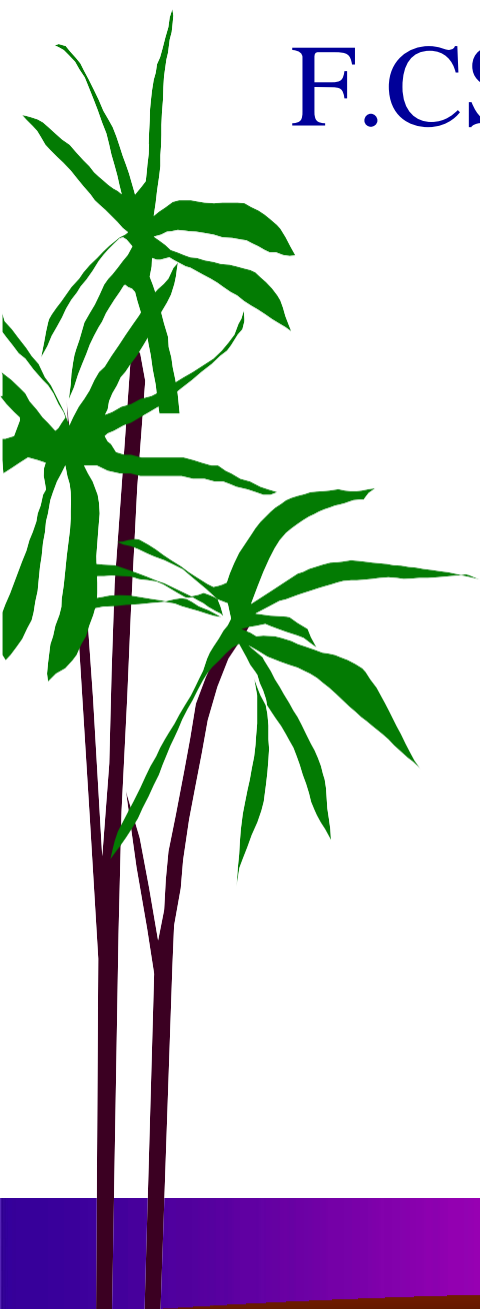
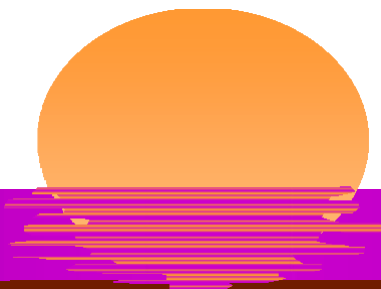


Ү.СS203 - Өгөгдлийн бүтэц ба алгоритм 2022-2023

Д.Золбоо
МТ-н салбарын багш



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-овоо,
leftist tree-зүүний мод.

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

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

Хэрэглээ

Эрэмбэлэлт

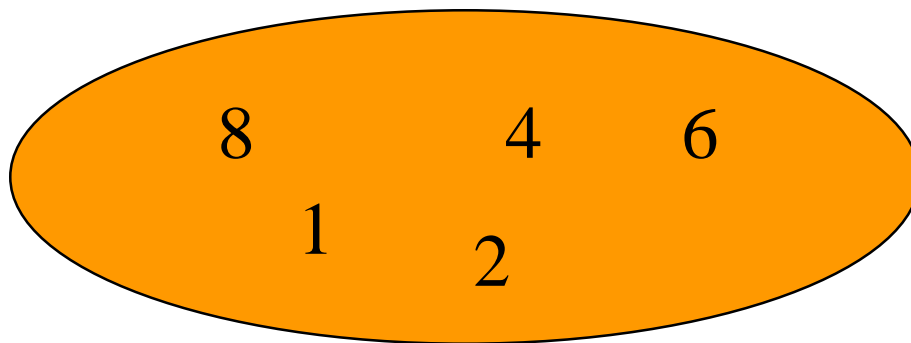
- Элементийн түлхүүрийг эрэмбэ болгон ашиглах
- Эрэмбэлэх элементүүдийг эрэмбэтэй дараалалд хийх
- Эрэмбийн дараалаар элементийг гаргаж авах
 - Хэрвээ **min** эрэмбэтэй дараалал бол, элементүүдийг эрэмбийн(түлхүүрийн) өсөх дарааллаар гаргана
 - Хэрвээ **max** эрэмбэтэй дараалал бол, элементүүдийг эрэмбийн(түлхүүрийн)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

Эрэмбэлэлтийн жишээ

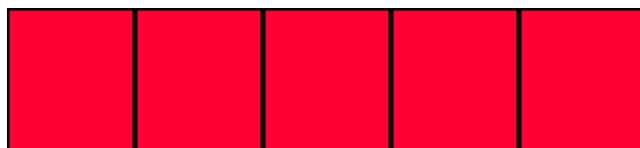
6, 8, 2, 4, 1 гэсэн түлхүүртэй таван элементийг тах эрэмбэтэй дарааллын аргаар эрэмбэлэе.

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

Мах эрэмбэтэй дараалалд хийсний дараа

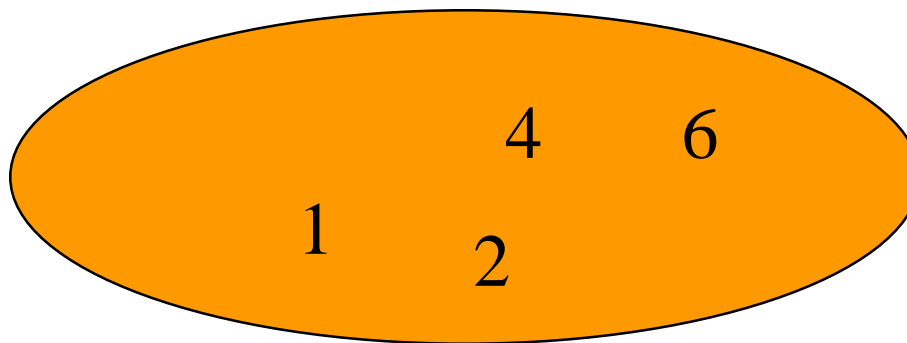


Мах
эрэмбэтэй
дараалал



Эрэмбэлэгдсэн массив

Эхний Remove Max үйлдлийн дараа

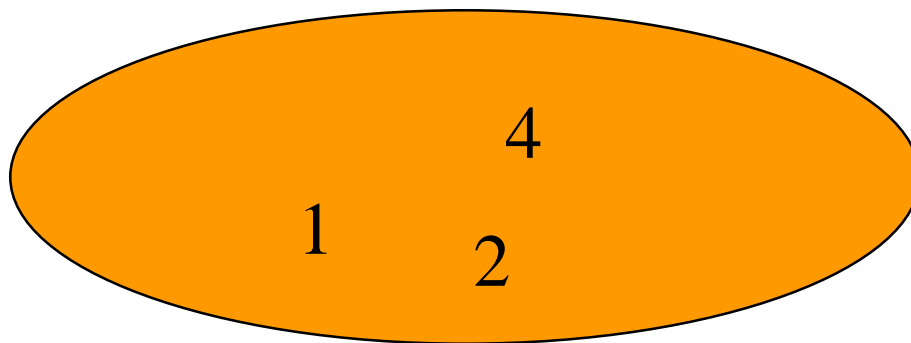


Max
эрэмбэтэй
дараалал

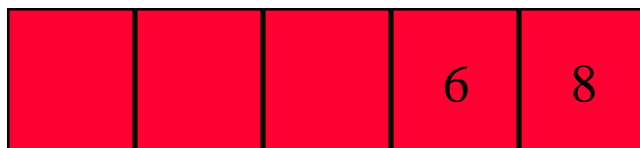


Эрэмбэлэгдсэн массив

Хоёр дахь Remove Max үйлдлийн дараа

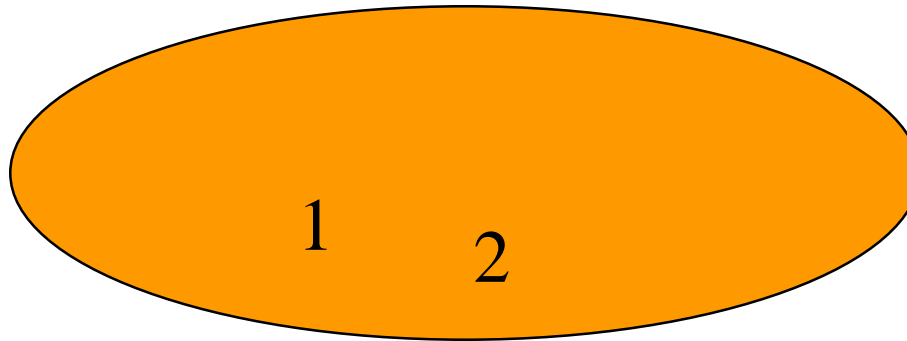


Max
эрэмбэтэй
дараалал

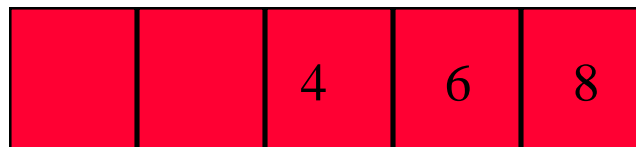


Эрэмбэлэгдсэн массив

Гурав дахь Remove Max үйлдлийн дараа

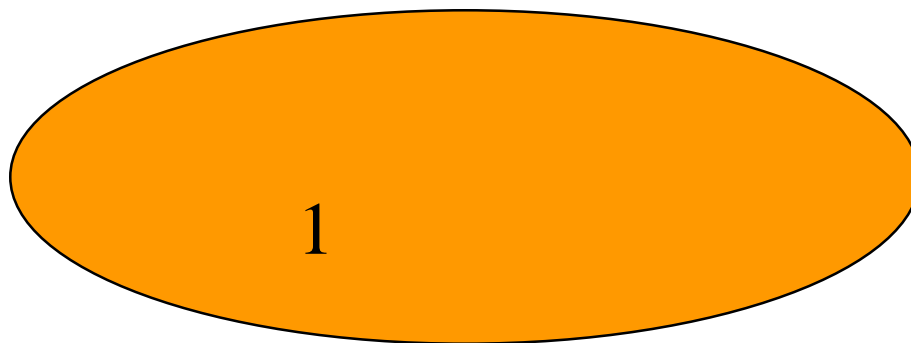


Max
эрэмбэтэй
дараалал

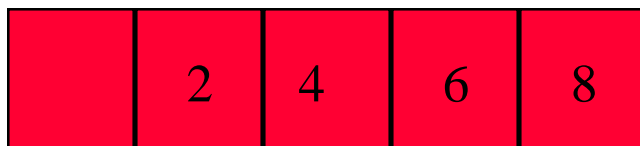


Эрэмбэлэгдсэн массив

Дөрөв дэх Remove Max үйлдлийн дараа

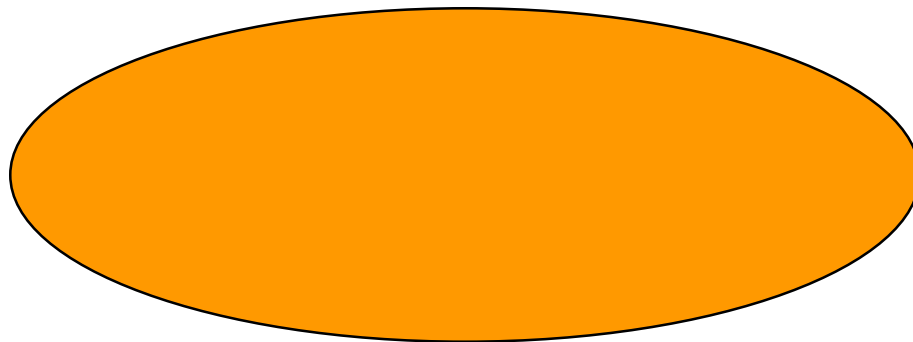


Мах
эрэмбэтэй
дараалал



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

Овоог хэрэгжүүлэхдээ тах эрэмбэтэй дарааллыг ашиглана.

