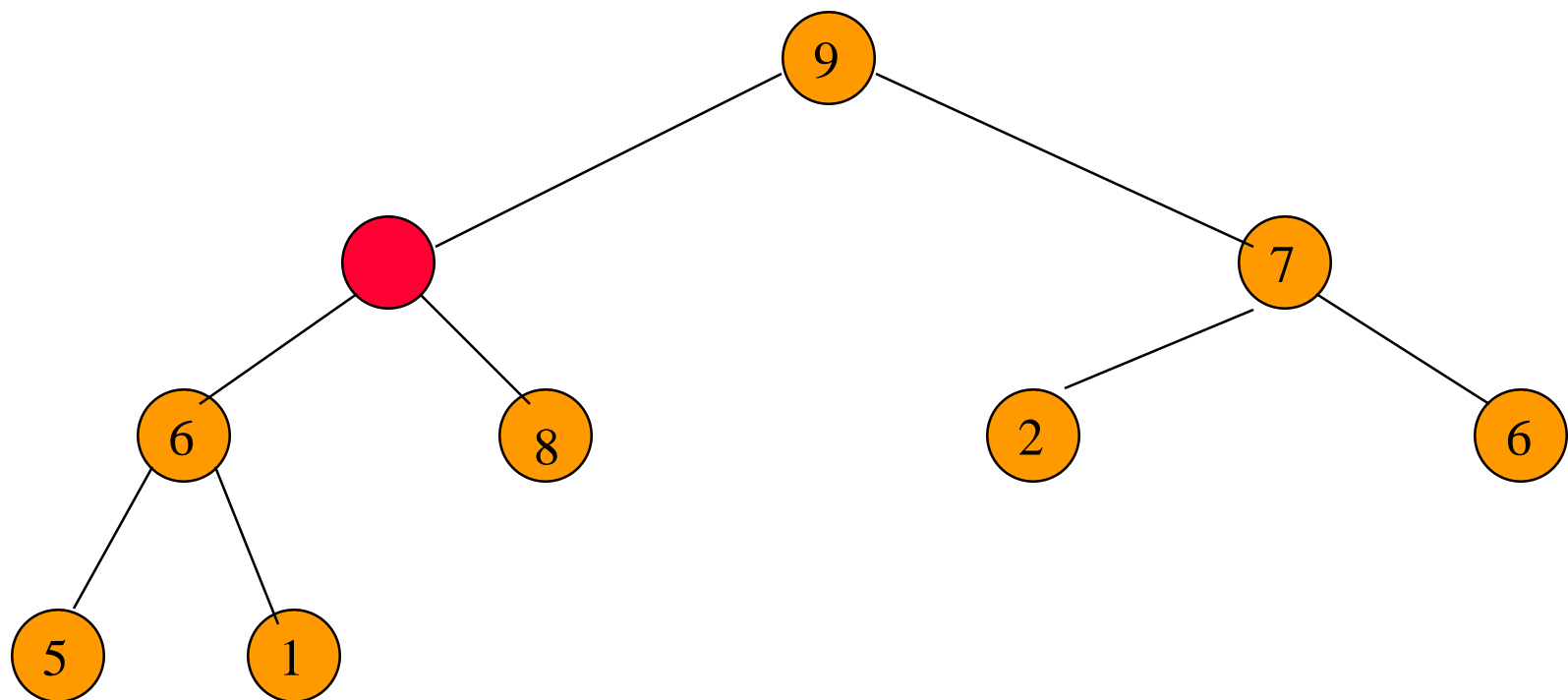
9 зангилаатай овоолго.

Мах элементийн устгах



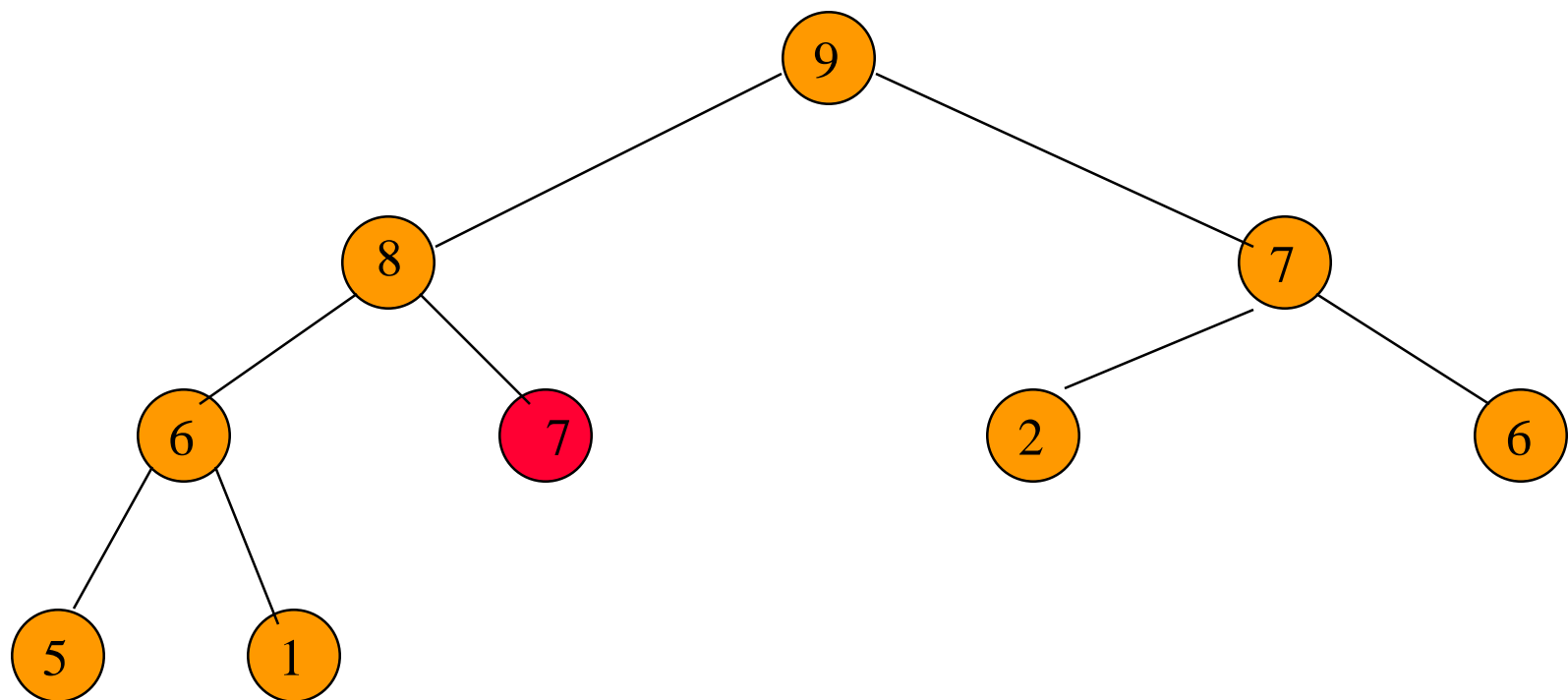
7 -г дахин хийх

Мах элементийн устгах



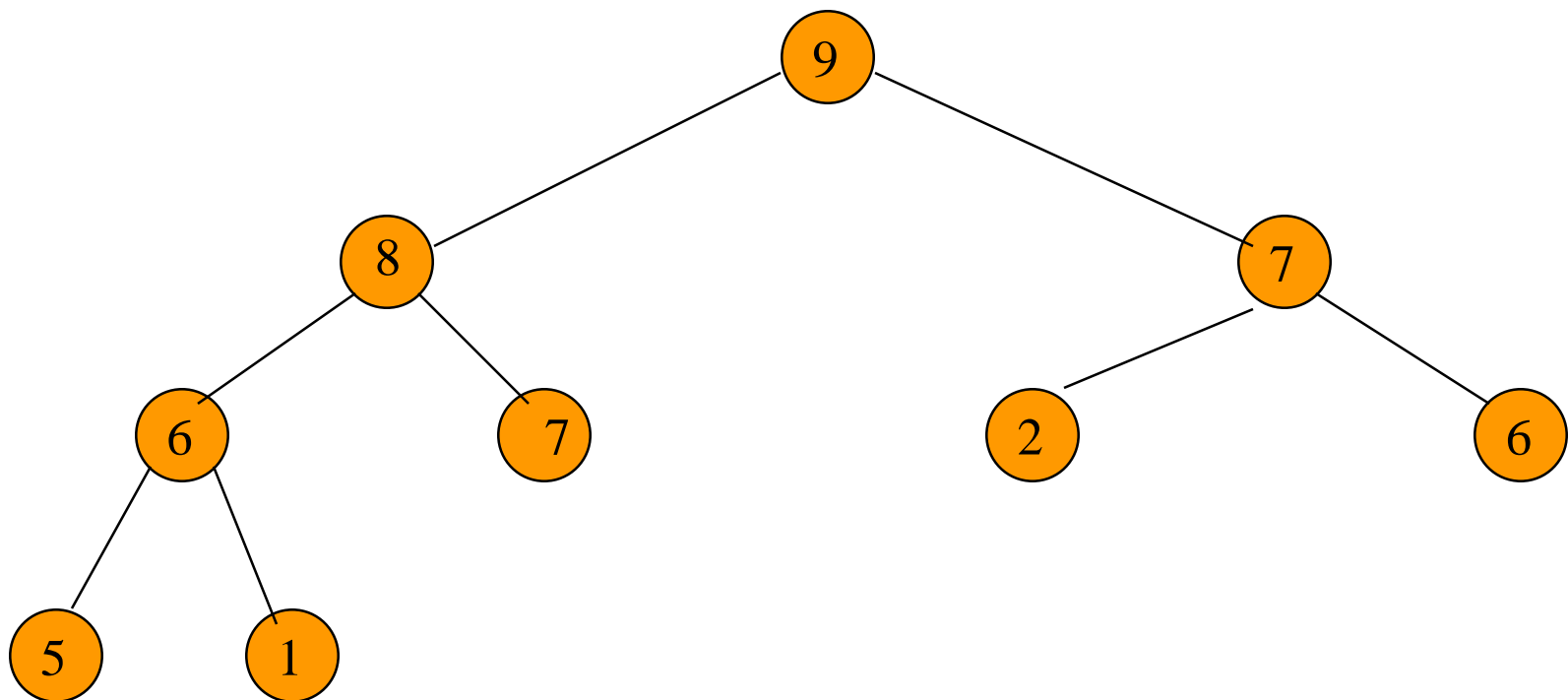
7 -г дахин хийх

Мах элементийн устгах



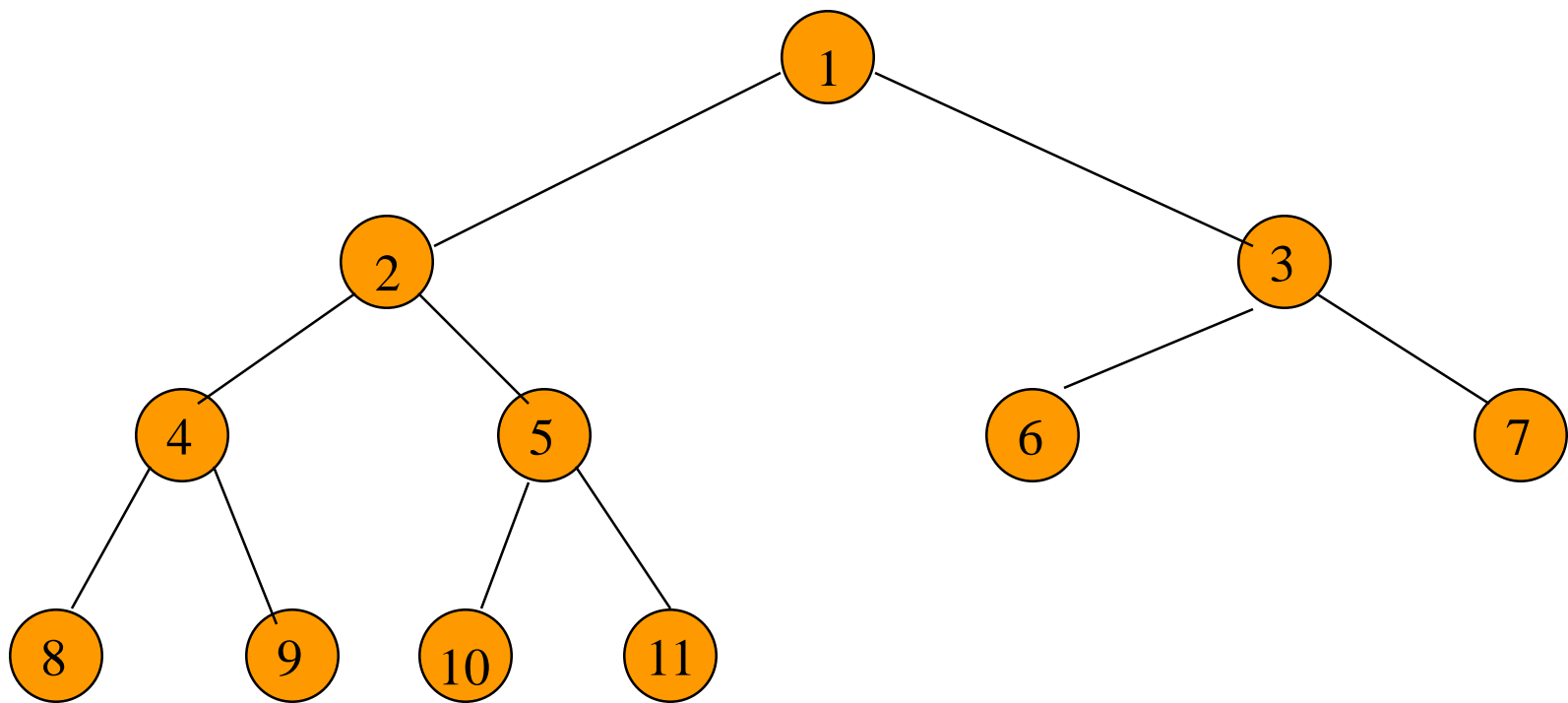
7 -г дахин хийх

Remove Max үйлдлийн хугацаа



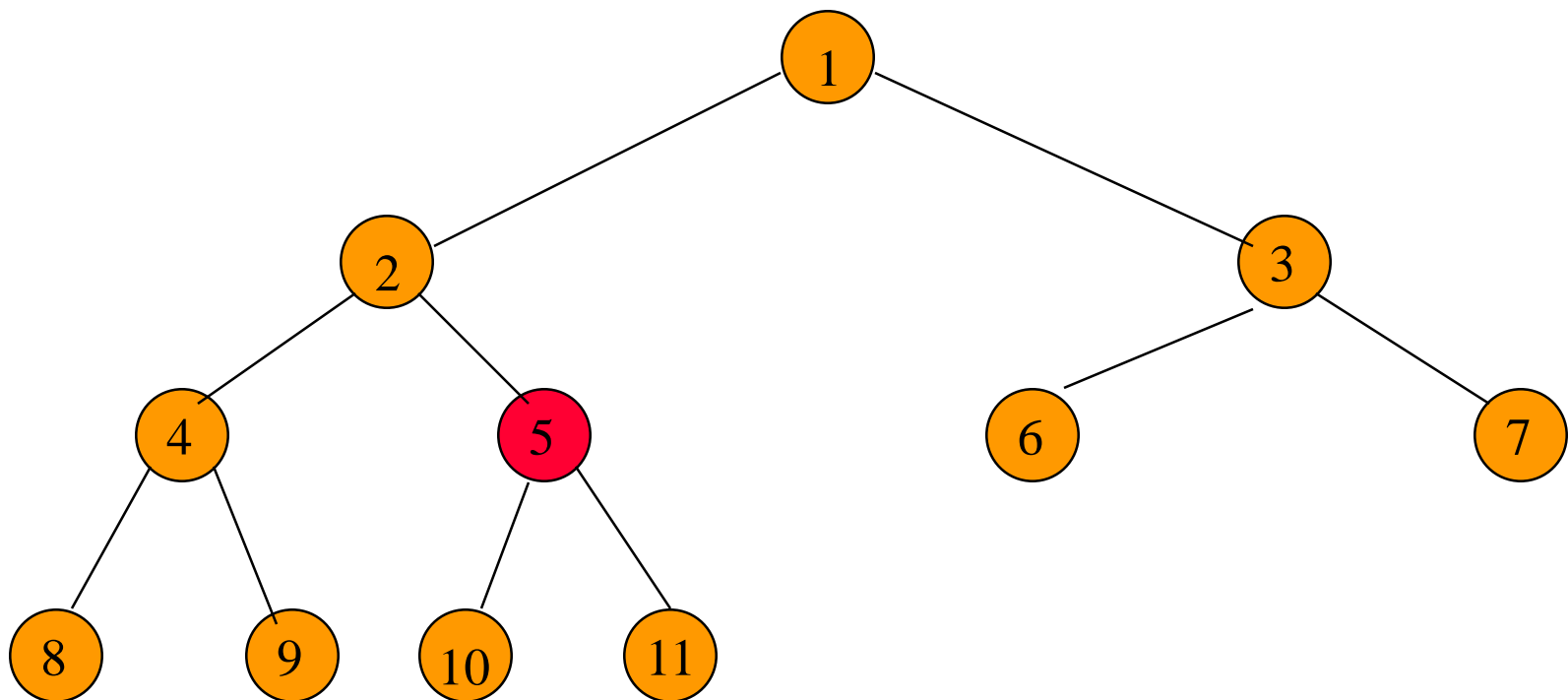
Хугацаа $O(\log n)$.