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

Машины төлөвлөлт

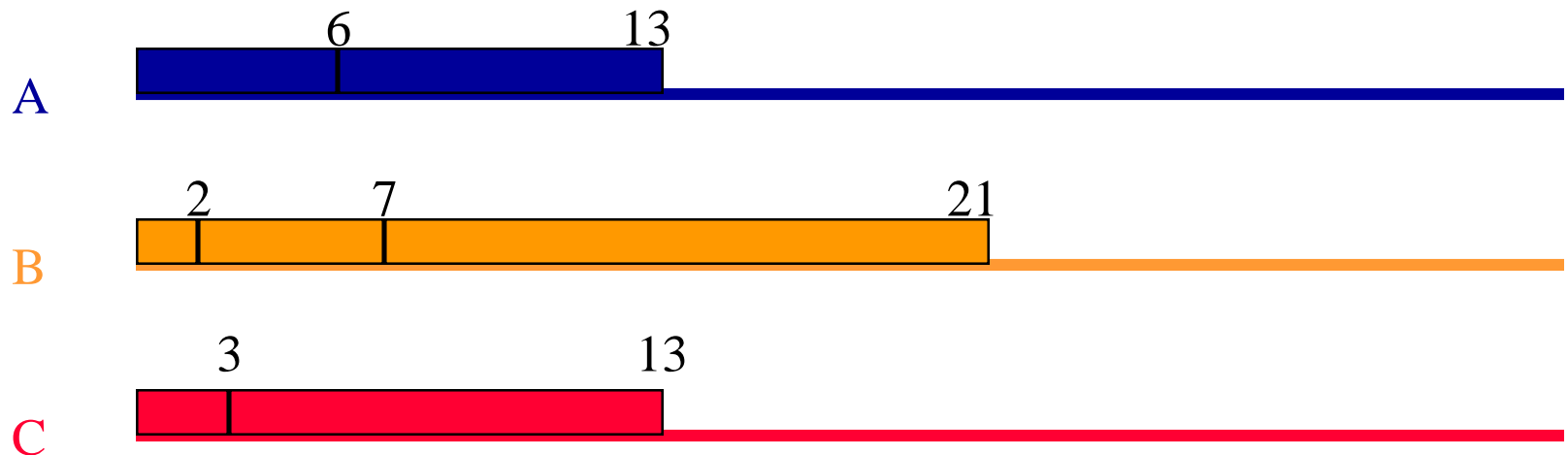
- m ижил төрлийн машин (өрөм, зүсэгч, г.м.)
- n ажил гүйцэтгэх хэрэгтэй
- Машиныудыг ажлаар ачаалахдаа сүүлийн ажлыг гүйцэтгэх хугацааг минимум болгох

Машины төлөвлөлтийн жишээ

3 машин, 7 ажил

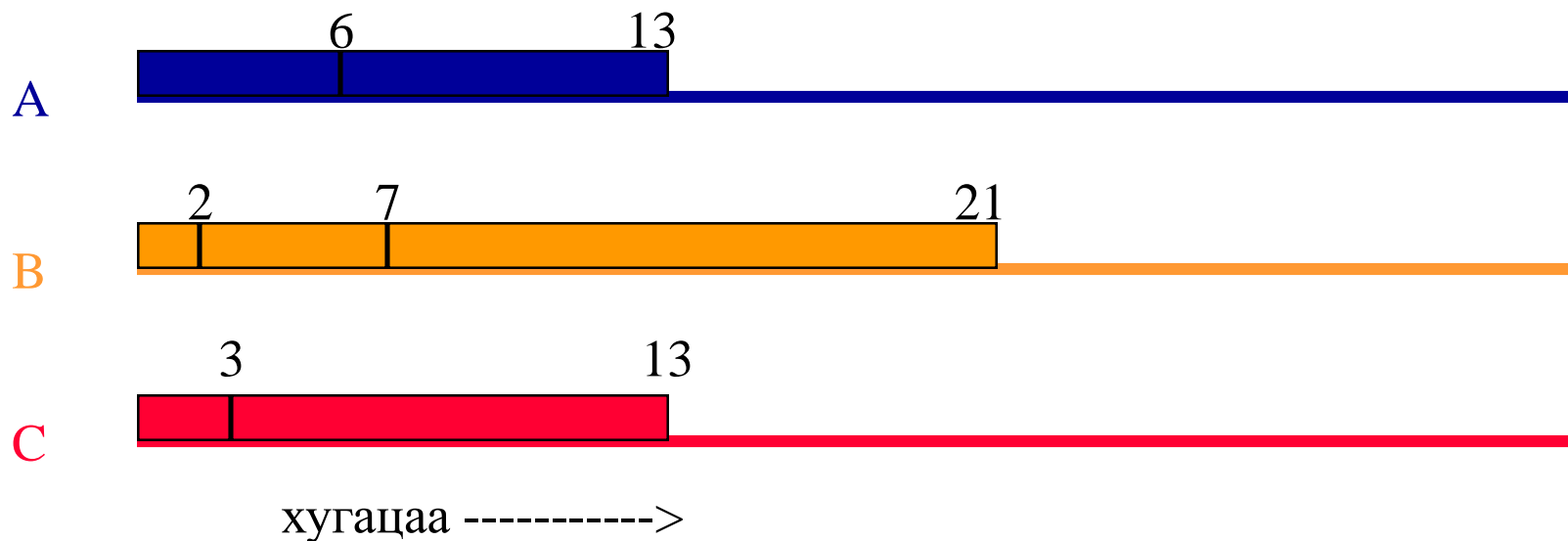
Ажлын хугацаа [6, 2, 3, 5, 10, 7, 14]

Боломжит төлөвлөлт:



хугацаа ----->

Машины төлөвлөлтийн жишээ



Дуусах хугацаа = 21

Зорилго: Дуусах хугацааг минимум болгох.

LPT Төлөвлөлт

Longest Processing Time - эхлээд.

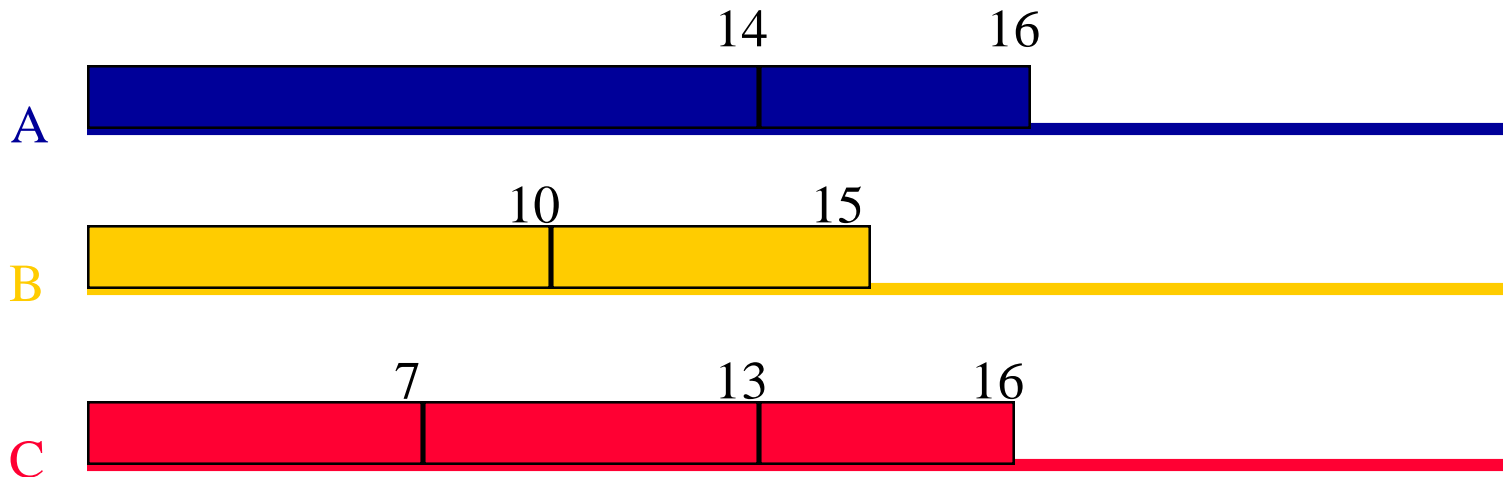
Ажлууд дараах байдлаар эрэмбэлэгджээ

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)

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

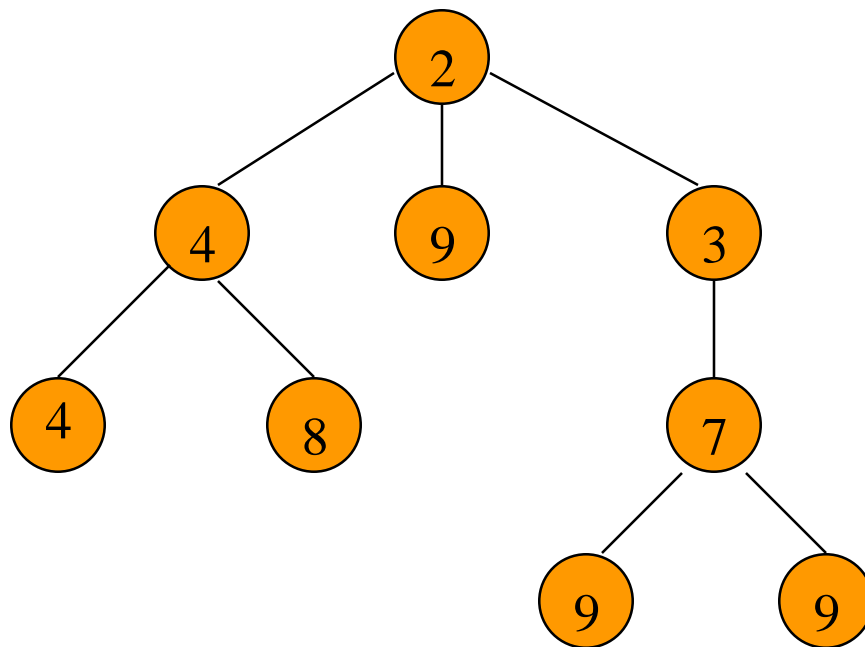
Min модны тодорхойлолт

Модны зангилаа бүр утгатай.