Мах овоолгыг идэхижүүлэх



Оролтын массив = [-, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

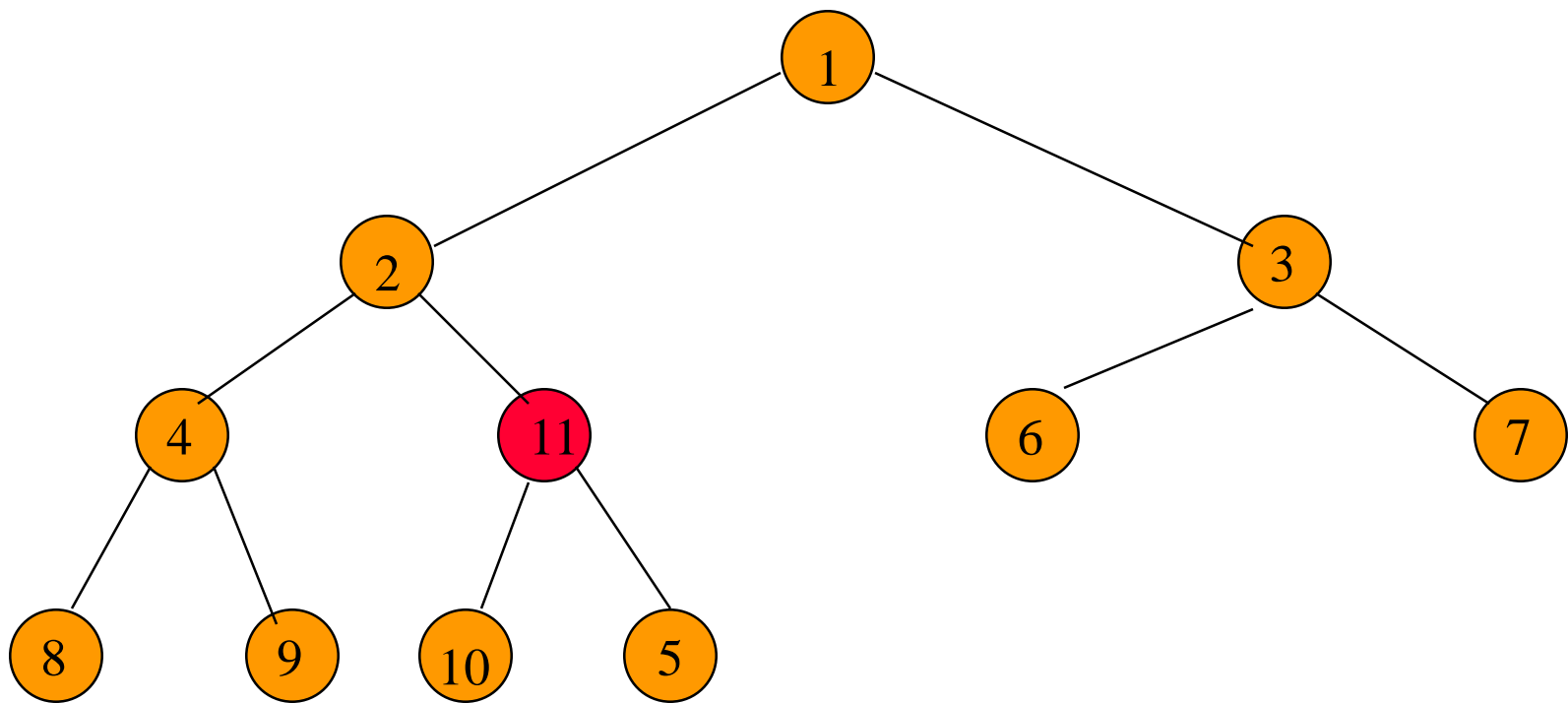
Мах овоолгыг идэхижүүлэх



Массивын хамгийн баруун байршилтай хүүхэдтэй
зангилаанаас эхэлнэ.

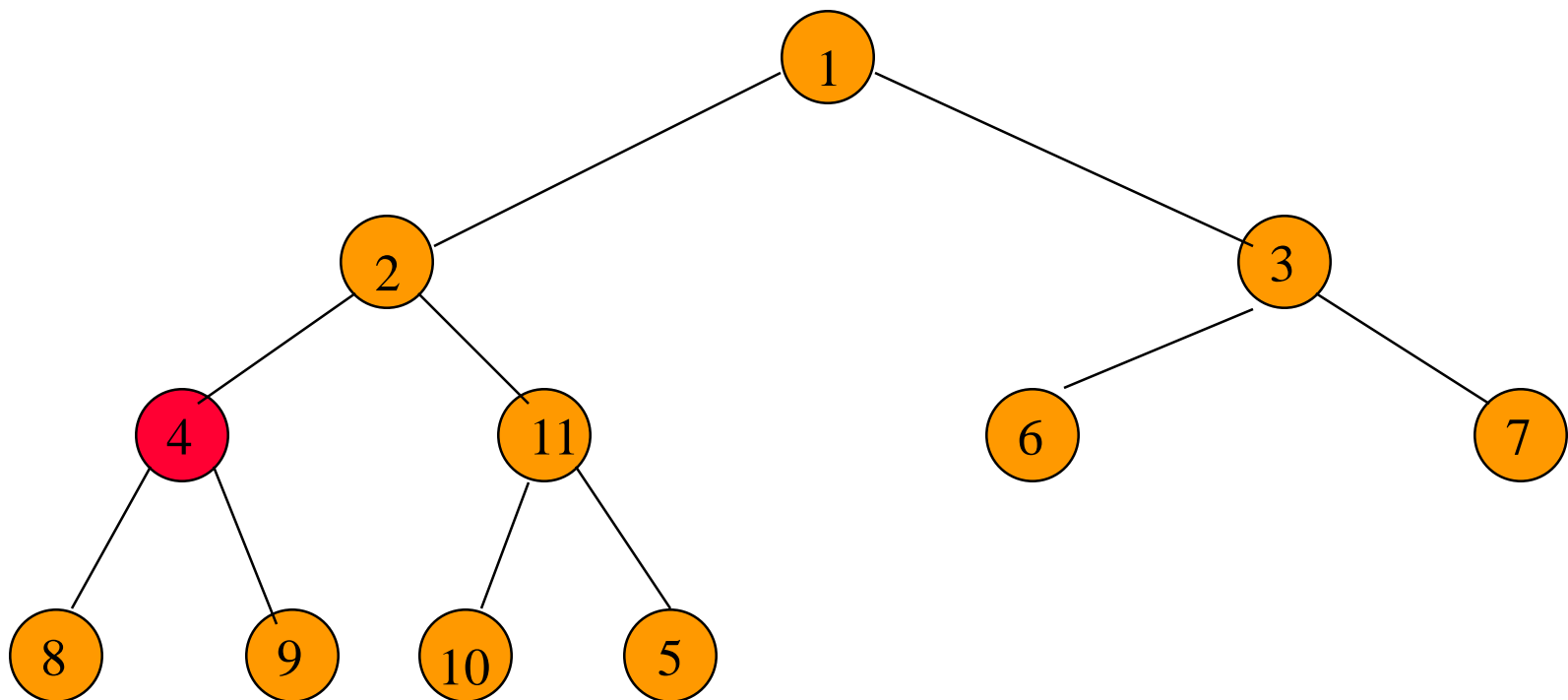
Индекс нь $n/2$.

Мах овоолгыг идэхижүүлэх

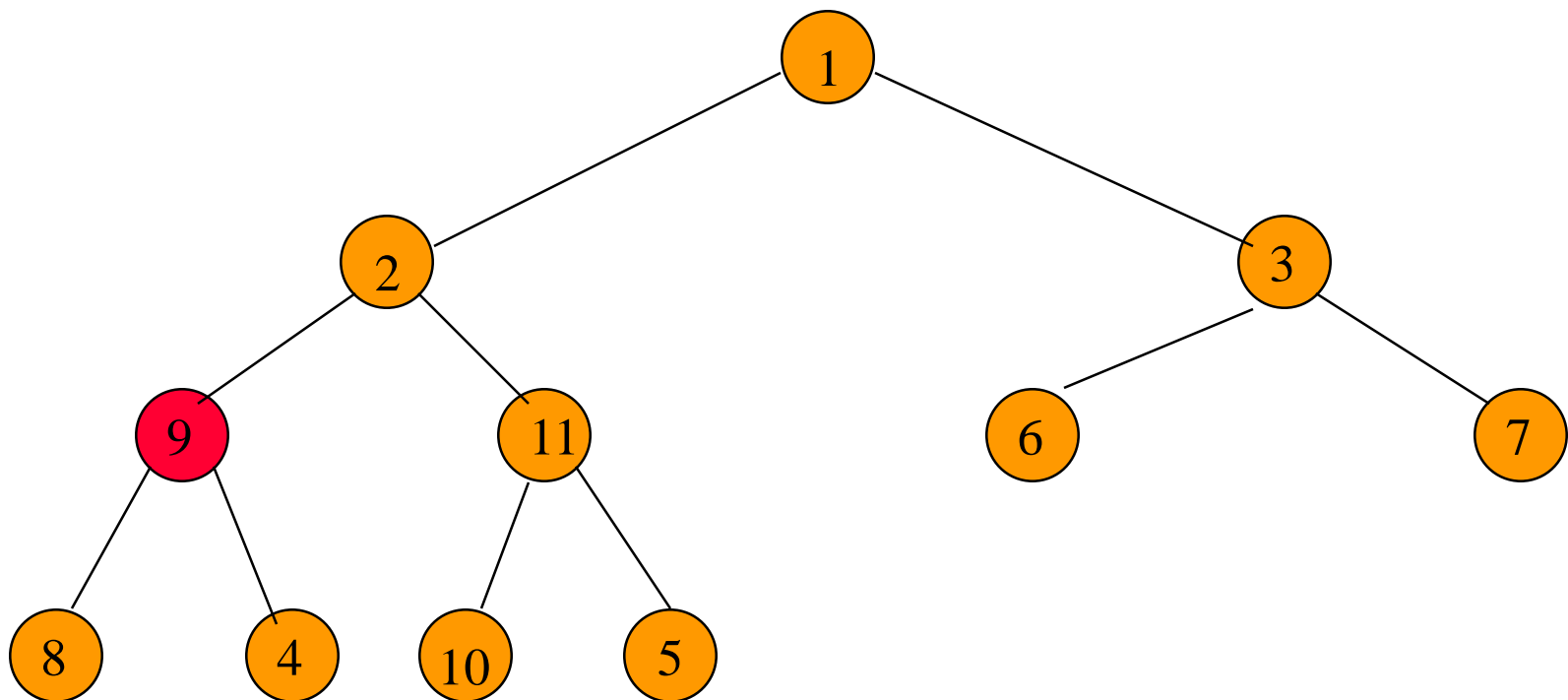


Массивын дараачийн доод байршил руу явна.

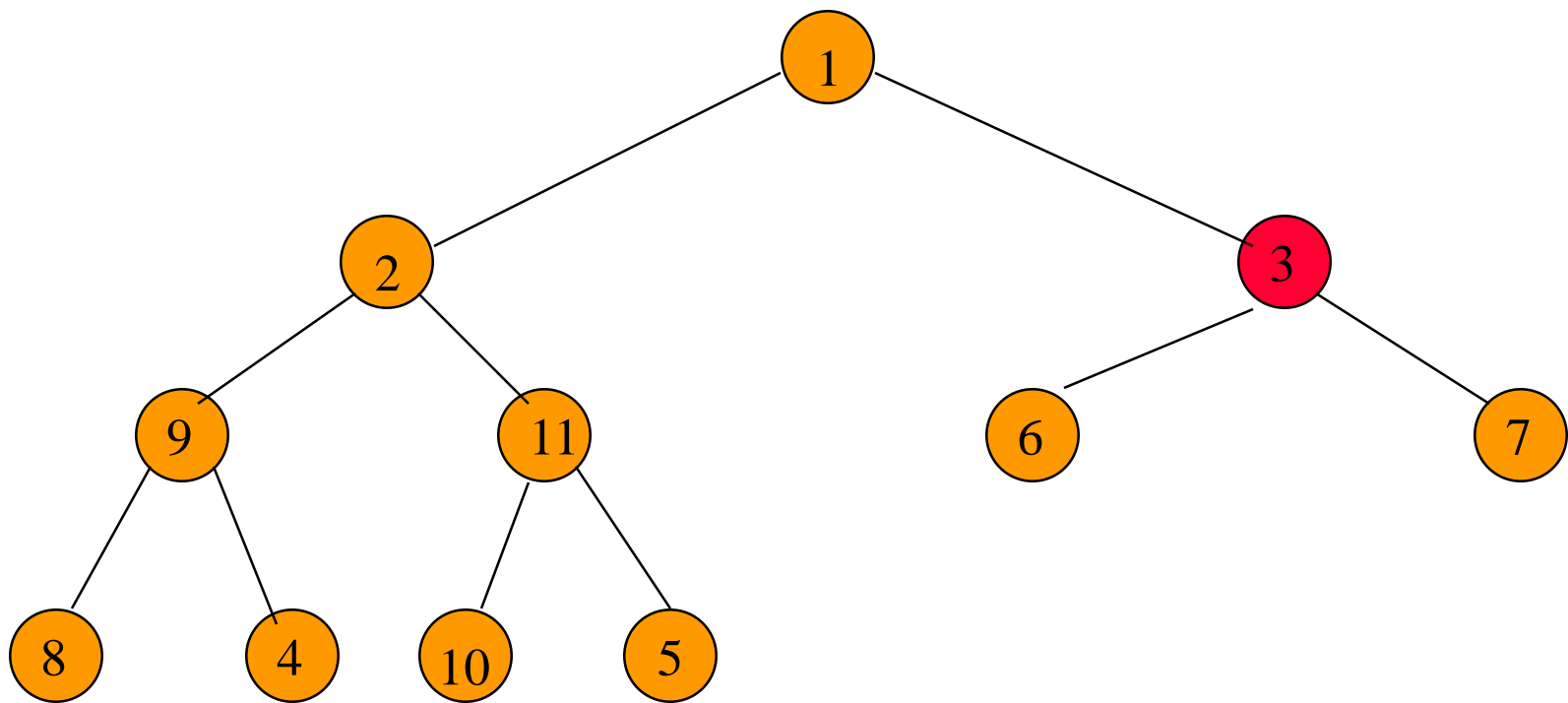
Мах овоолгыг идэхижүүлэх



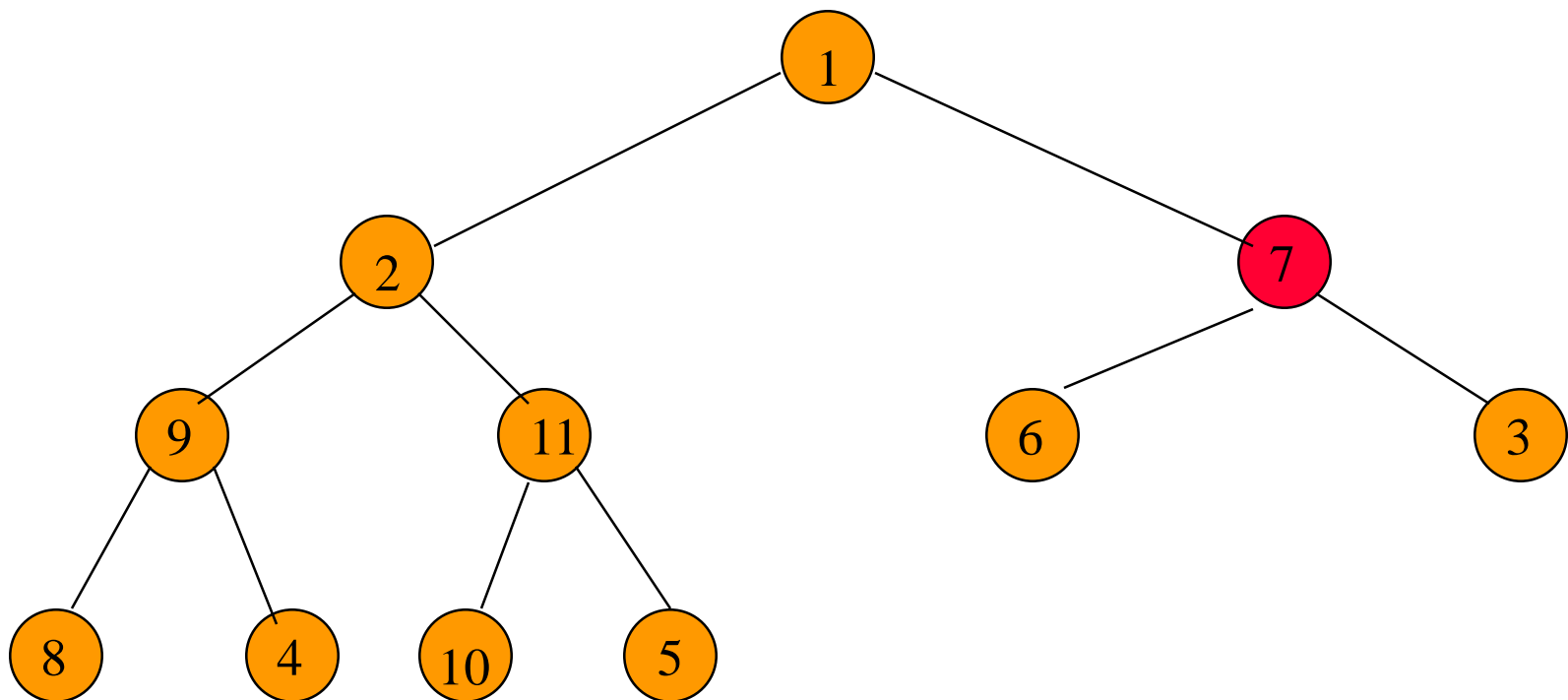
Мах овоолгыг идэхижүүлэх



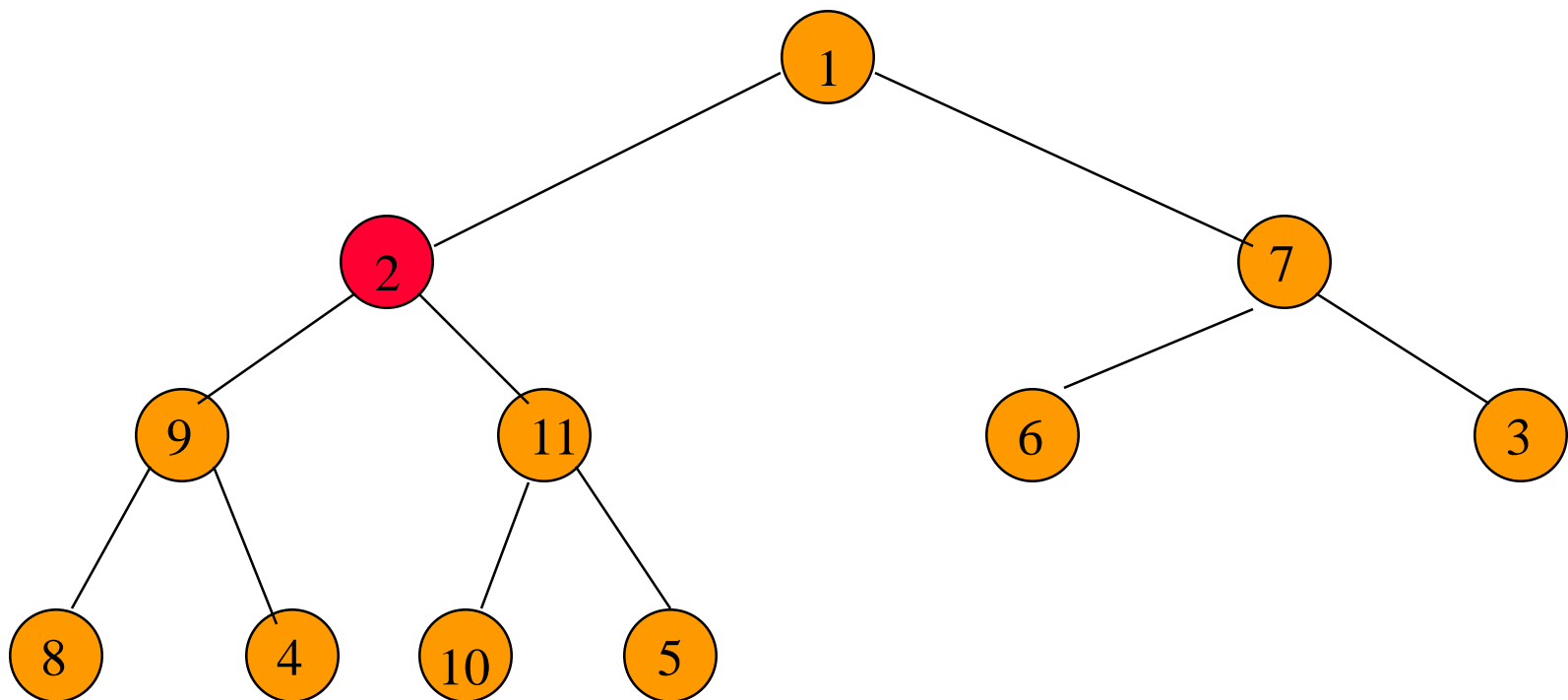
Мах овоолгыг идэхижүүлэх



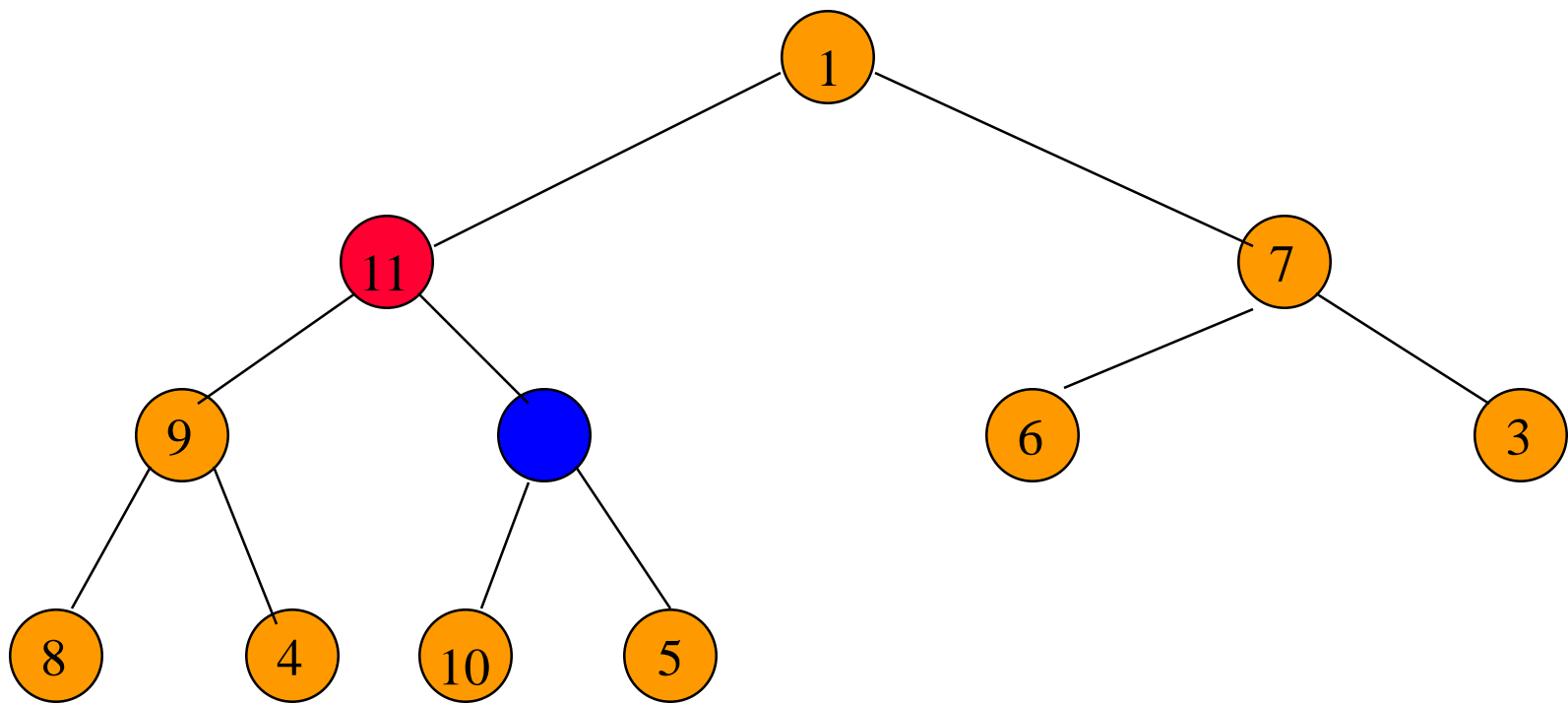
Мах овоолгыг идэхижүүлэх



Мах овоолгыг идэхижүүлэх

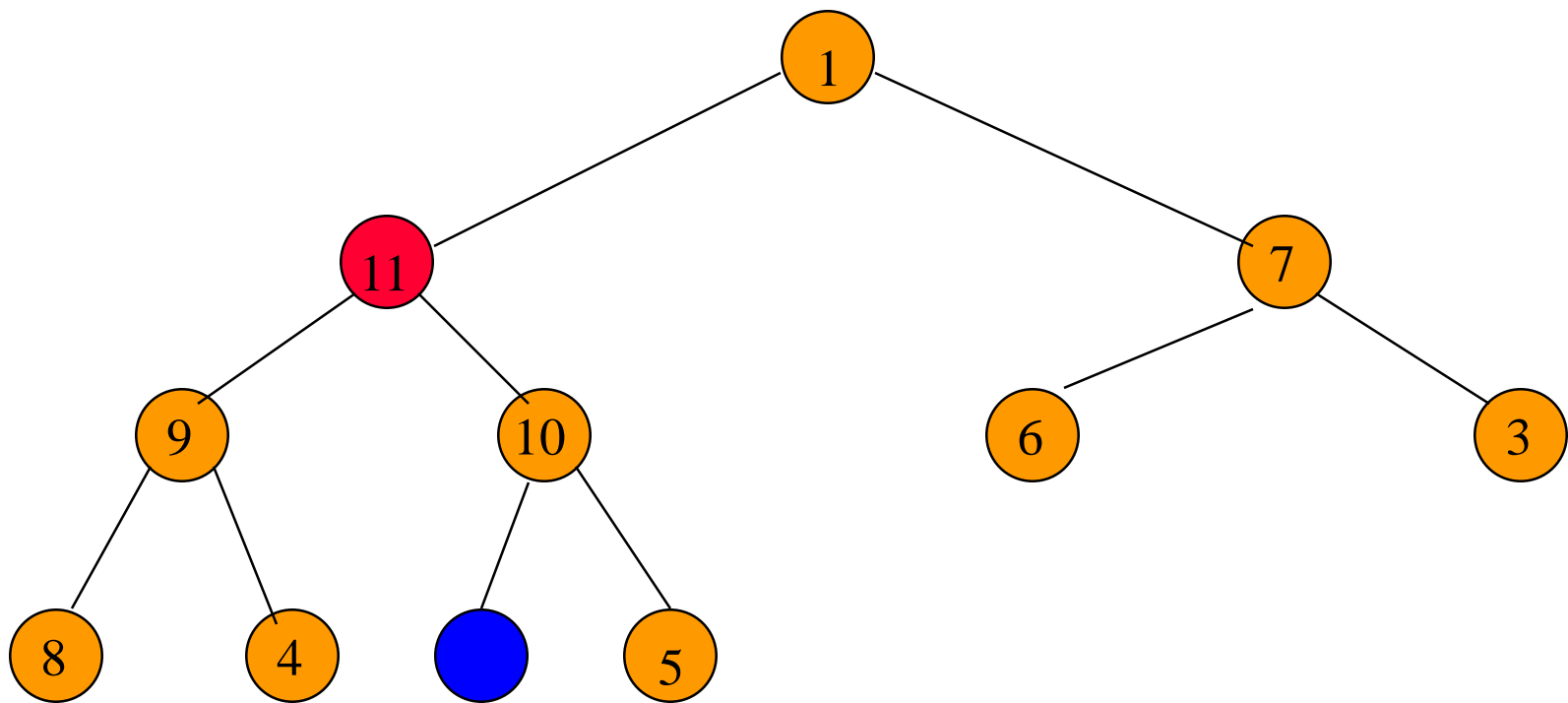


Мах овоолгыг идэхижүүлэх



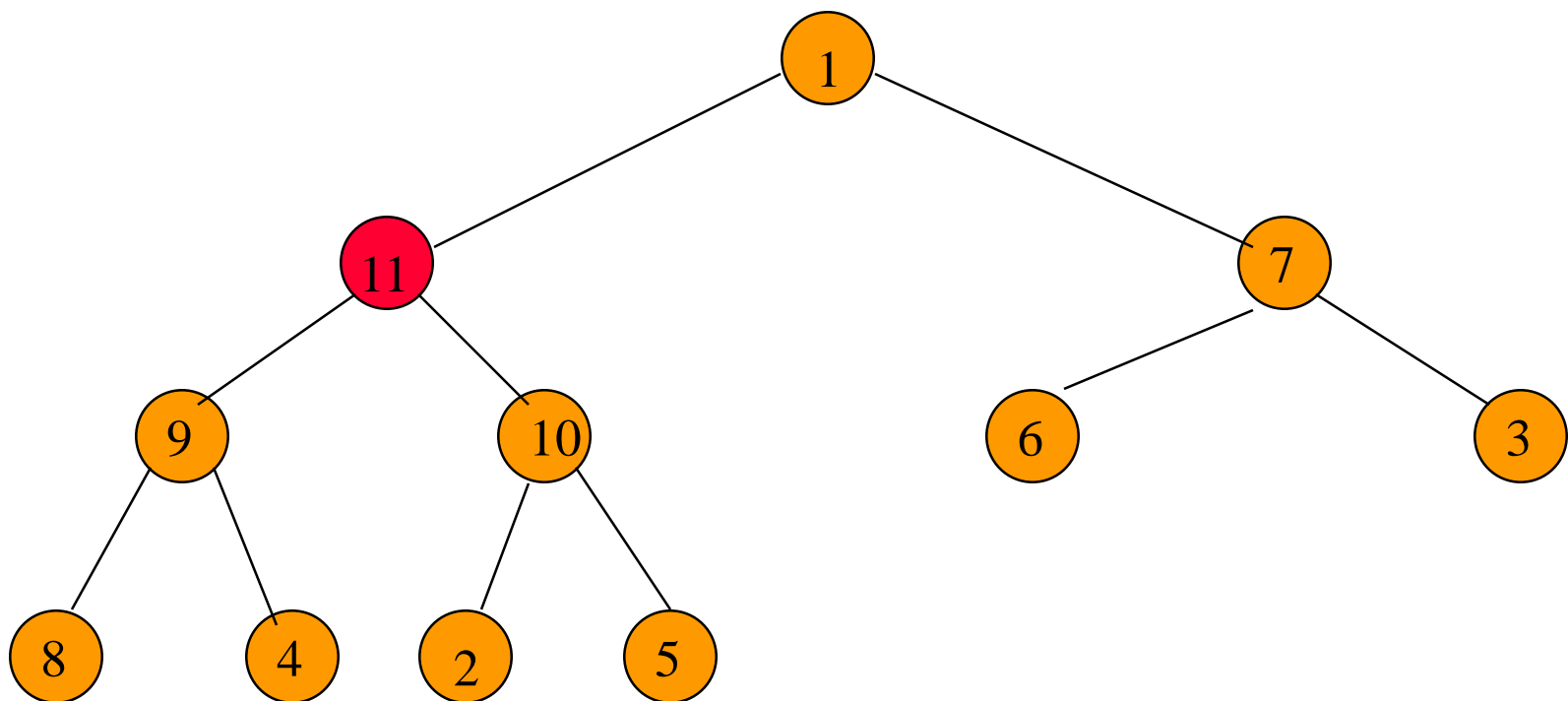
2 -н байрыг олох

Мах овоолгыг идэхижүүлэх



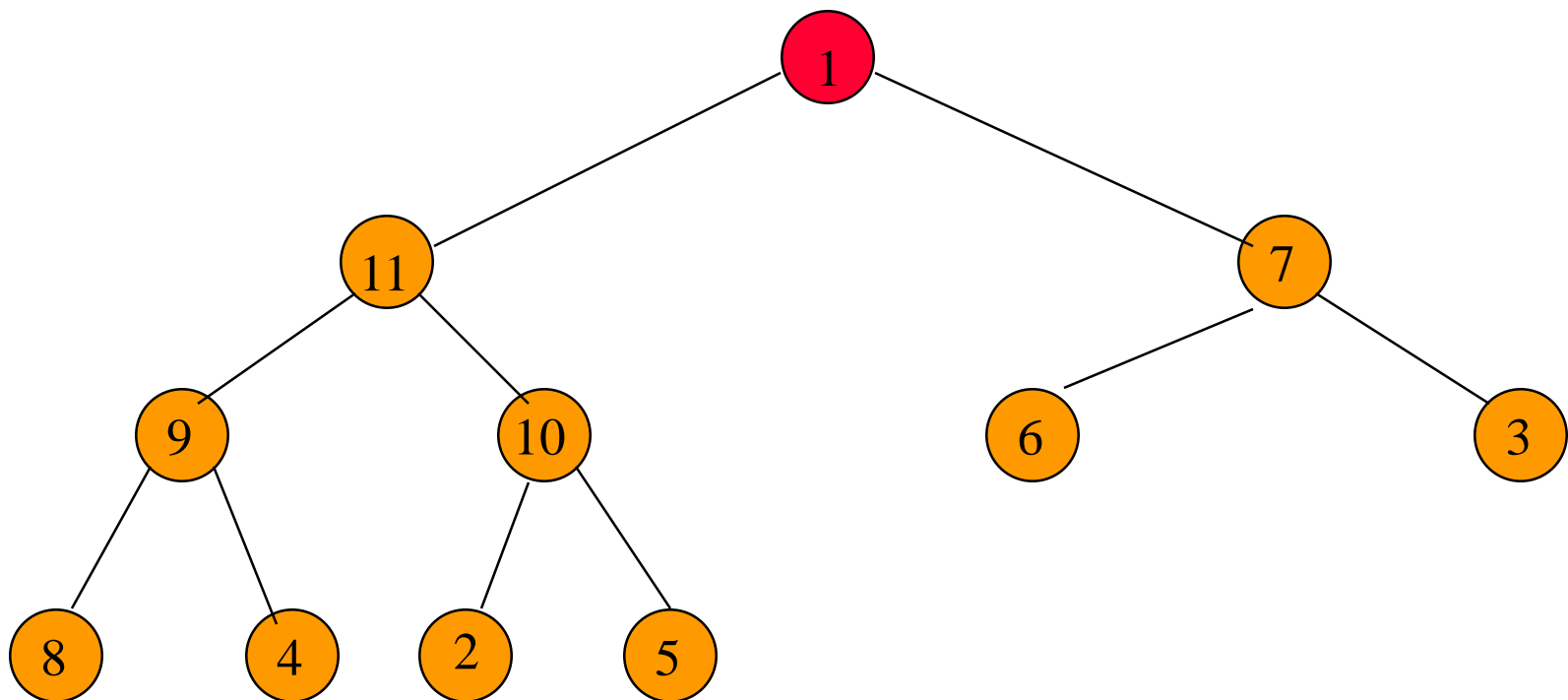
2 -н байрыг олох