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

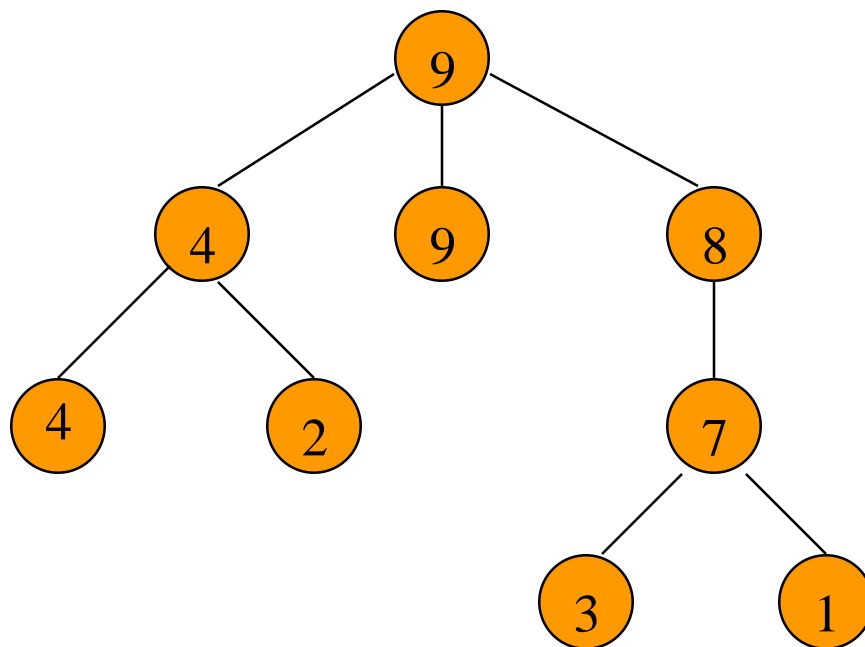
Өөрөөр хэлбэл бага утгатай хойч байхгүй.

Min модны жишээ



Үндэс минимум элементтэй.

Мах модны жишээ

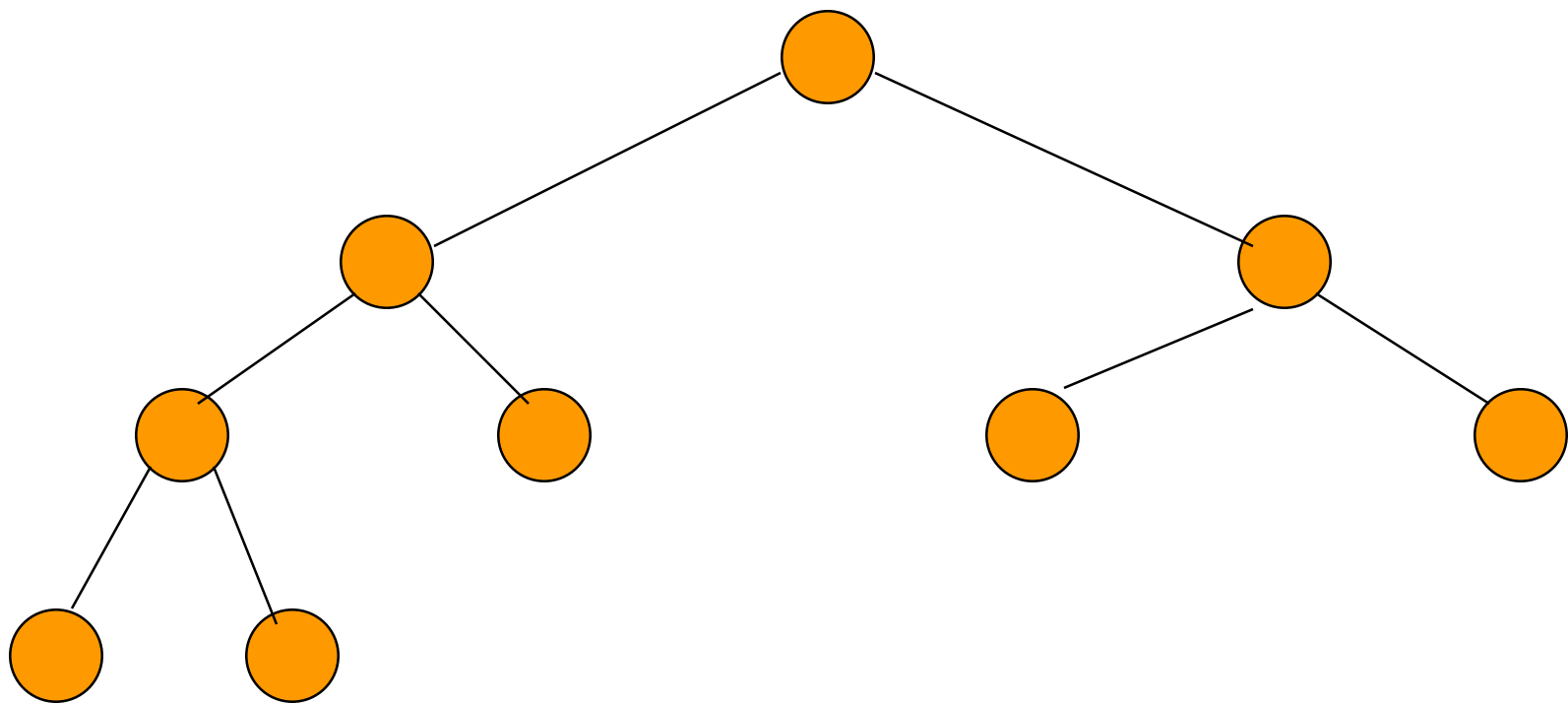


Үндэс максимум элементтэй.

Min овоолгын тодорхойлолт

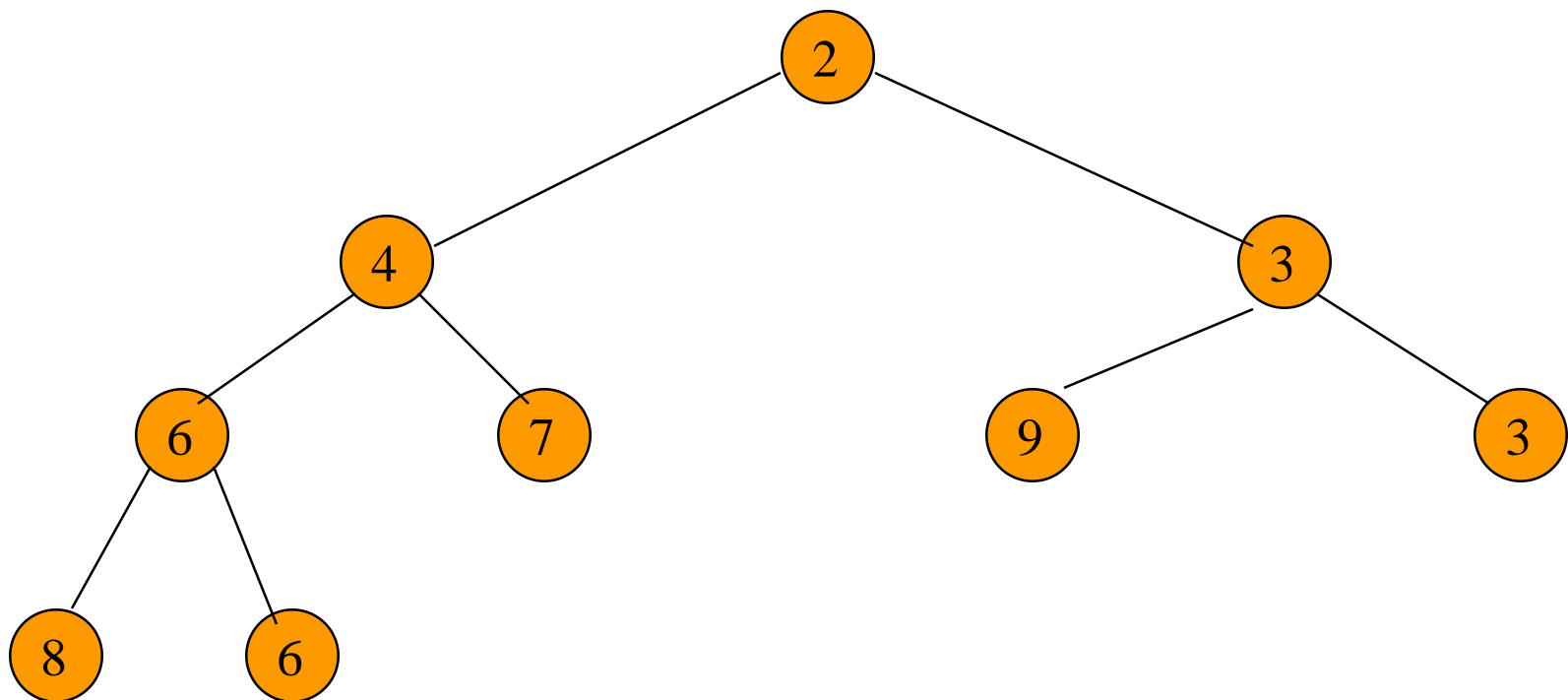
- Төгс хоёртын мод
- min мод

9 зангилаатай Min овоолго



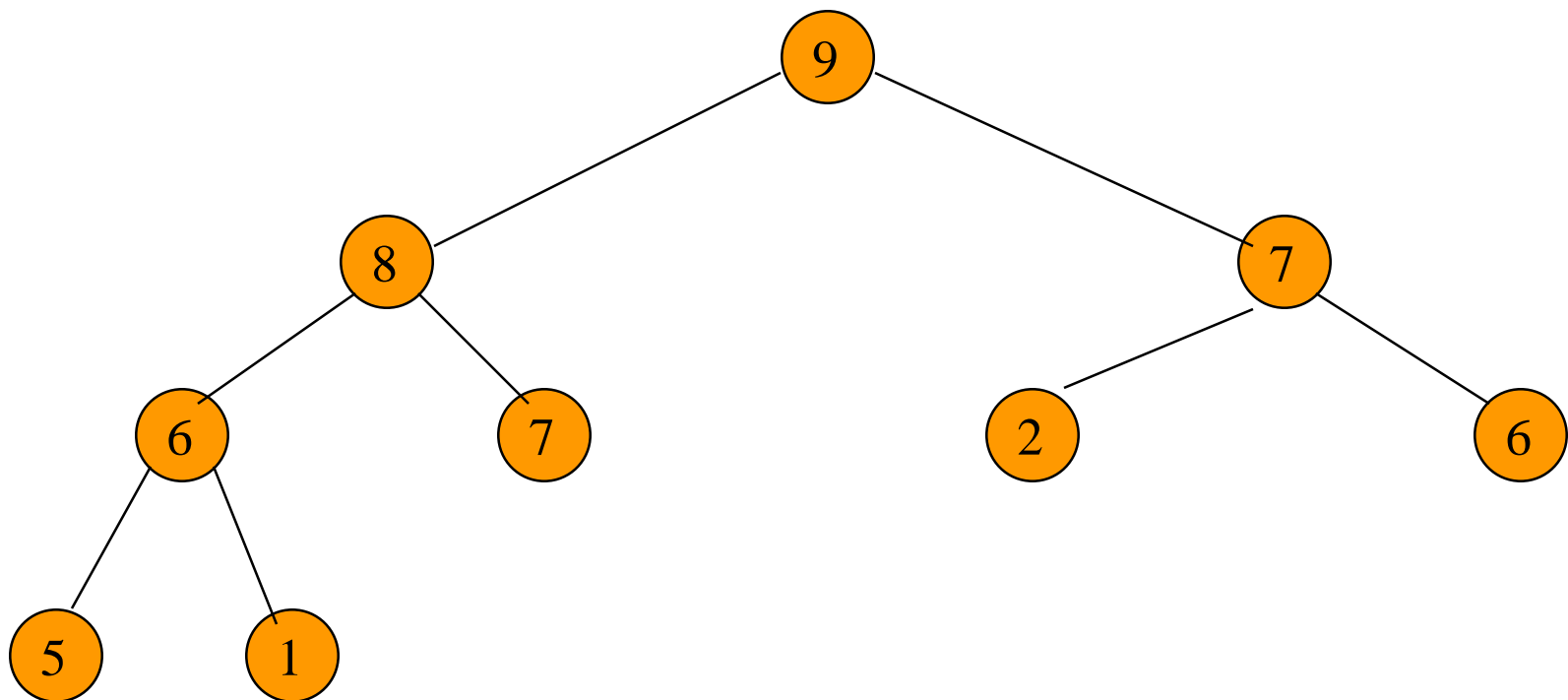
9 зангилаатай төгс хоёртын мод.