Мах овоолгыг идэхижүүлэх



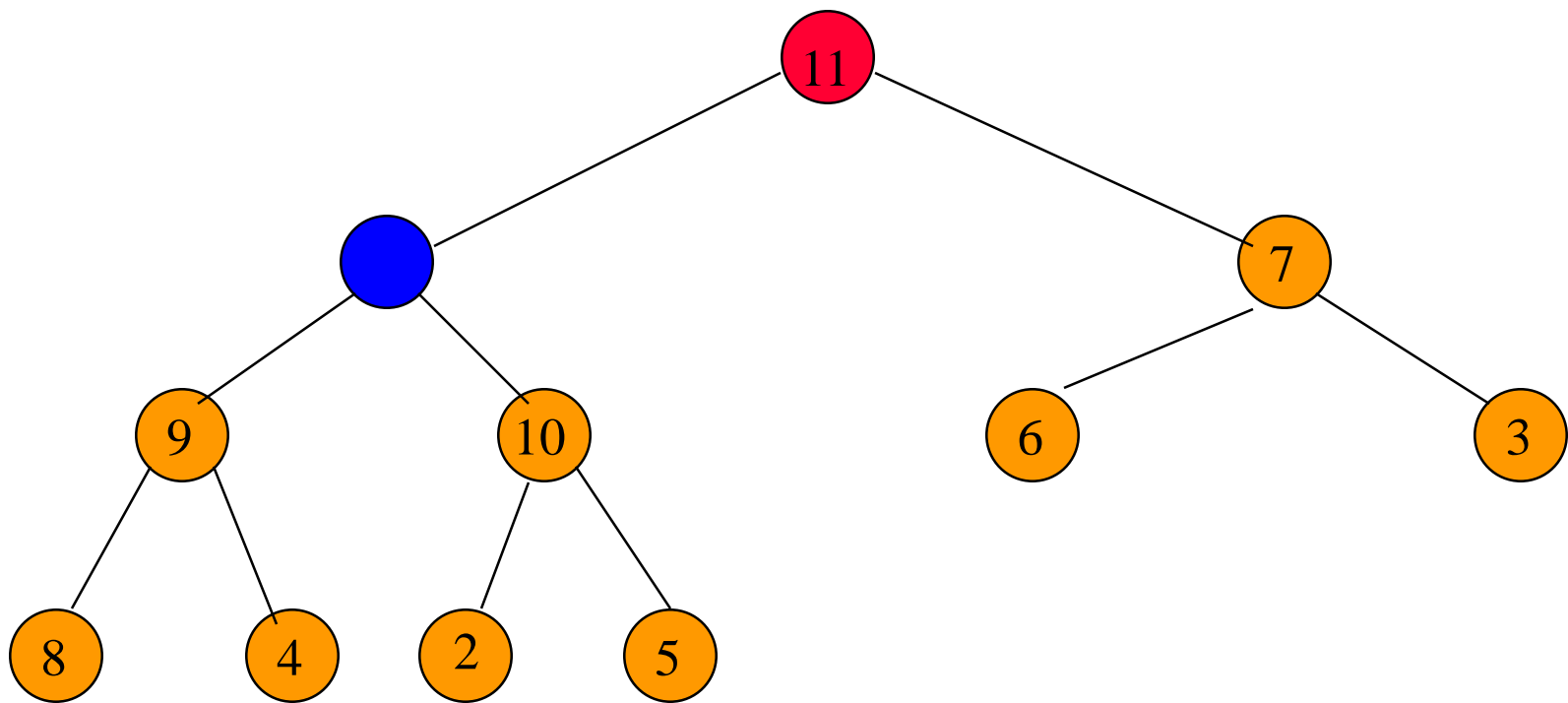
Гүйцлээ, массвын дараачийн доод байршил руу
явна.

Мах овоолгыг идэхижүүлэх



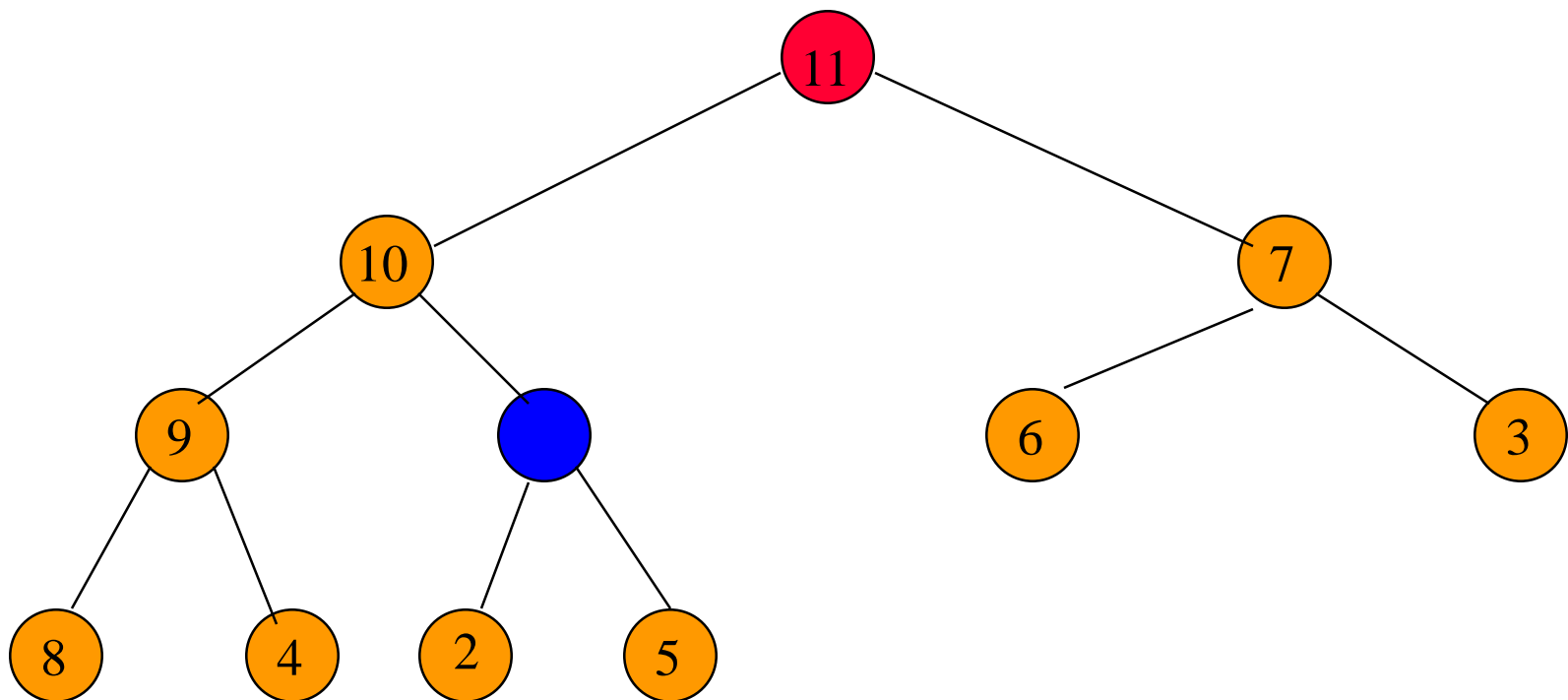
1 -н байрыг олох

Мах овоолгыг идэхижүүлэх



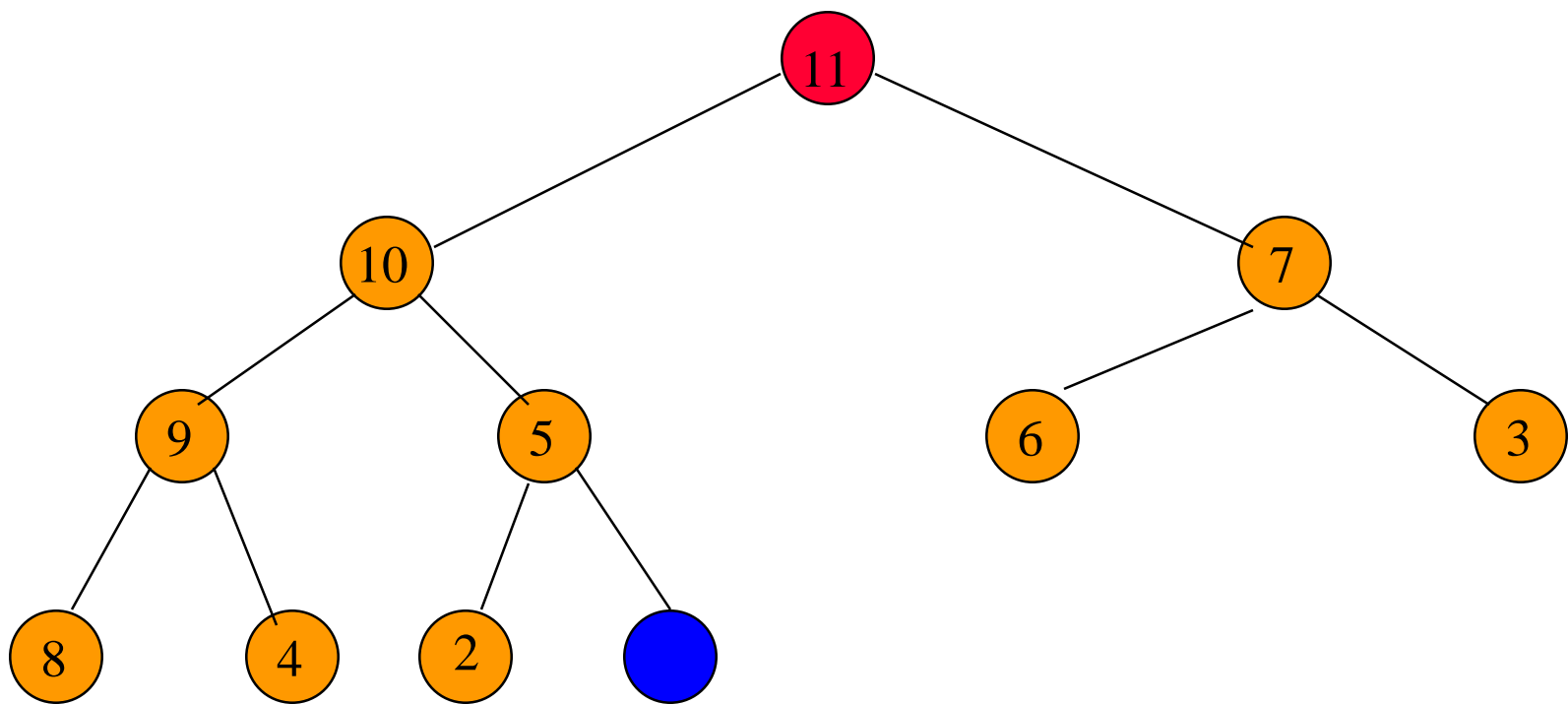
1 -н байрыг олох

Мах овоолгыг идэхижүүлэх



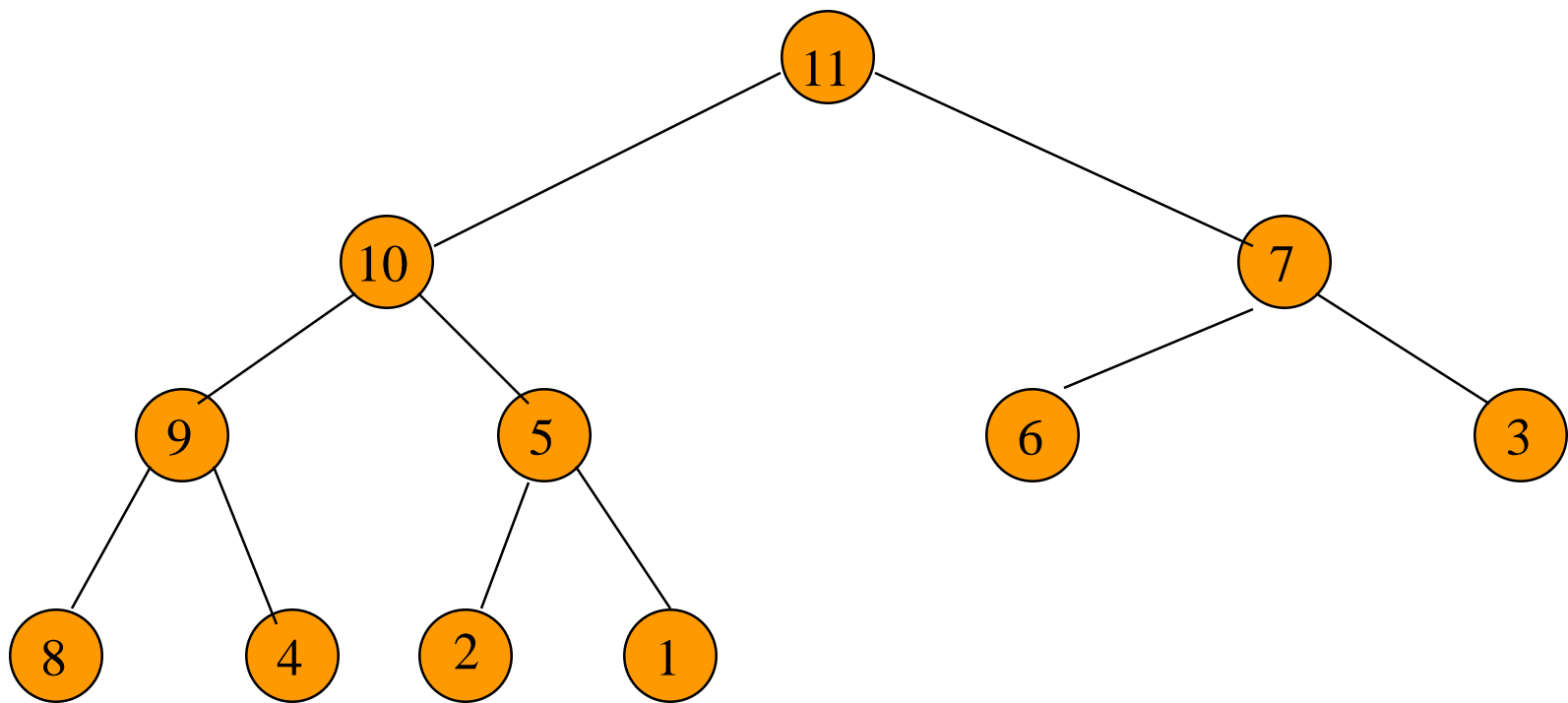
1 -н байрыг олох

Мах овоолгыг идэхижүүлэх



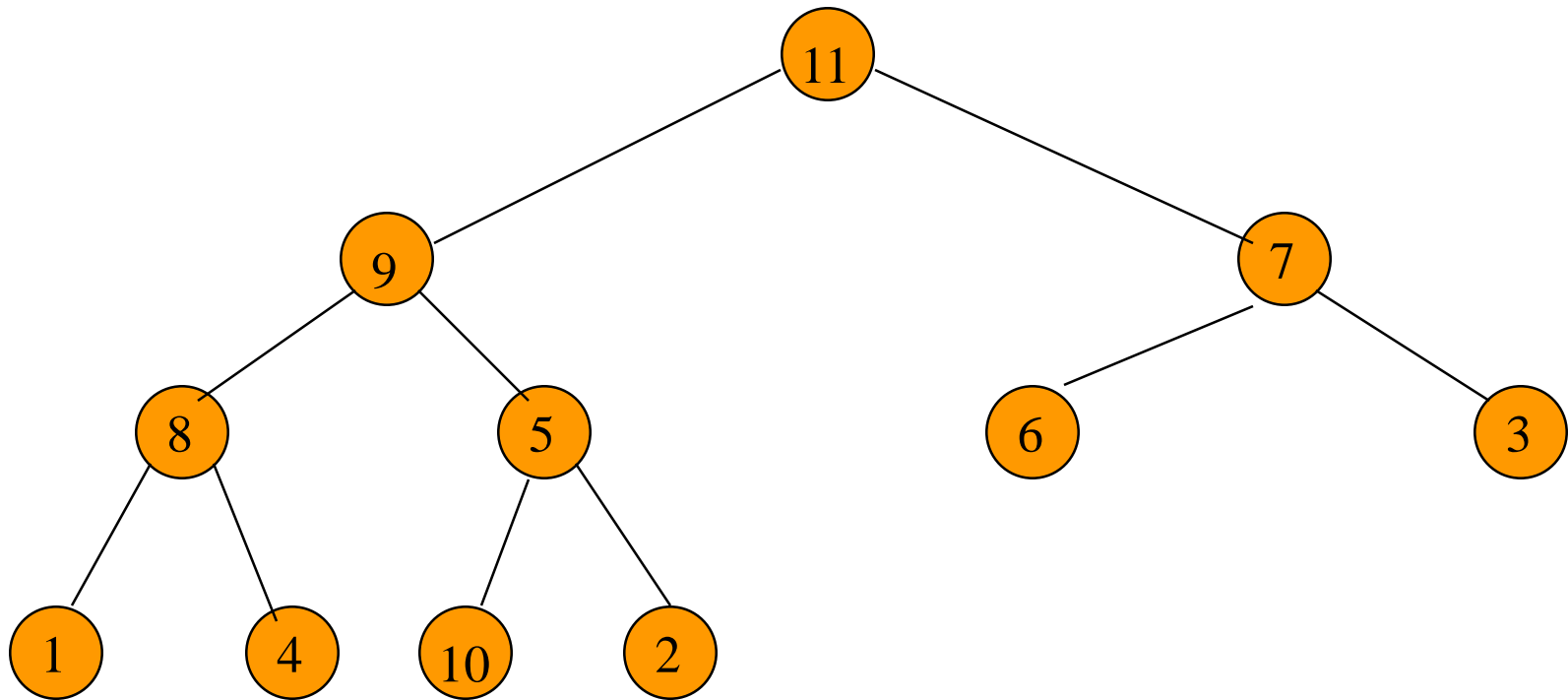
1 -н байрыг олох

Мах овоолгыг идэхижүүлэх



Гүйцлээ.

Хугацаа



Овоолгын өндөр = h .

j түвшинд үндэстэй дэд моднуудын тоо $\leq 2^{j-1}$.

Дэд мод бүрийн хугацаа $O(h-j+1)$.

Хугацаа



j түвшний моднуудын хугацаа $\leq 2^{j-1}(h-j+1) = t(j)$.

Нийт хугацаа $t(1) + t(2) + \dots + t(h-1) = O(n)$.

Зүүний-Leftist мод

Холбоостой хоёртын мод.

Овоолгын хийж чадах бүхнийг ойролцоо хугацаанд хийж чадна.

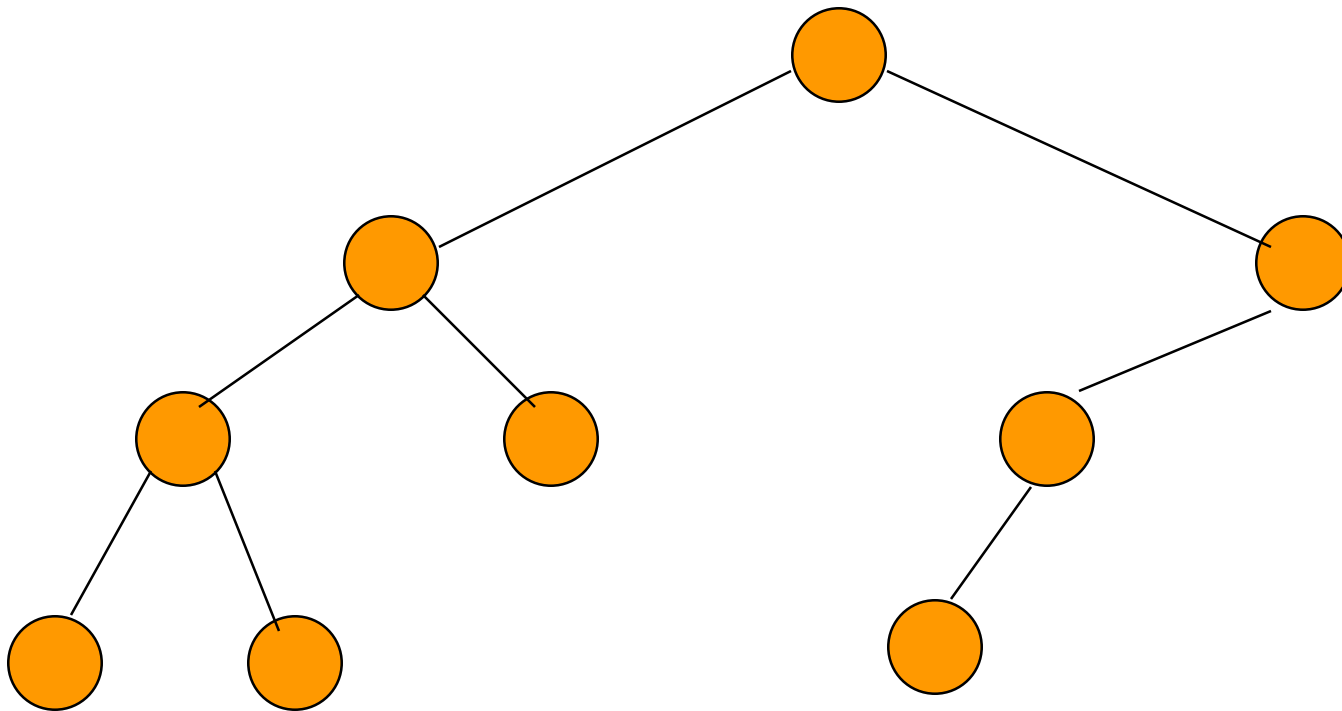
Хоёр зүүний модны эрэмбэтэй дарааллыг нэгтгэхэд $O(\log n)$ хугацаа шаардана.

Өргөтгөсөн хоёртын мод

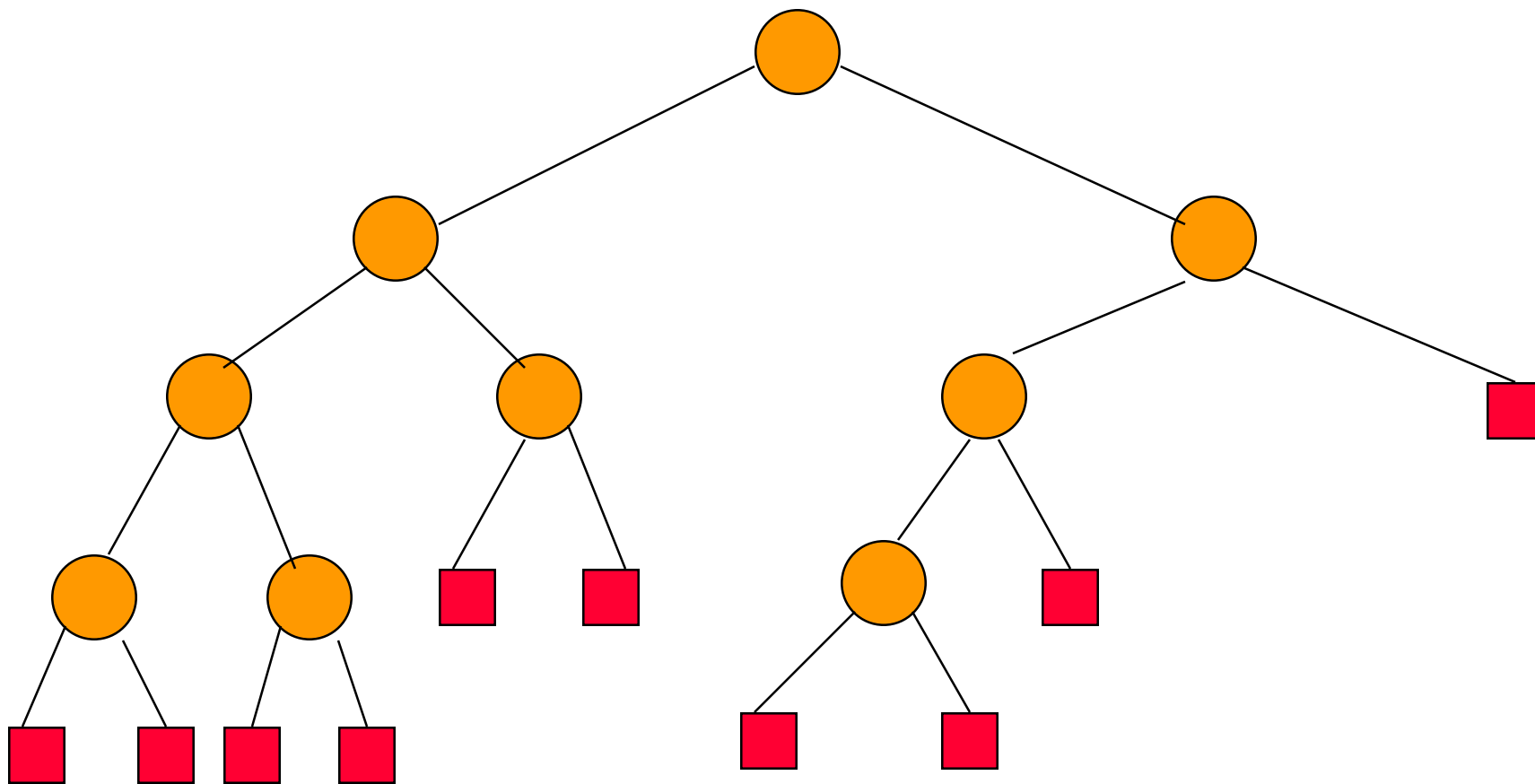
Дурын хоёртын модноос эхлээд
хоосон дэд мод үүсгэж гадны
зангилааг нэмнэ

Үр дүнд нь өргөтгөсөн хоёртын
мод үүснэ.

Хоёртын мод



Өргөтгөсөн хоёртын мод



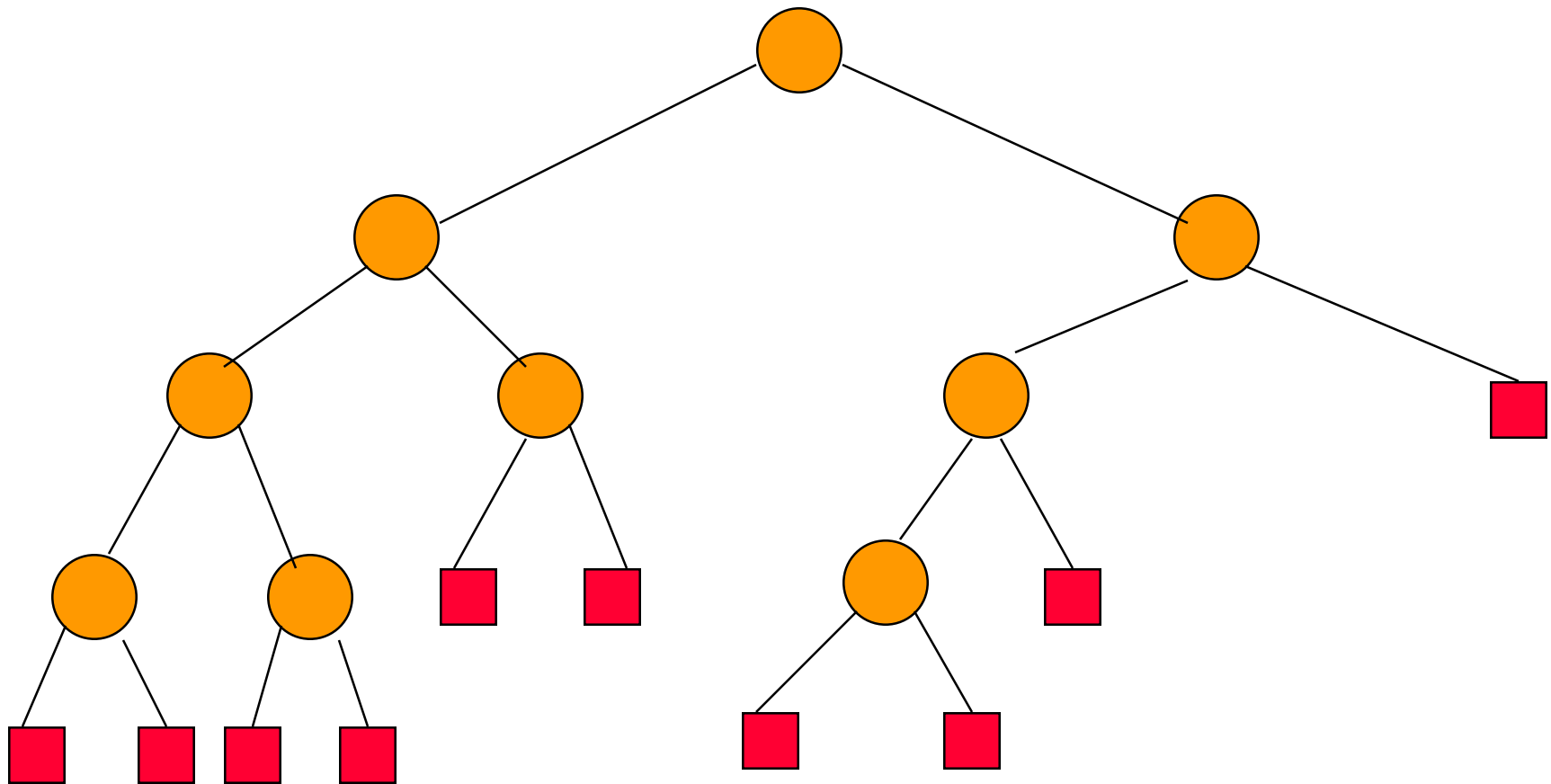
Гадны зангилааны тоо **n+1**

$s()$ функц

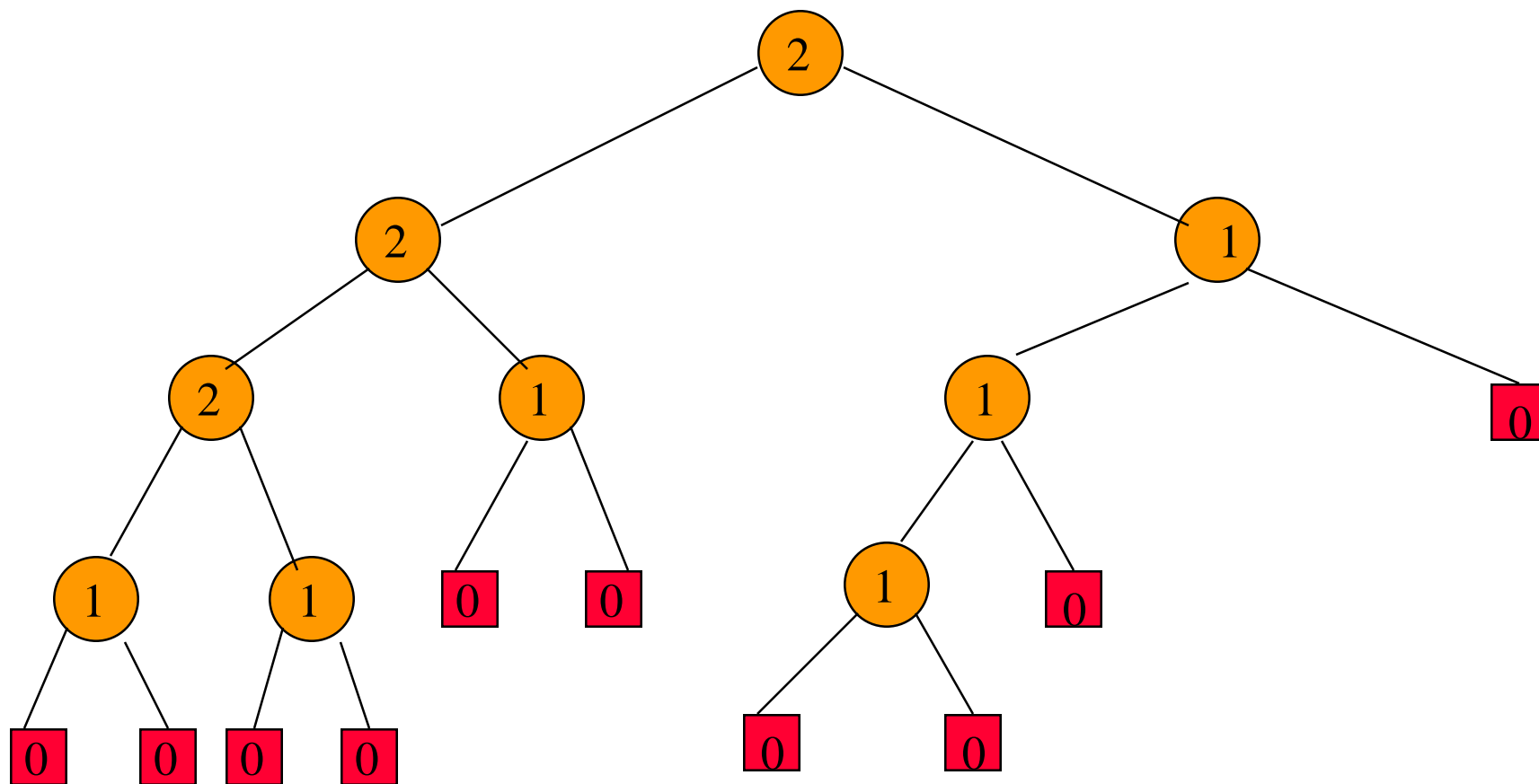
Өргөтгөсөн хоёртын модны дурын x зангилааны хувьд,

$s(x)$ нь x –ээс түүнийг үндэс болгосон дэд модны гадны зангилаа хүртэлх хамгийн богино зам.

$s()$ утга - жишээ



s() утга - жишээ



$s()$ —Н ШИНЖ

Хэрвээ x гадны зангилаа бол $s(x) = 0$.