9 зангилаатай Min овоолго



9 зангилаатай төгс хоёртын мод
мөн min мод.

9 зангилаатай Мах овоолго

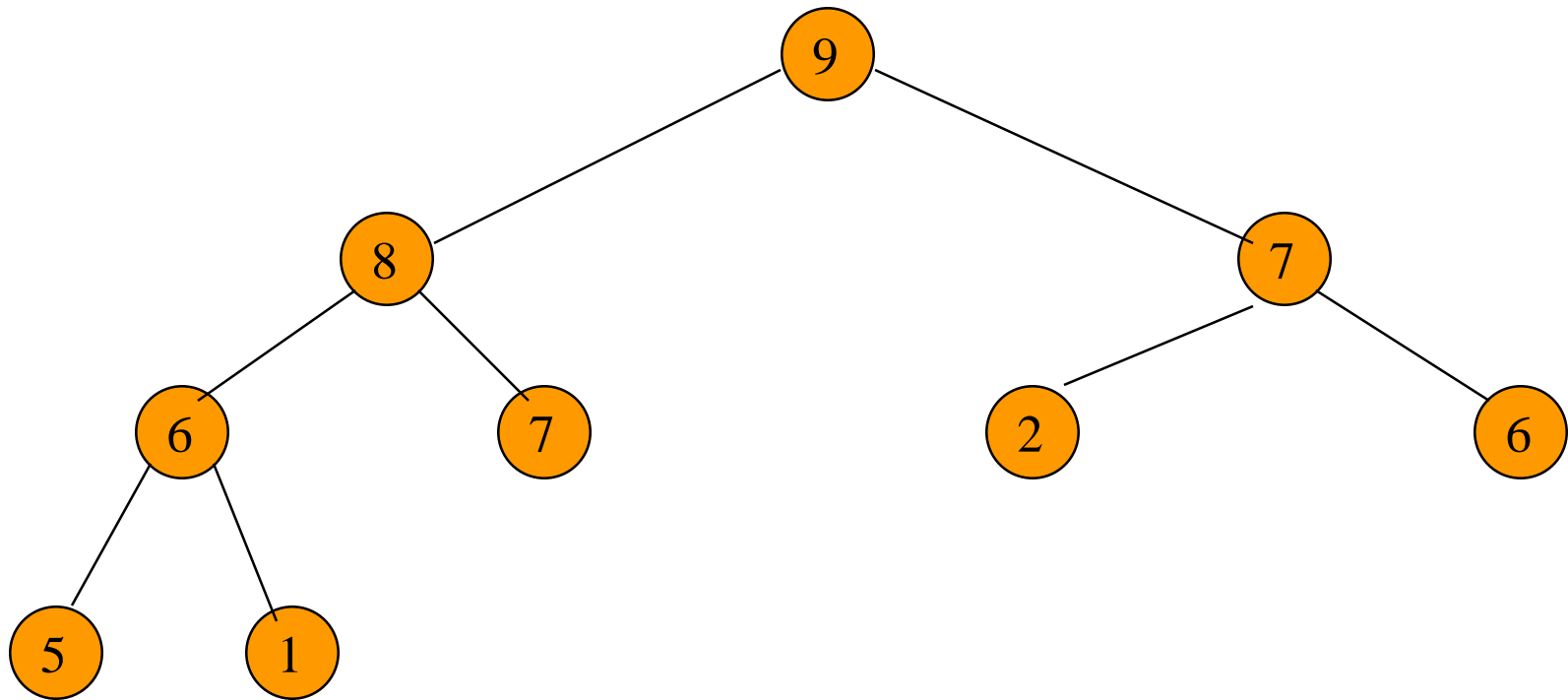


9 зангилаатай төгс хоёртын мод
мөн тах мод.

Овоолгын өндөр

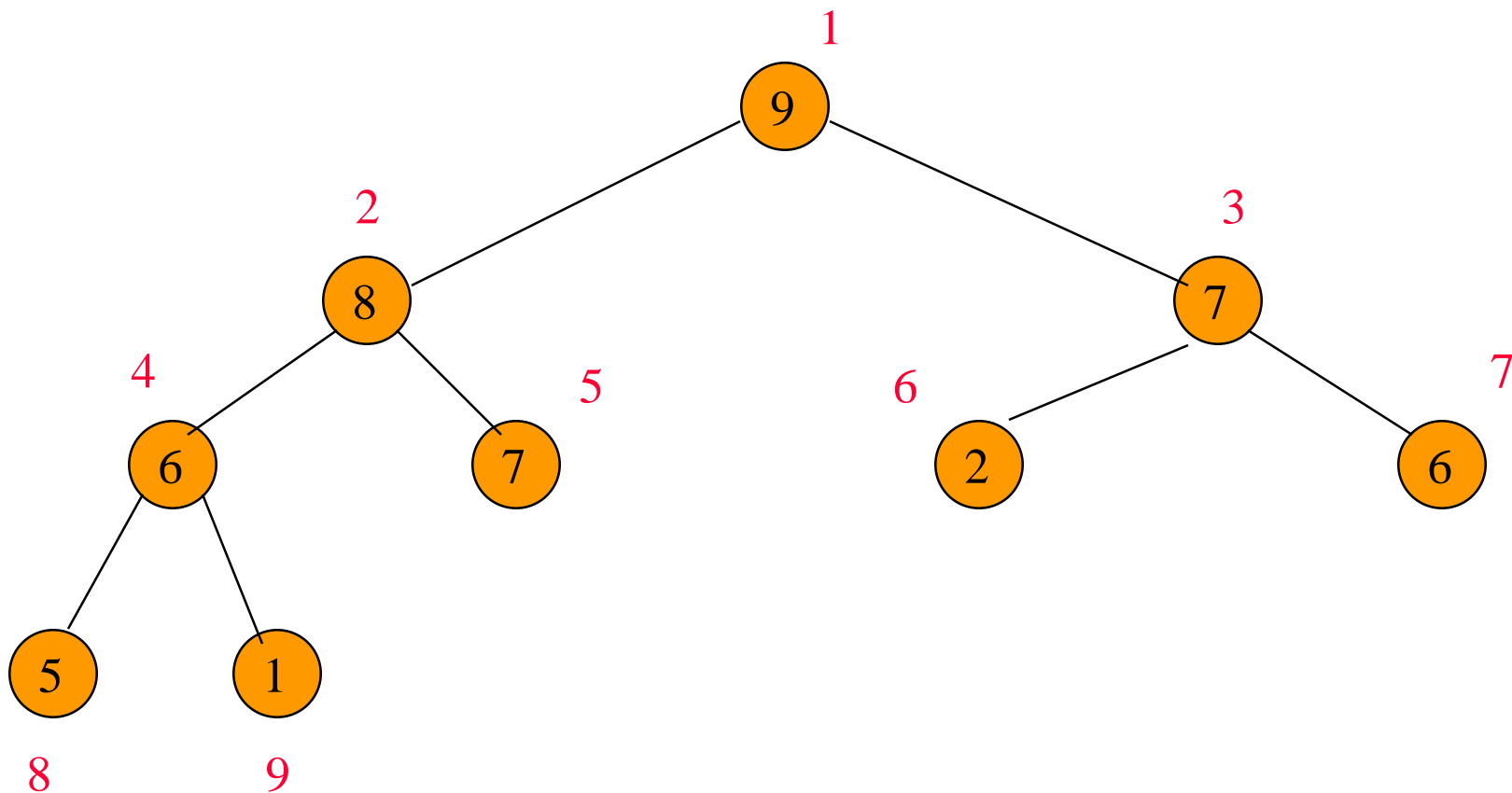
Нэгэнт овоолго нь төгс хоёртын
мод болохоор n зангилаатай
овоолгын өндөр $\log_2 (n+1)$.