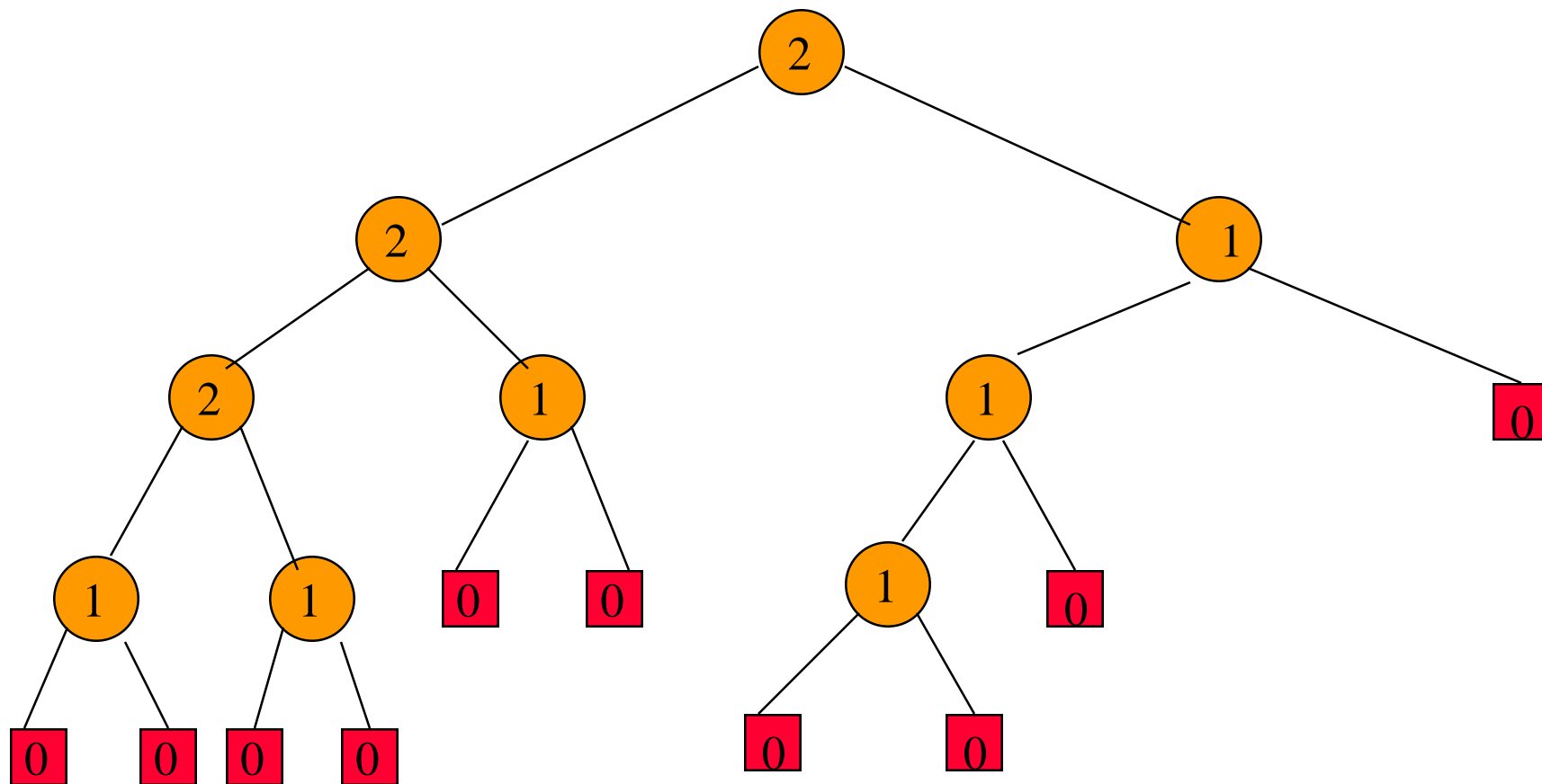
Бусад тохиолдолд,

$$s(x) = \min \{ s(\text{leftChild}(x)), \\ s(\text{rightChild}(x)) \} + 1$$

Өндрөөр налсан зүүний мод

Хоёртын мод (өндрөөр налсан) зүүний
мод болохын тулд бүх дотоод
зангилаа x -н хувьд $s(\text{leftChild}(x)) \geq$
 $s(\text{rightChild}(x))$

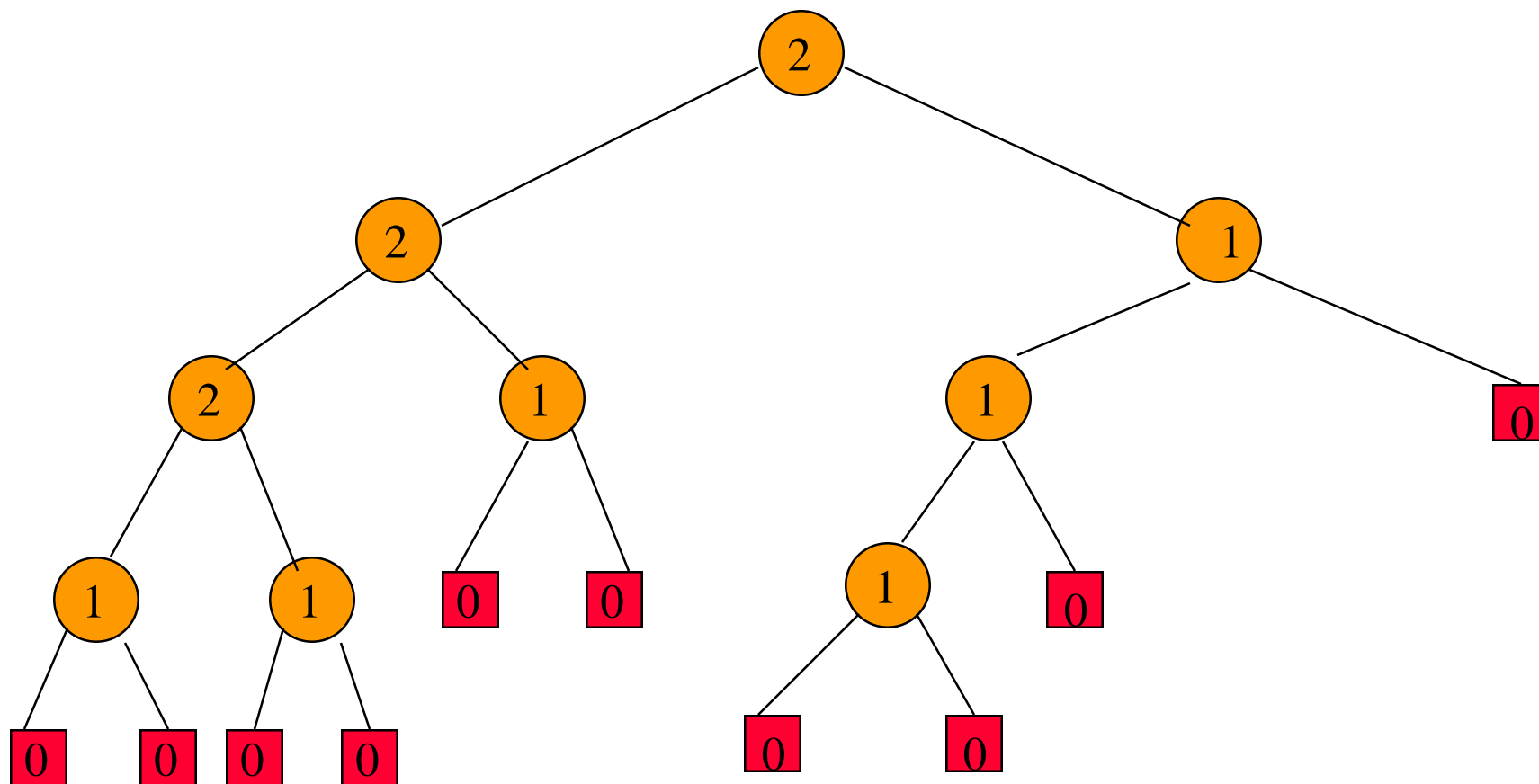
Зүүний мод



Зүүний модны шинж 1

Зүүний модонд, хамгийн баруун зам нь
үндсээс гадны зангилаа хүртэлх
хамгийн богино зам болдог бөгөөд урт
нь $s(\text{root})$.

Зүүний мод



Хамгийн баруун замын урт 2.

Зүүний модны шинж 2

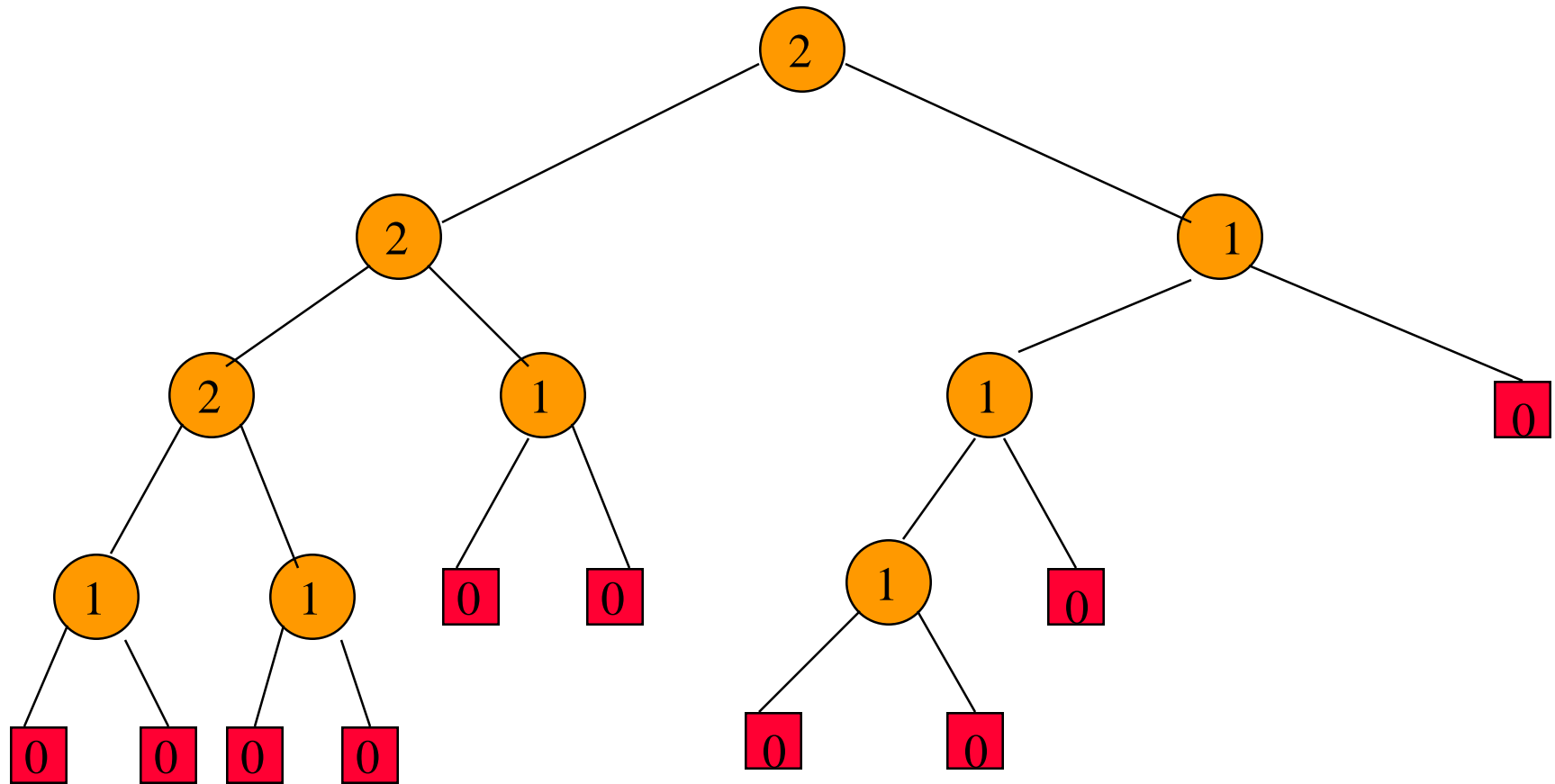
Дотоод зангилааны тоо дор хаяж

$$2^{s(\text{root})} - 1$$

1 -ээс $s(\text{root})$ түвшингүүдэд гадны зангилаа байхгүй болохоор

$$s(\text{root}) \leq \log(n+1)$$

Зүүний мод



1, 2 -р тҮВШИНД ГАДНЫ ЗАНГИЛАА АЛГА.

Зүүний модны шинж 3

Хамгийн баруун замын урт нь $O(\log n)$,
үүнд n – зүүний модны зангилааны
тоо.

1 , 2 –р шинжээс гарч ирсэн.

Зүүний мод эрэмбэтэй дараалал болох

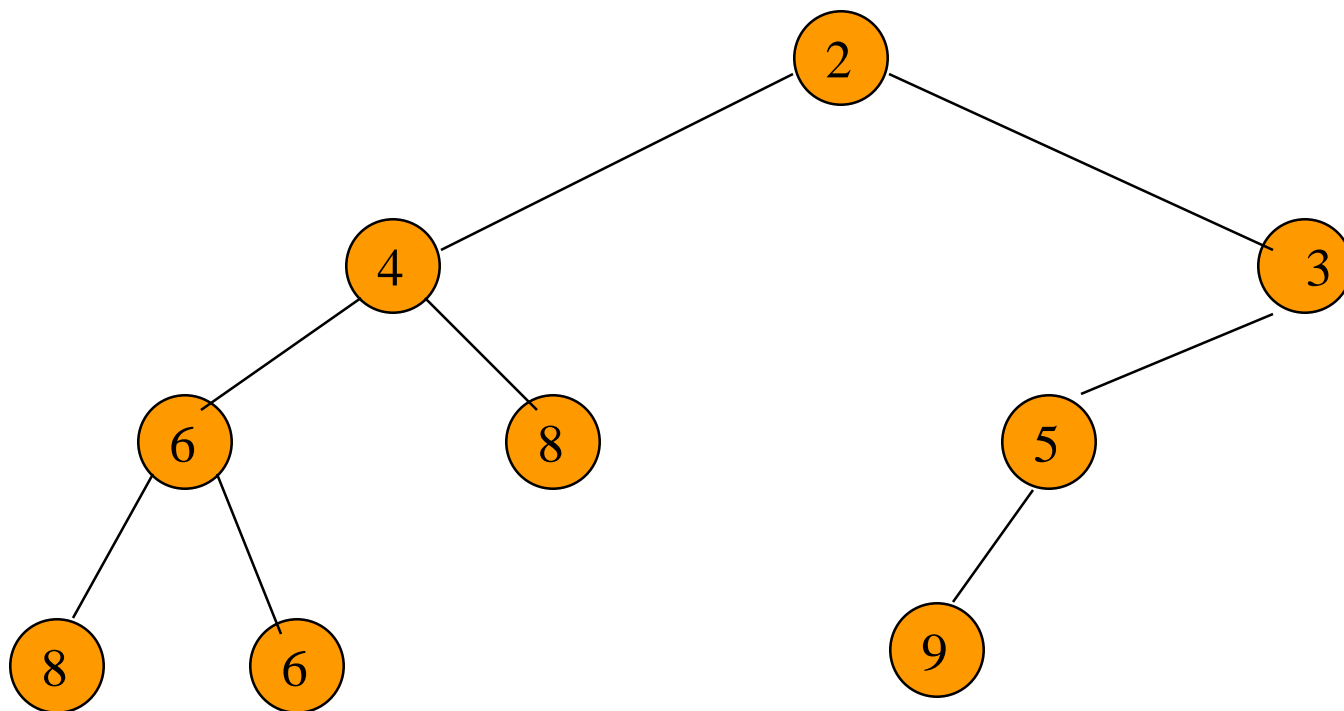
Min зүүний мод бол min мод.

Тэгэхээр min эрэмбэтэй модны байдлаар ашиглана.

Max зүүний мод бол max мод.

Тэгэхээр max эрэмбэтэй модны байдлаар ашиглана.