Овоолгыг массиваар оновчтой дүрсэлж болно

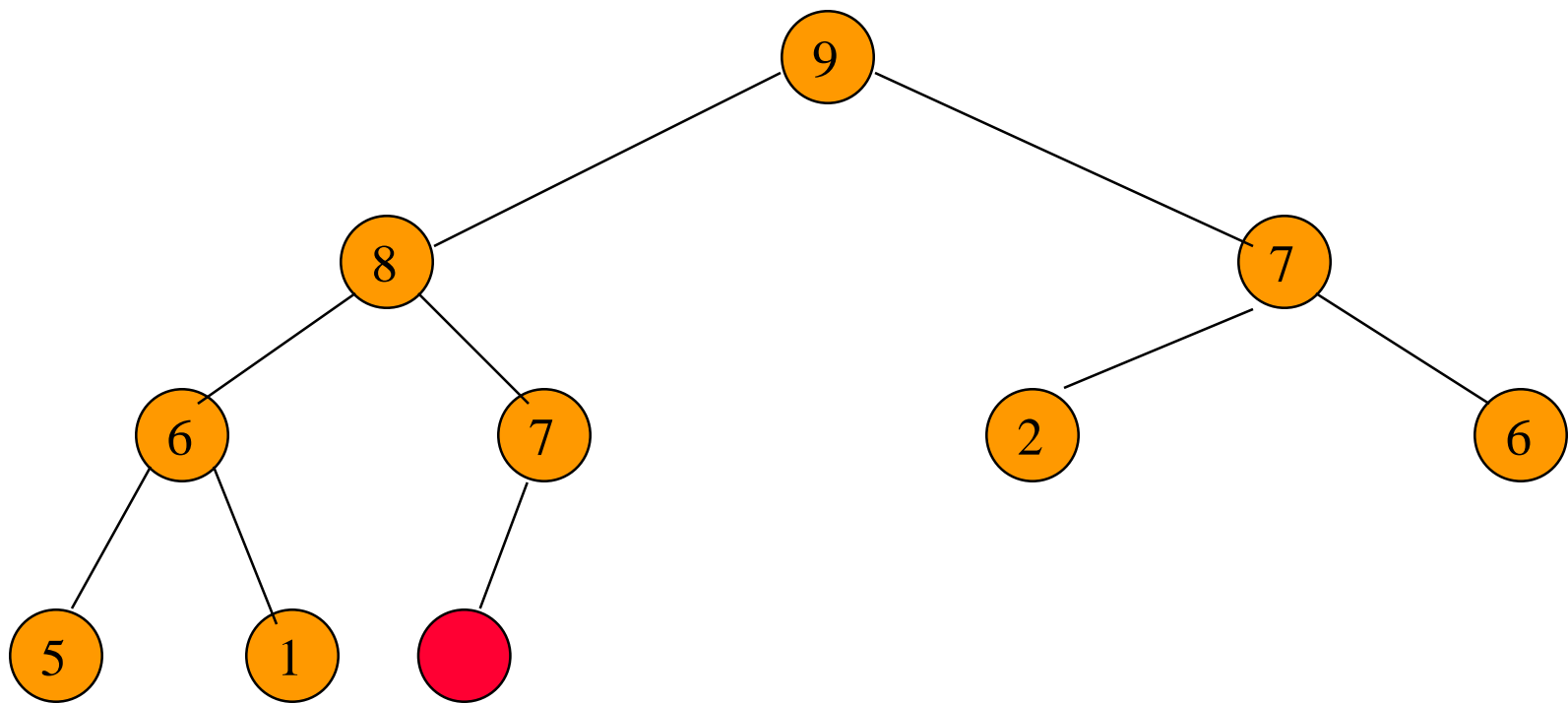


0 1 2 3 4 5 6 7 8 9 10

Овоолгоор дээш, доош явах

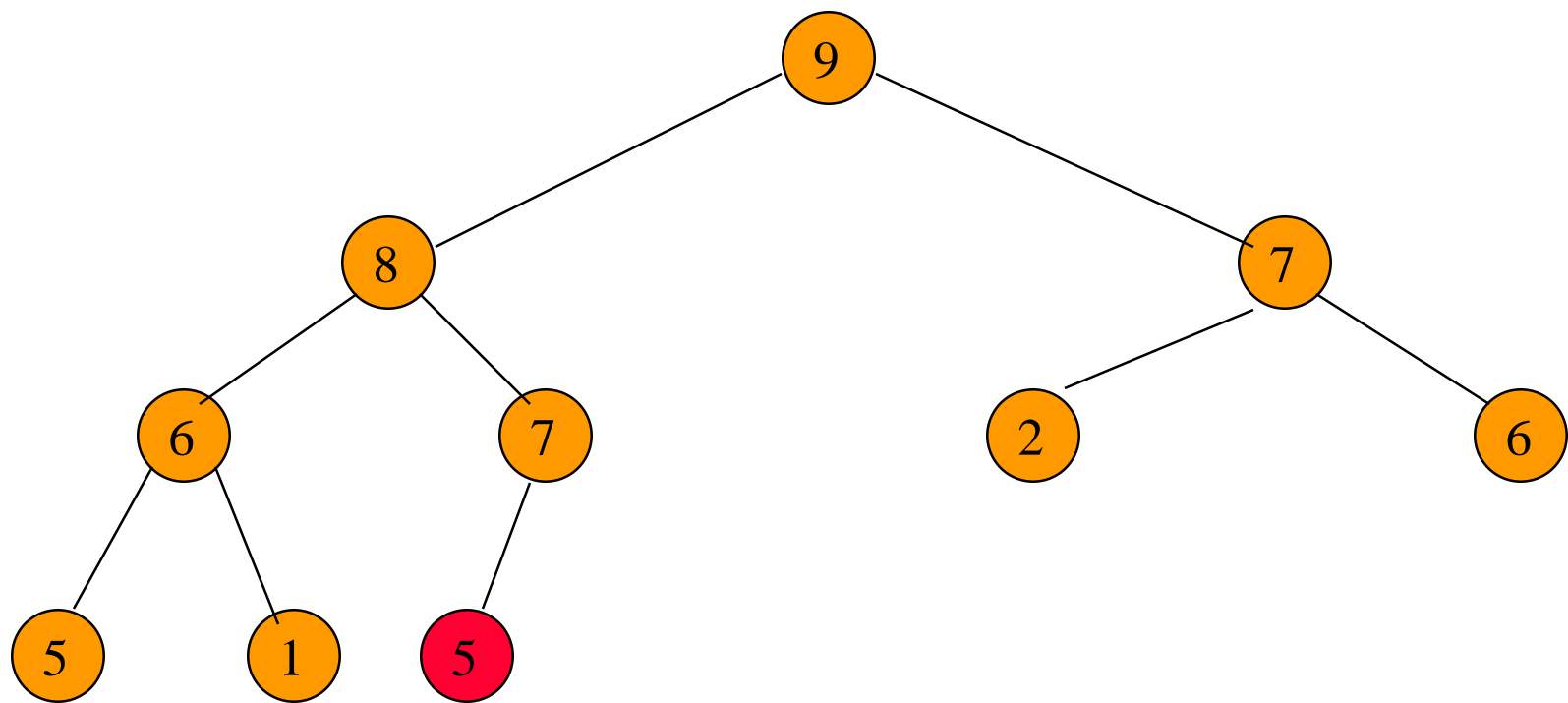


Мах овоолгод элемент хийх



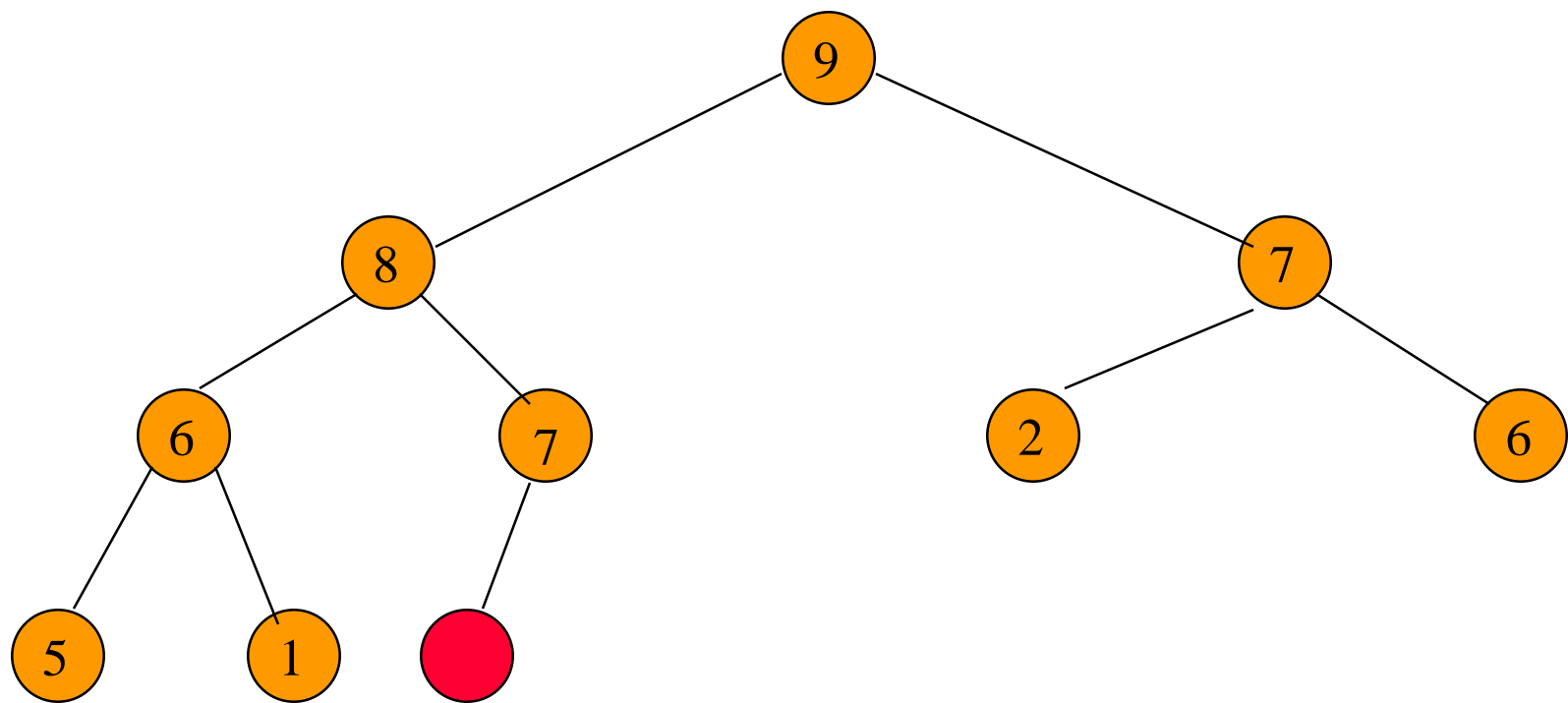
10 зангилаатай төгс хоёртын мод.

Мах овоолгод элемент хийх



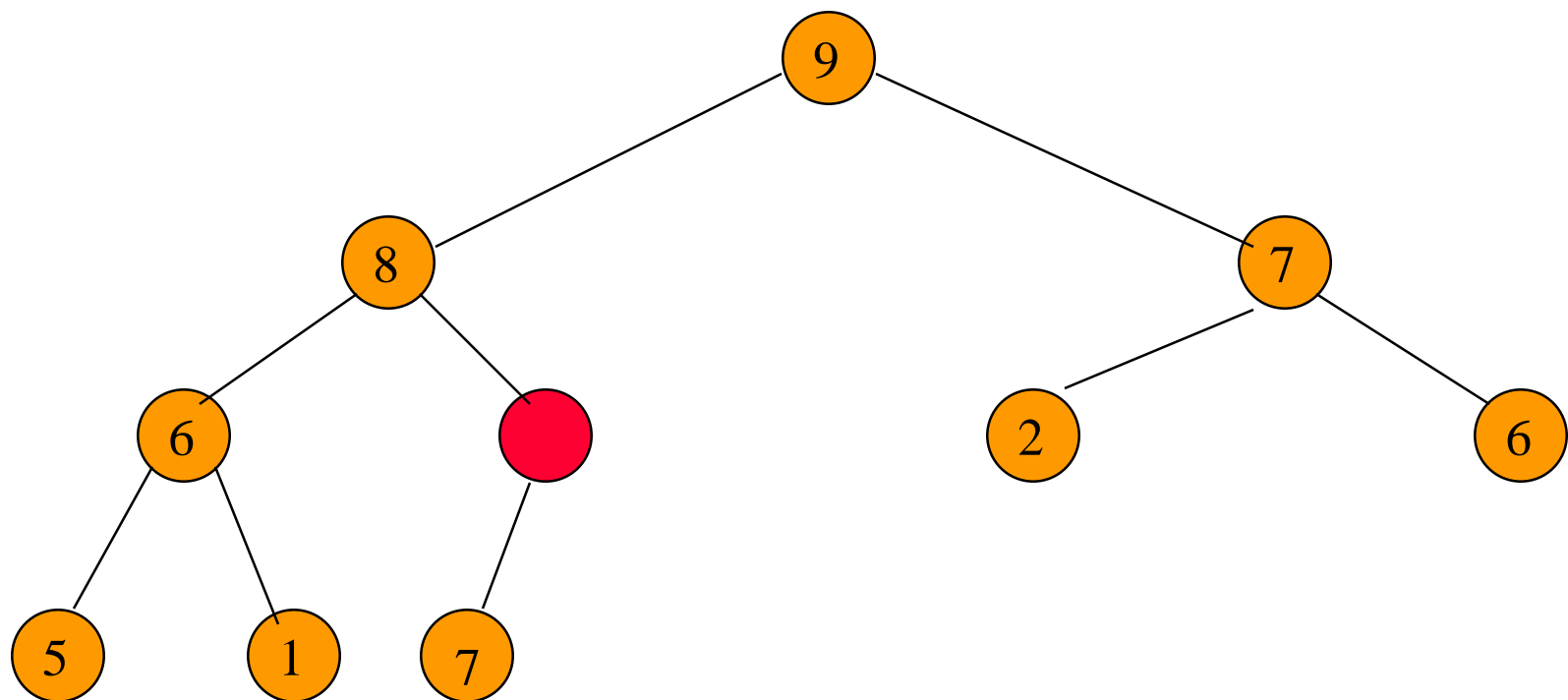
Шинэ элемент 5.

Мах овоолгод элемент хийх



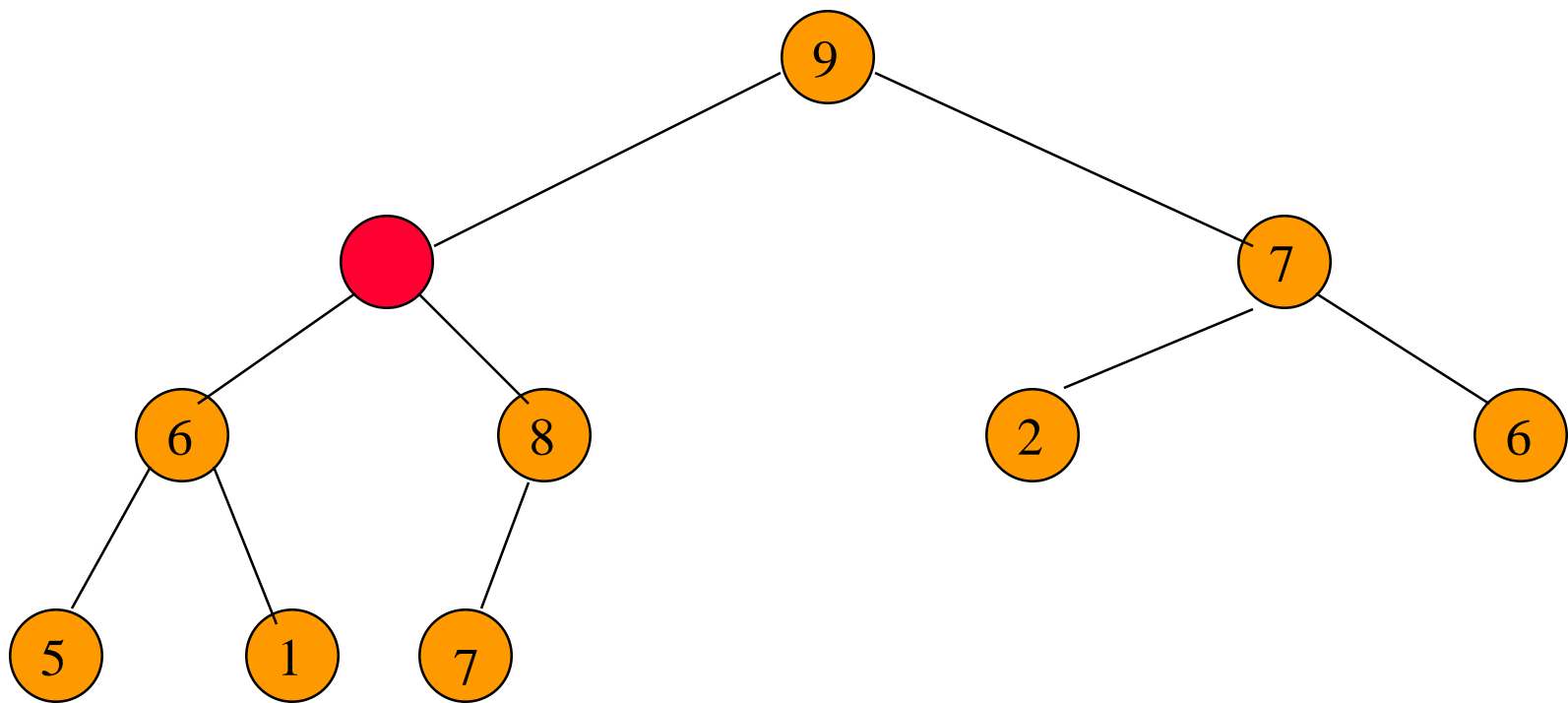
Шинэ элемент 20.

Мах овоолгод элемент хийх



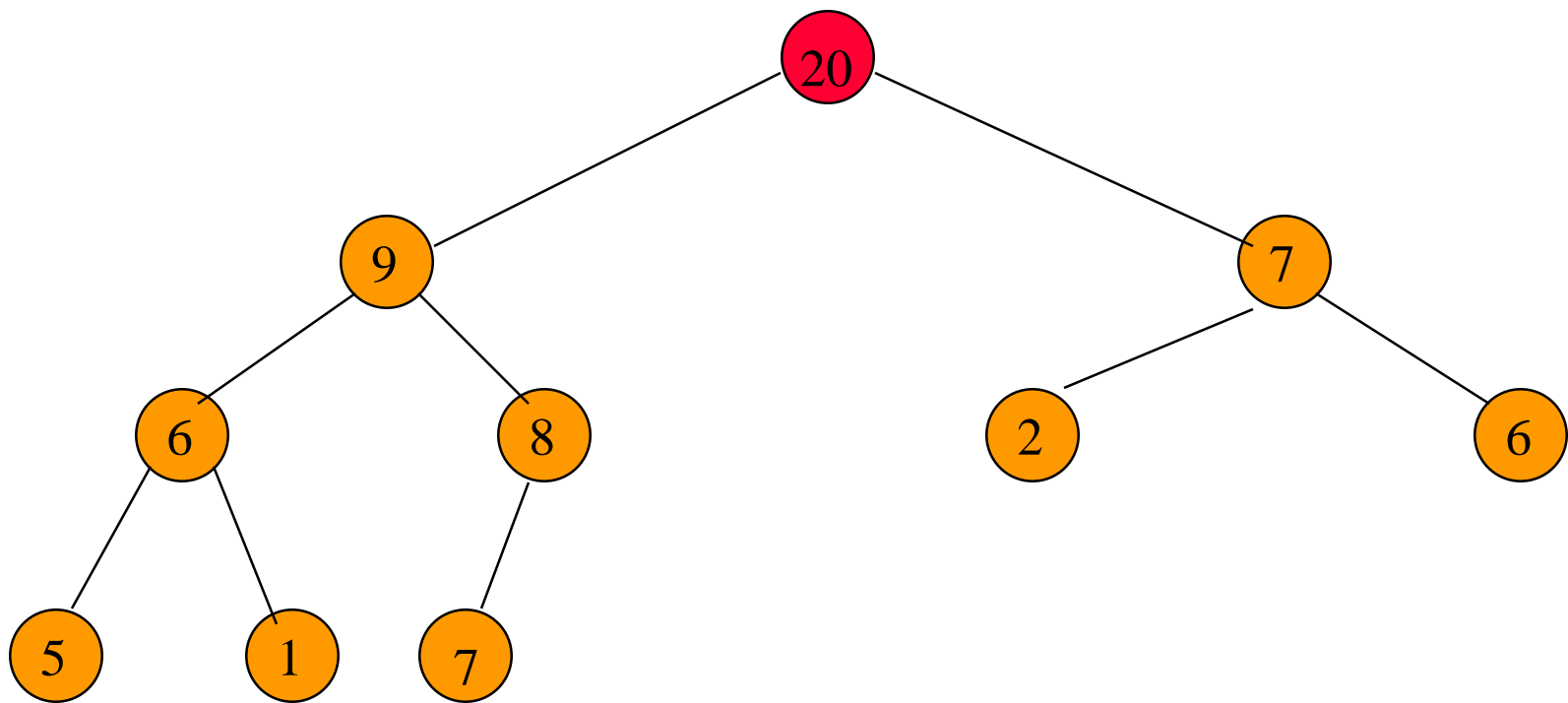
Шинэ элемент 20.

Мах овоолгод элемент хийх



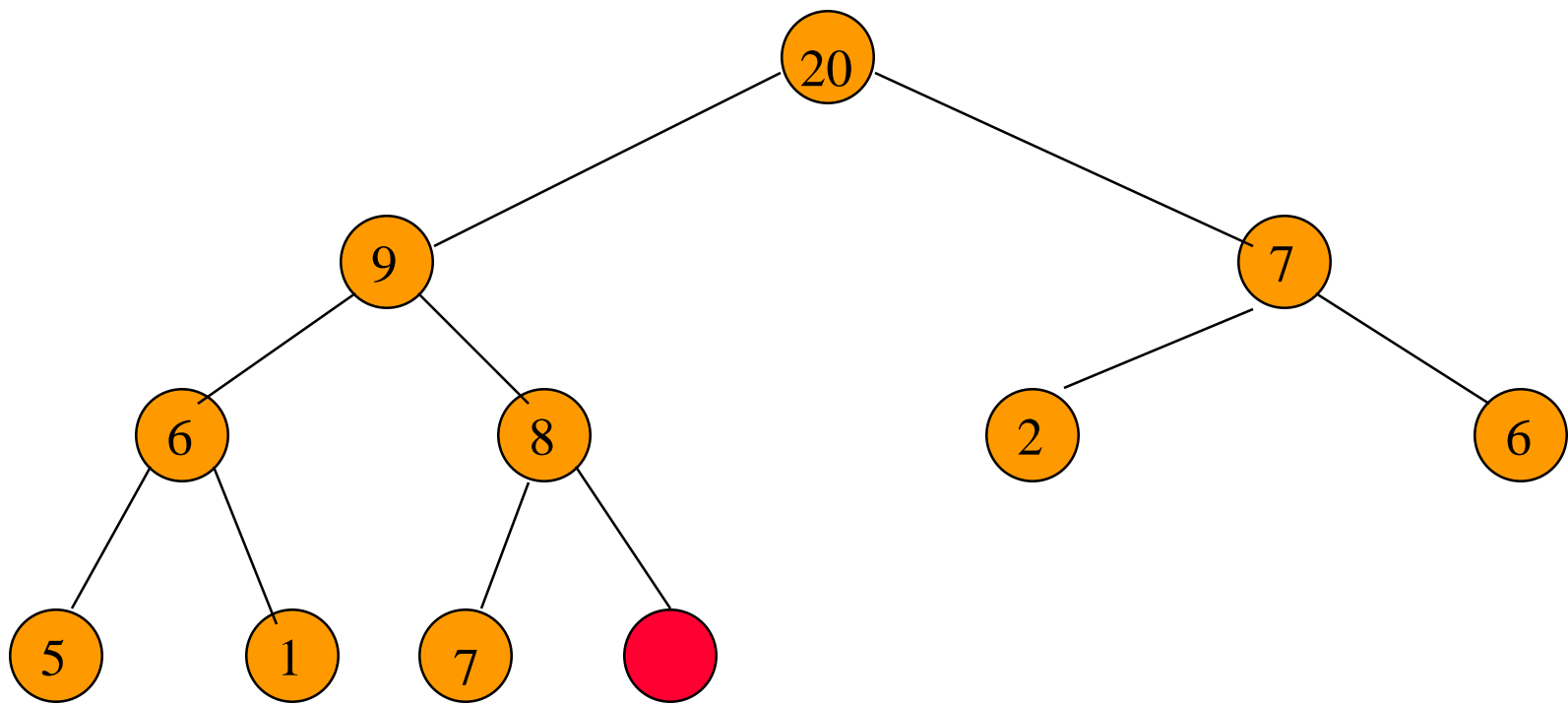
Шинэ элемент 20.