Min зүүний мод



Min зүүний модны зарим үйлдлүүд

put()

remove()

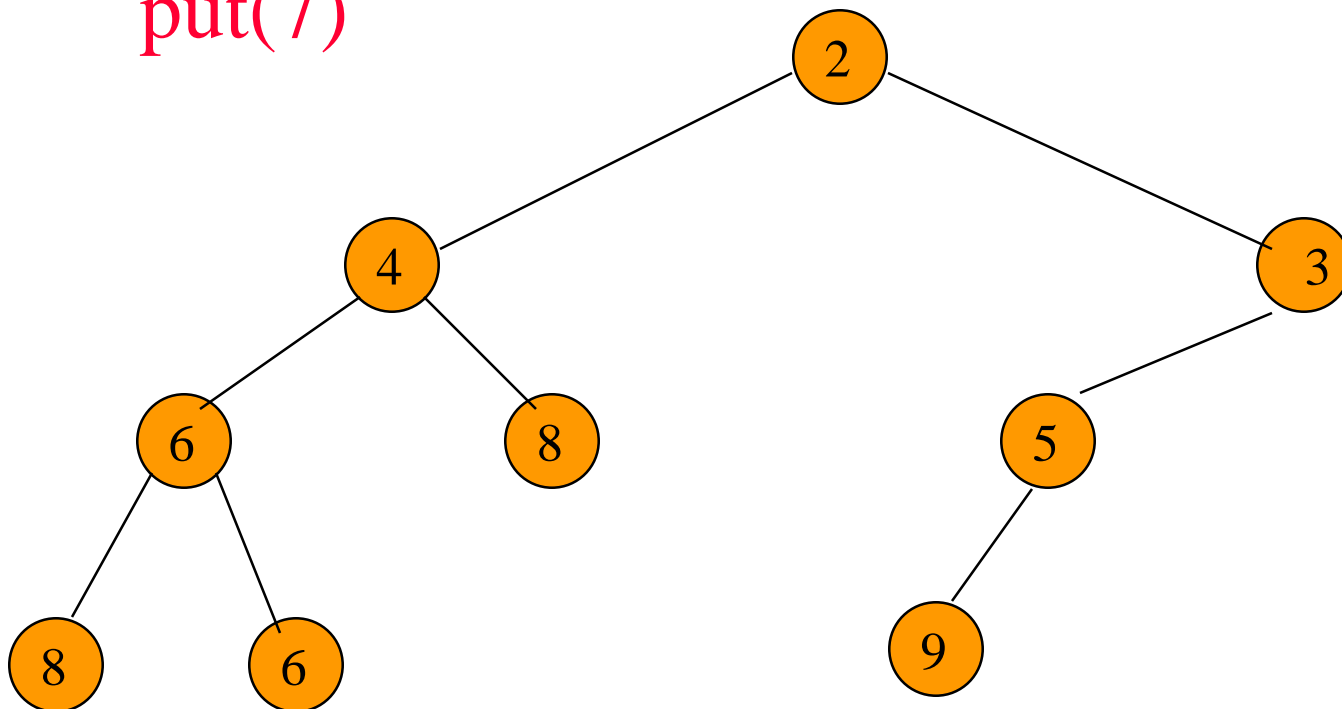
meld()

initialize()

put() , remove() -г meld() -д ашиглана

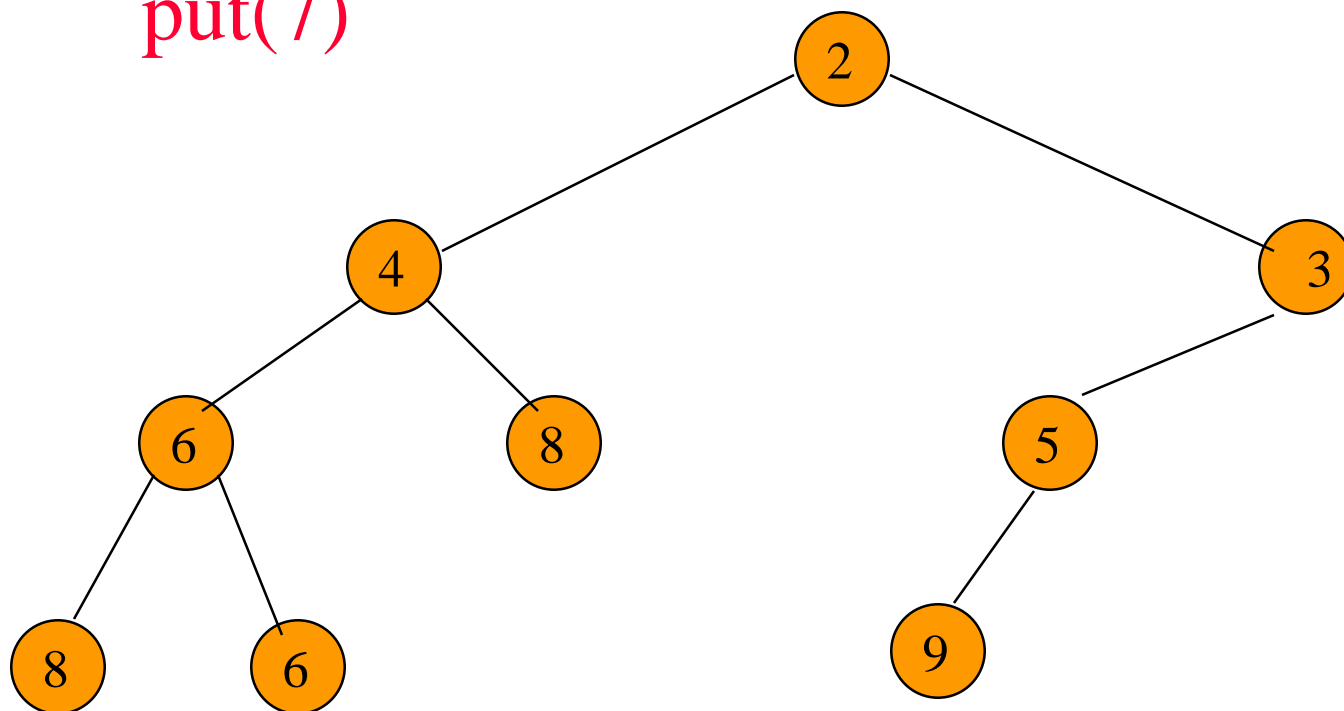
Put үйлдэл

put(7)



Put үйлдэл

put(7)

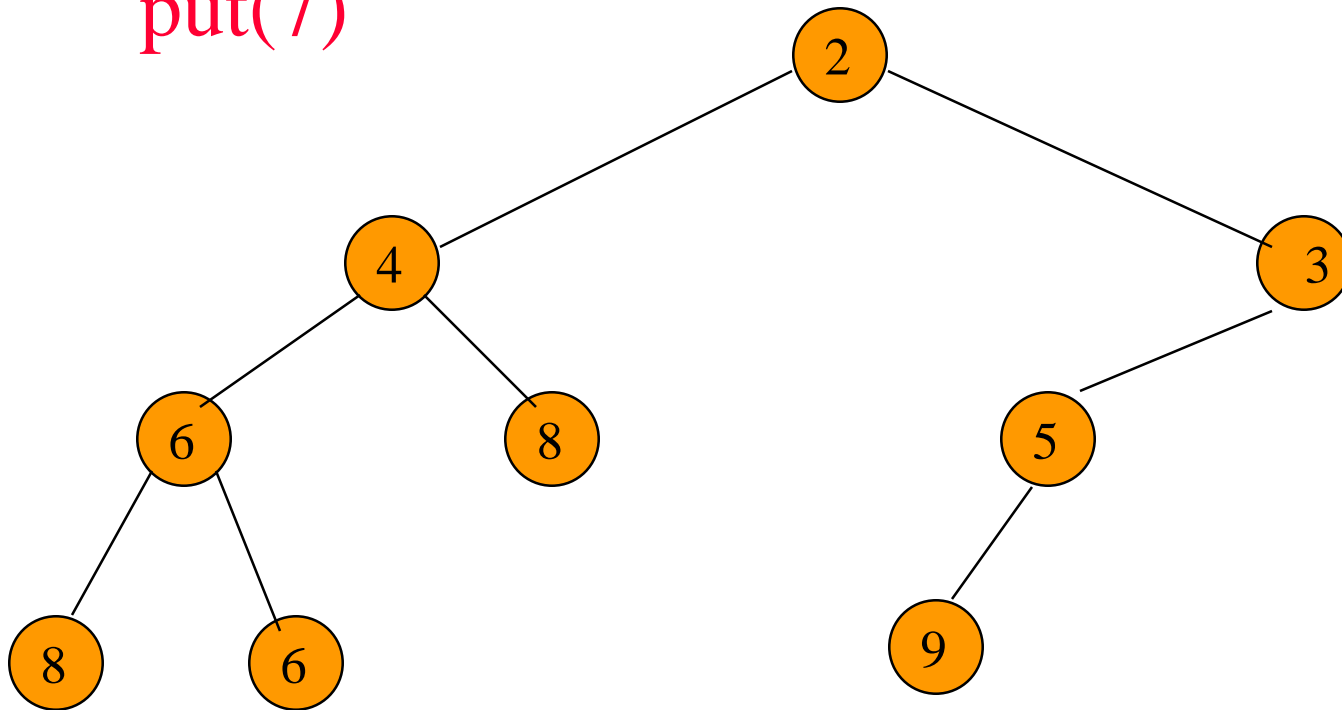


Дан зангилаатай min зүүний мод үүсгэх.

7

Put үйлдэл

put(7)

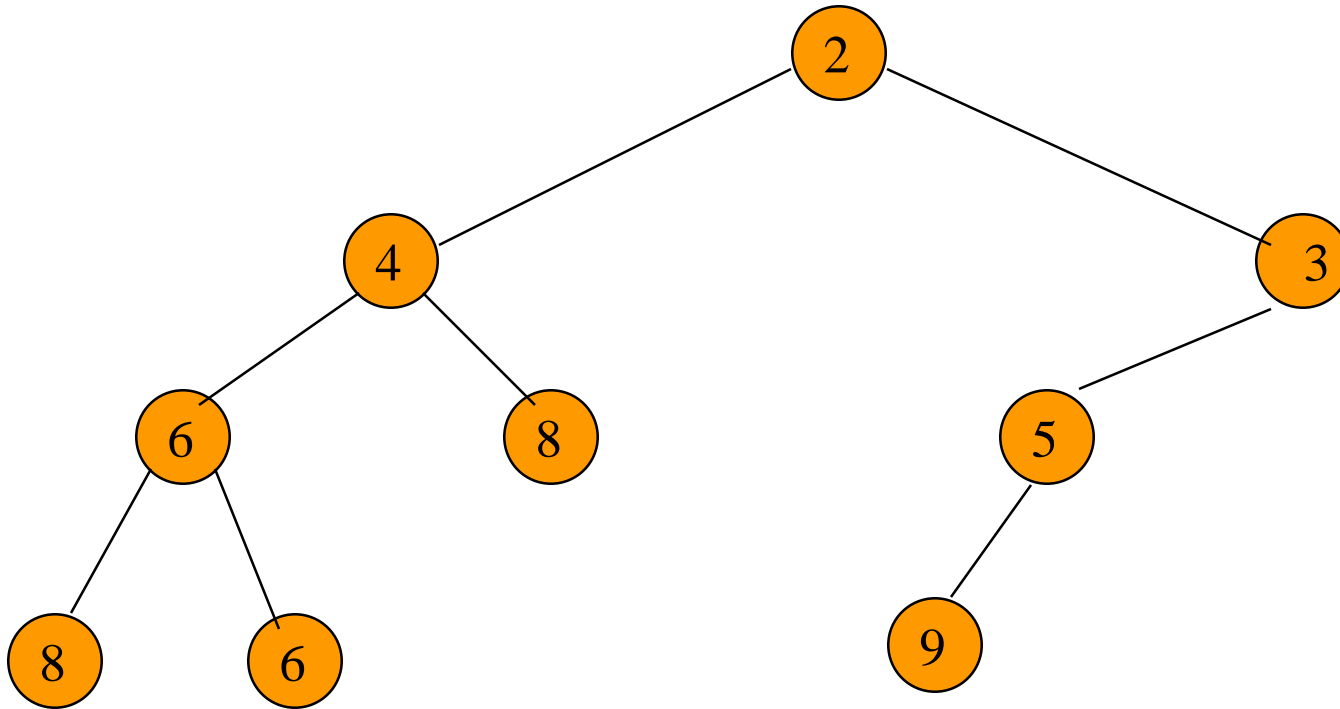


Дан зангилаатай min зүүний мод үүсгэх.

Хоёр min зүүний модыг нэгтгэх.

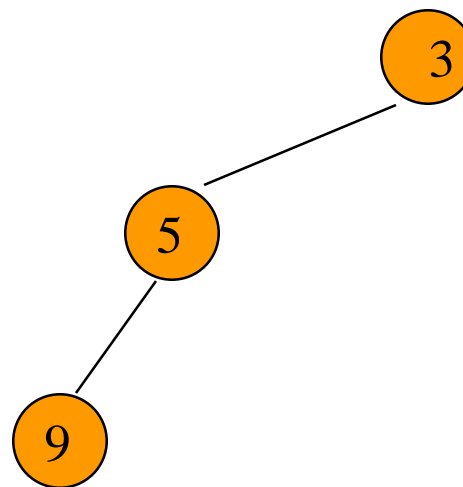
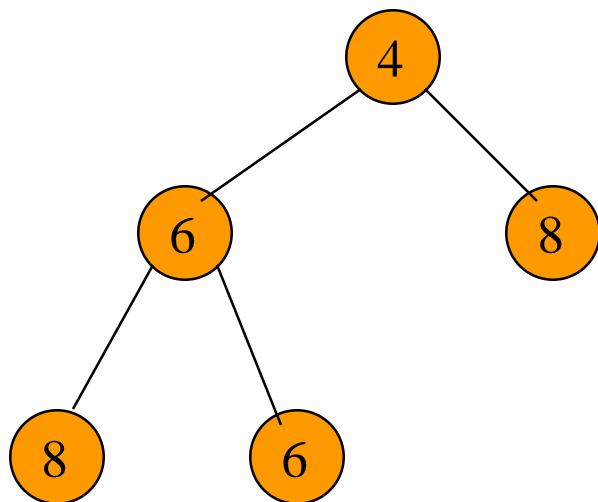
7

Remove Min



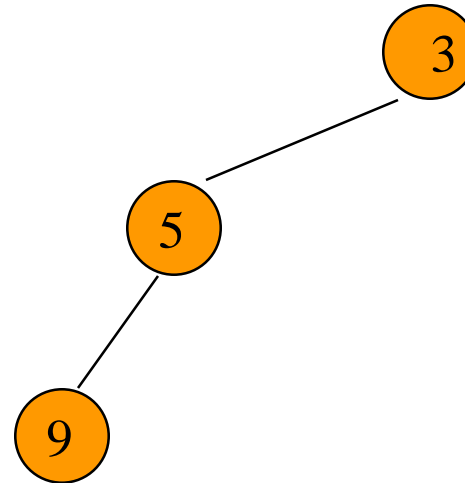
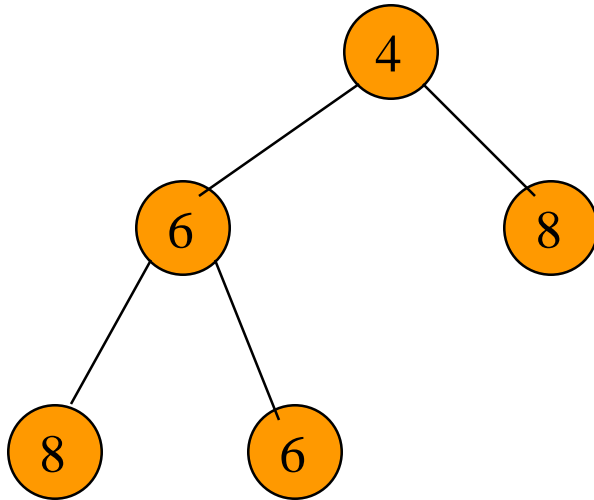
Remove Min

2



Үндсийг устгах.