Мах овоолгод элемент хийх



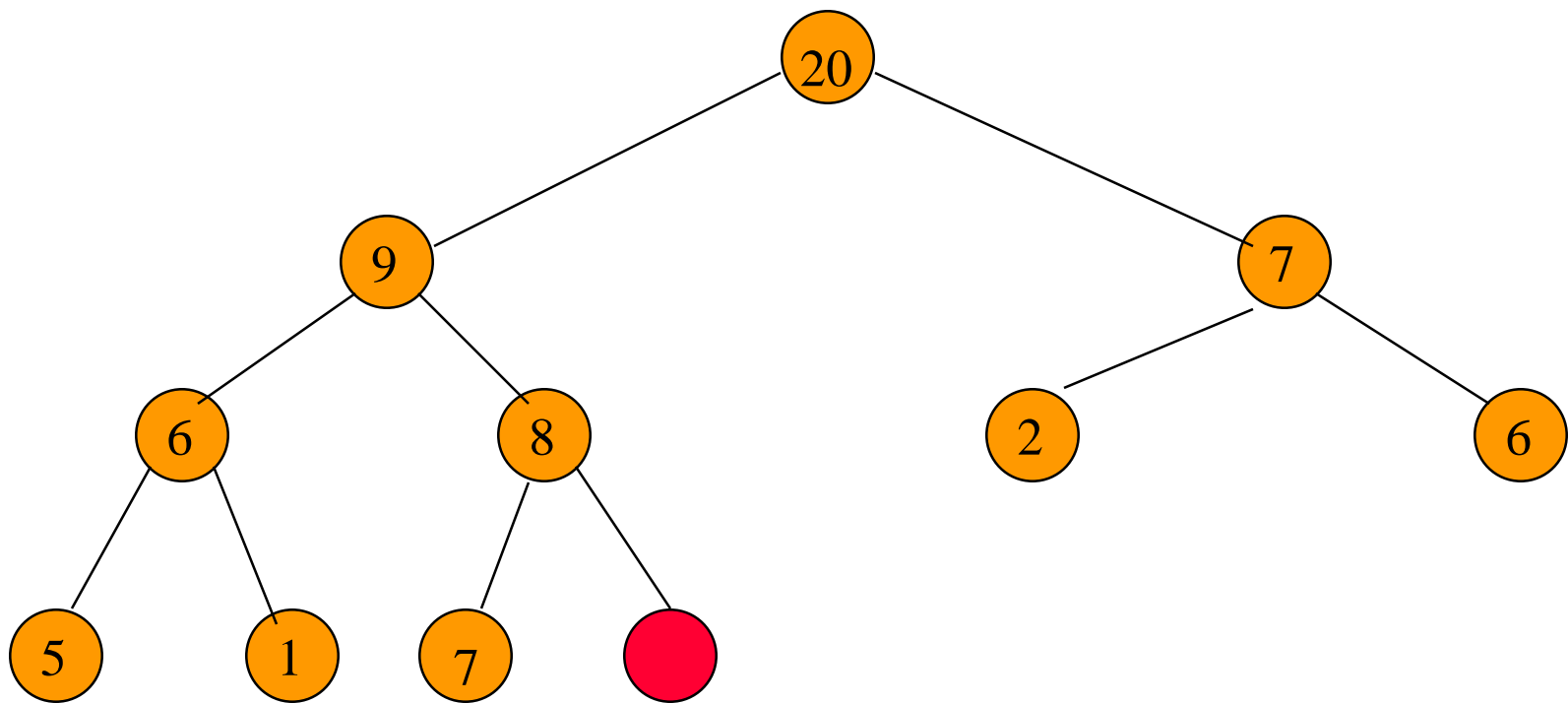
Шинэ элемент 20.

Мах овоолгод элемент хийх



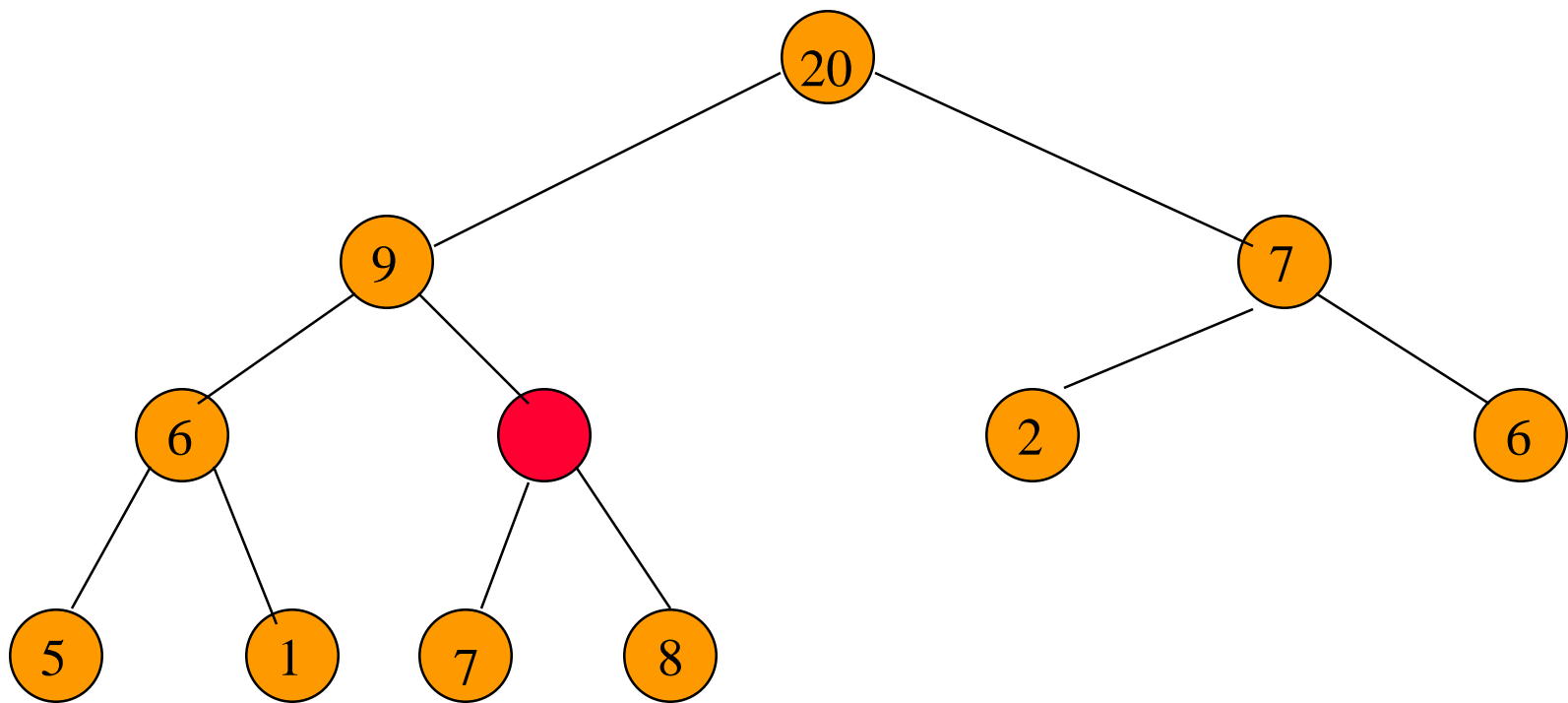
11 зангилаатай төгс хоёртын мод.

Мах овоолгод элемент хийх



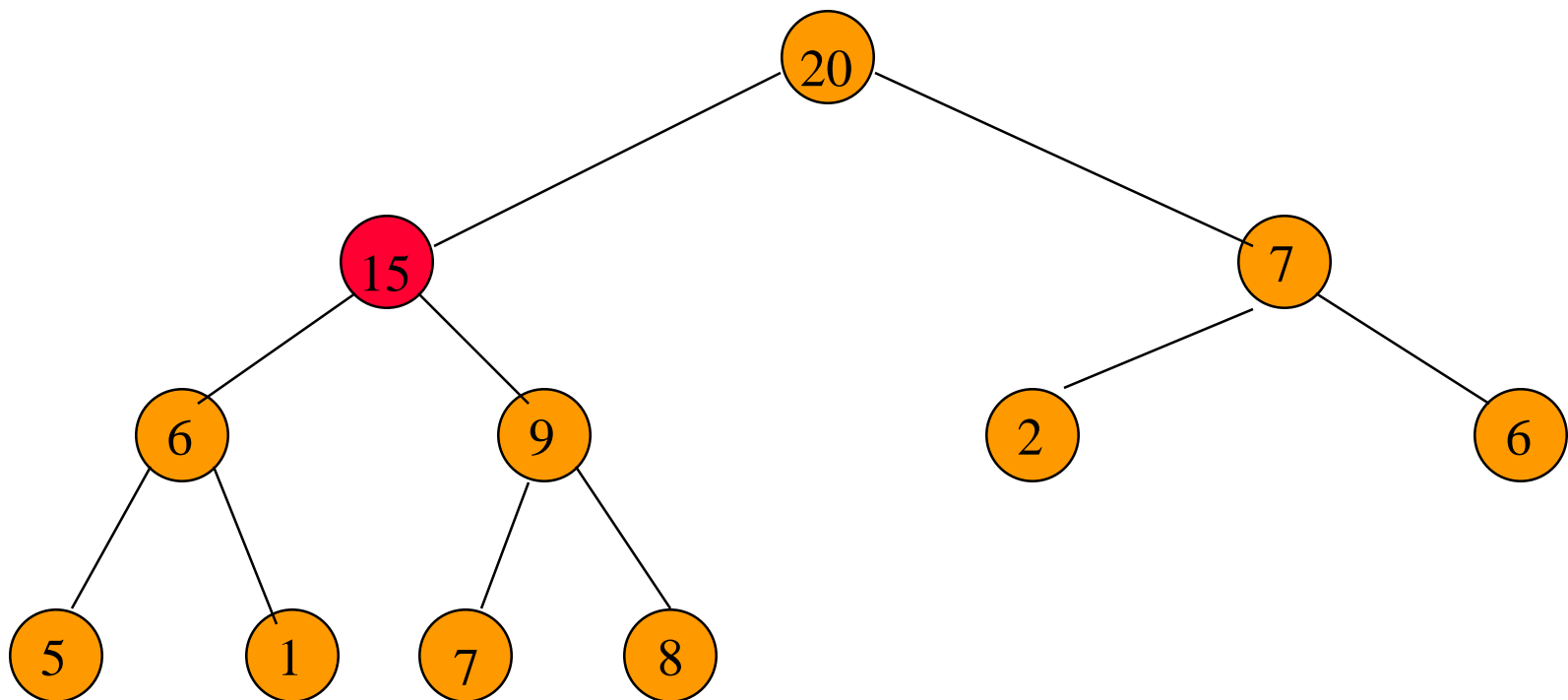
Шинэ элемент 15.