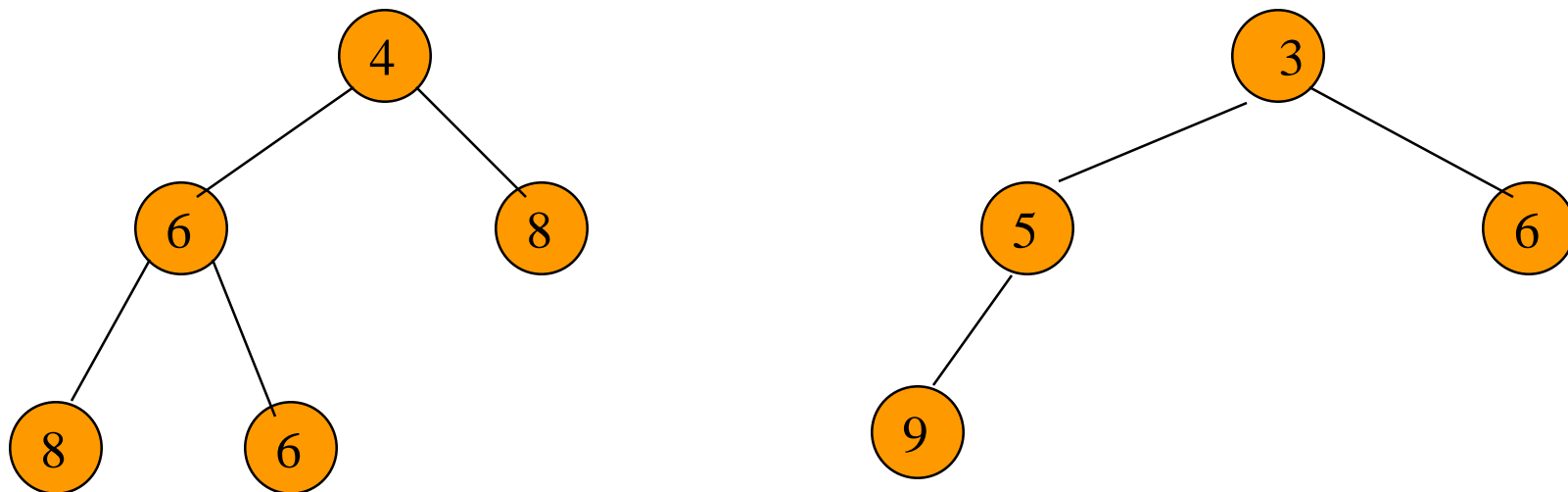
Remove Min



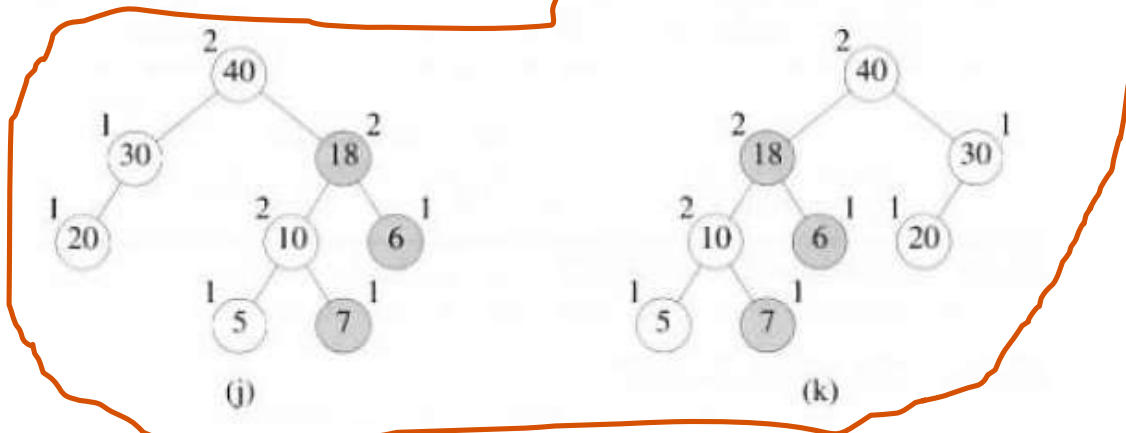
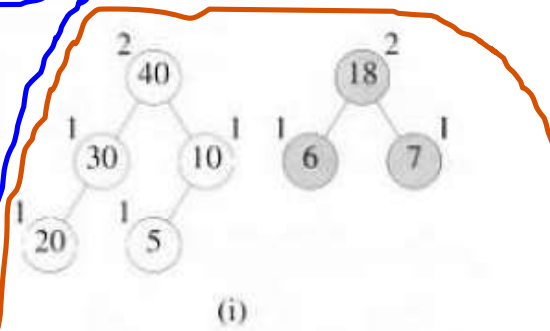
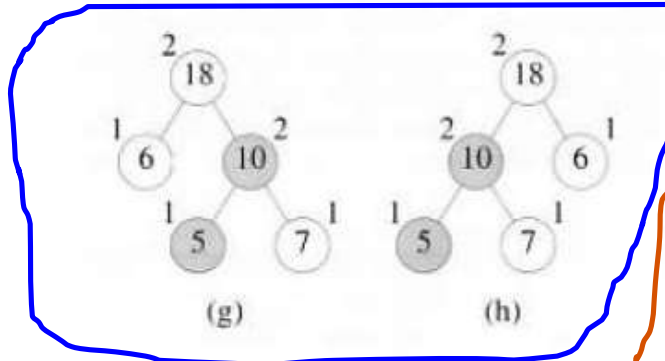
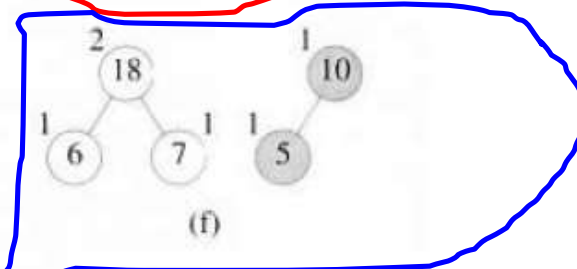
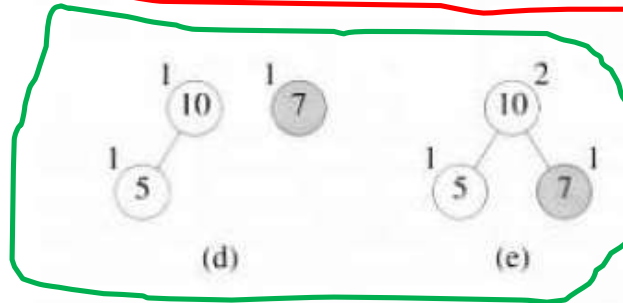
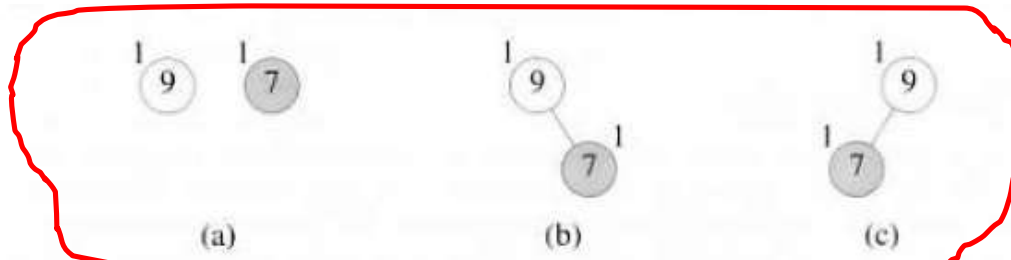
Үндсийг устгах.

Хоёр дэд модыг нийлүүлэх.

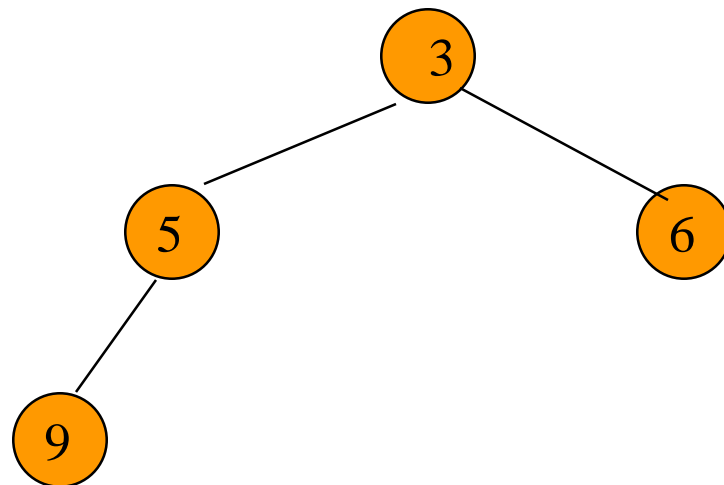
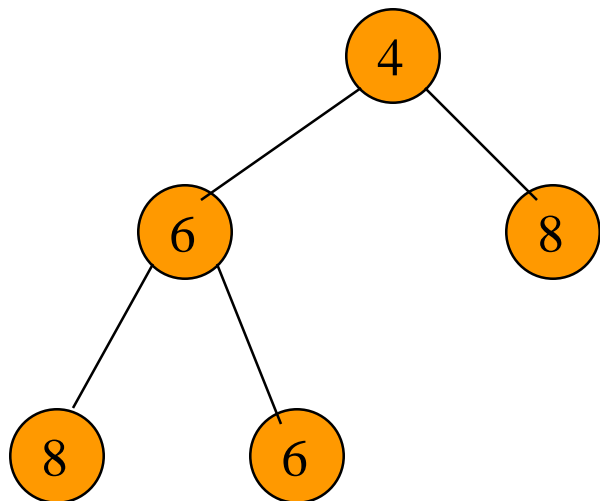
Хоёр Min зүүний модыг нийлүүлэх



Хамгийн баруун талын замаар нэвтрэхэд хугацаа
логарифмын хамаарлаар тооцогдоно.

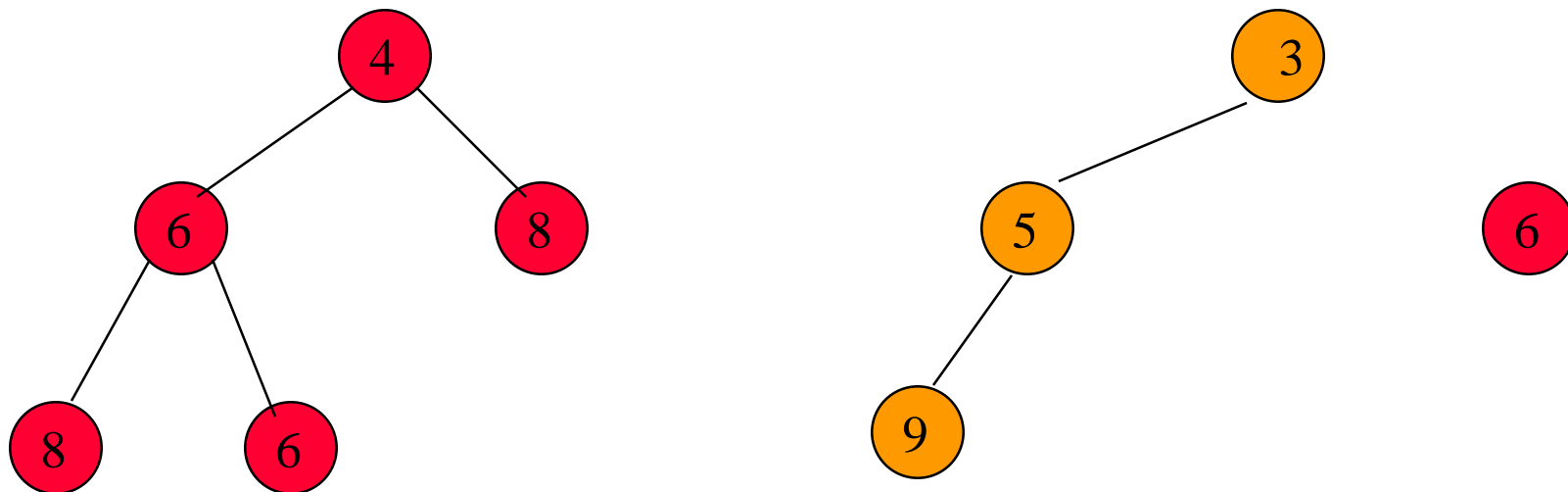


Хоёр Min зүүний модыг нийлүүлэх



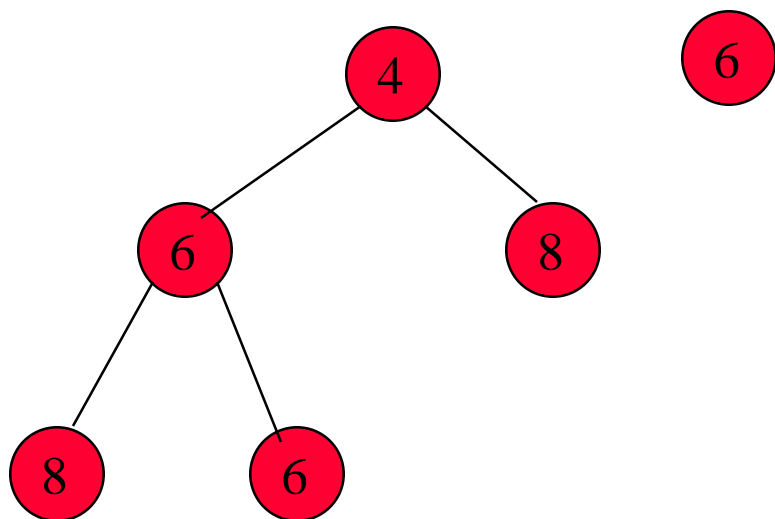
Хамгийн бага үндэстэй модны баруун дэд мод болон бусад модыг нэгтгэх.

Хоёр Min зүүний модыг нийлүүлэх



Хамгийн бага үндэстэй модны баруун дэд мод болон бусад модыг нэгтгэх.

Хоёр Min зүүний модыг нийлүүлэх



Хамгийн бага үндэстэй модны баруун дэд мод болон бусад модыг нэгтгэх.

Хоёр Min зүүний модыг нийлүүлэх

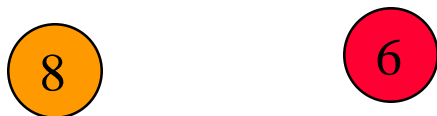
8

6

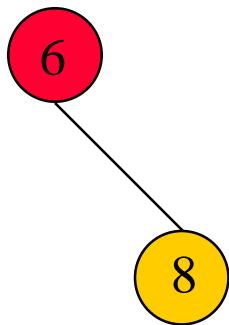
Хамгийн бага үндэстэй модны баруун дэд мод болон бусад модыг нэгтгэх.

6 –н баруун дэд мод хоосон байна. Иймд хамгийн бага үндэстэй модны баруун дэд мод болон бусад модыг нэгтгэсний үр дүн нь бусад мод болно.

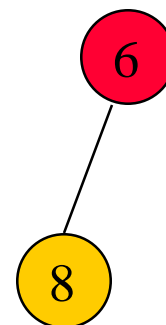
Хоёр Min зүүний модыг нийлүүлэх



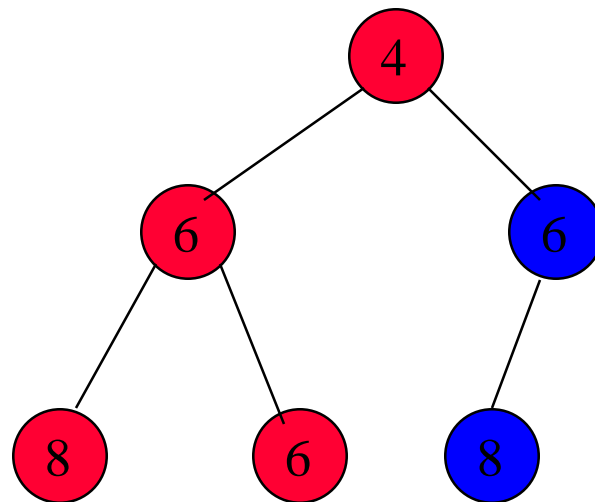
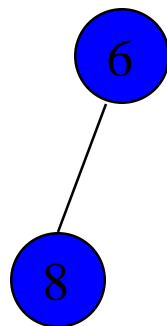
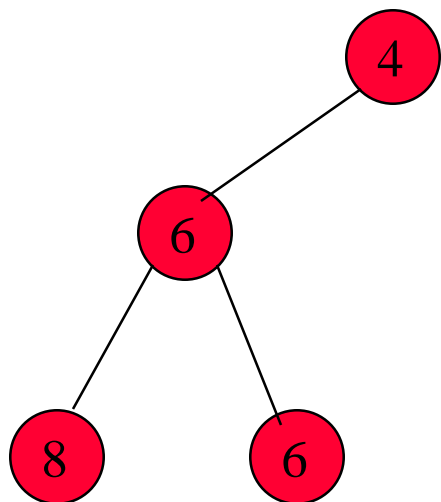
Бага үндэстэй баруун дэд модноос нэгтгэсэн дэд мод гаргах.



Хэрвээ $s(\text{left}) < s(\text{right})$ бол зүүн, баруун дэд моднуудын байрыг солих



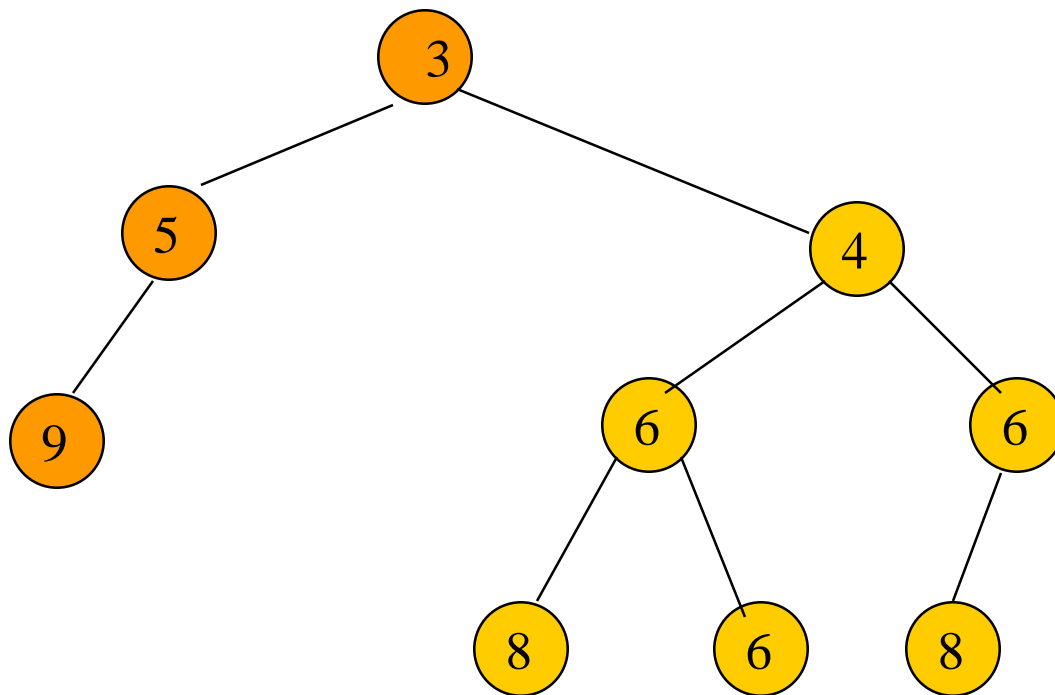
Хоёр Min зүүний модыг нийлүүлэх



Бага үндэстэй баруун дэд модноос нэгтгэсэн дэд мод гаргах.

Хэрвээ $s(\text{left}) < s(\text{right})$ бол зүүн, баруун дэд моднуудын байрыг солих

Хоёр Min зүүний модыг нийлүүлэх



Бага үндэстэй баруун дэд модноос нэгтгэсэн дэд мод гаргах.

Хэрвээ $s(\text{left}) < s(\text{right})$ бол зүүн, баруун дэд моднуудын байрыг солих

Хоёр Min зүүний модыг нийлүүлэх