Мах овоолгод элемент хийх



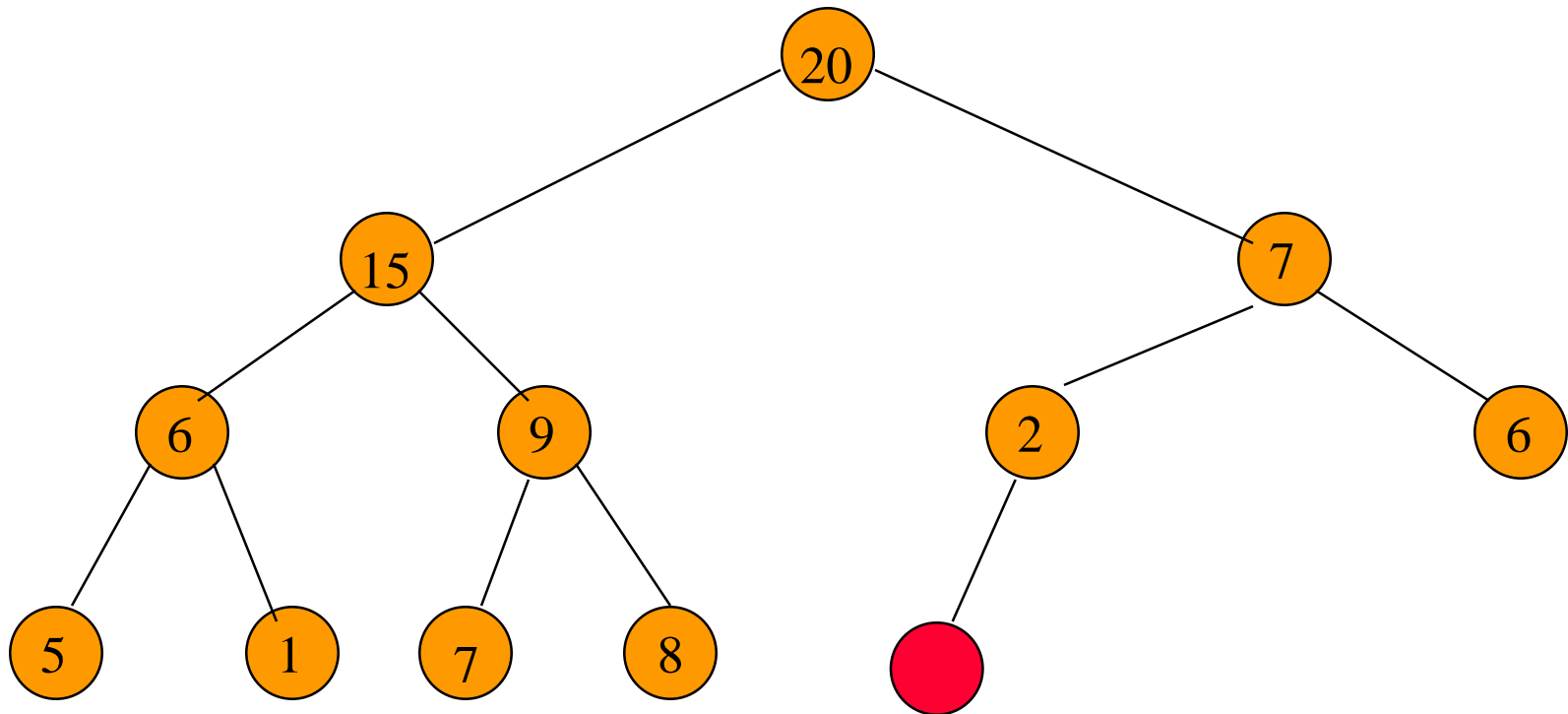
Шинэ элемент 15.

Мах овоолгод элемент хийх



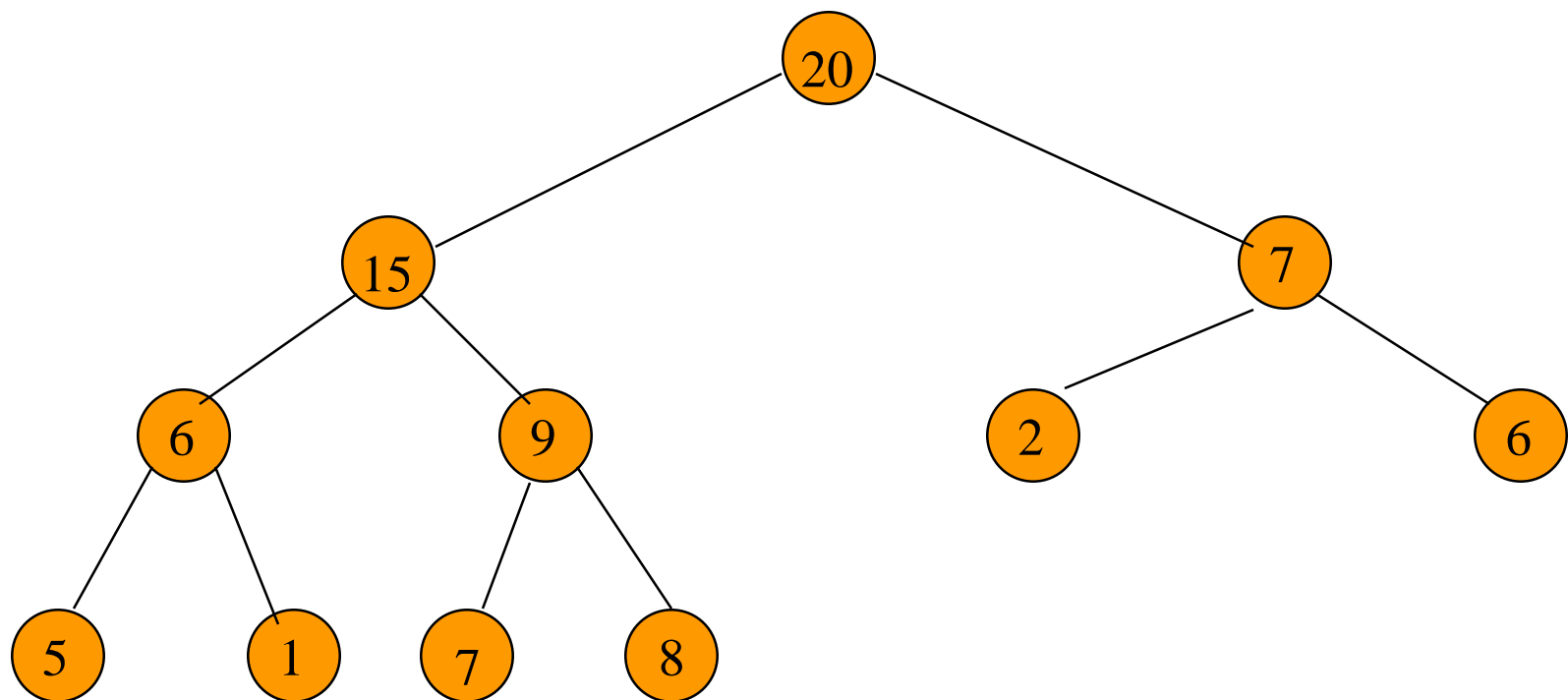
Шинэ элемент 15.

Put үйлдлийн хугацаа



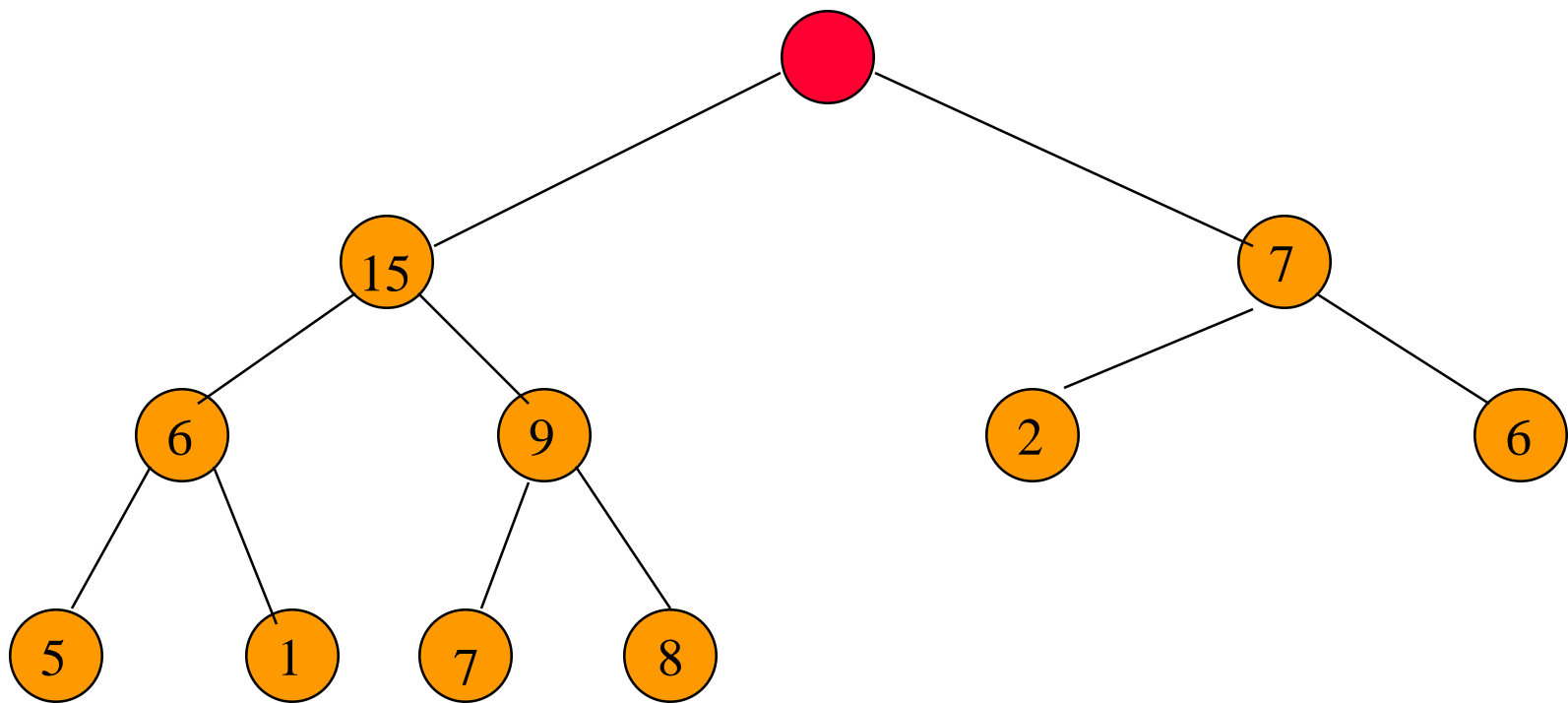
Хугацаа $O(\log n)$, үүнд n ОВООЛГЫН
ХЭМЖЭЭ.

Мах элементийн устгах



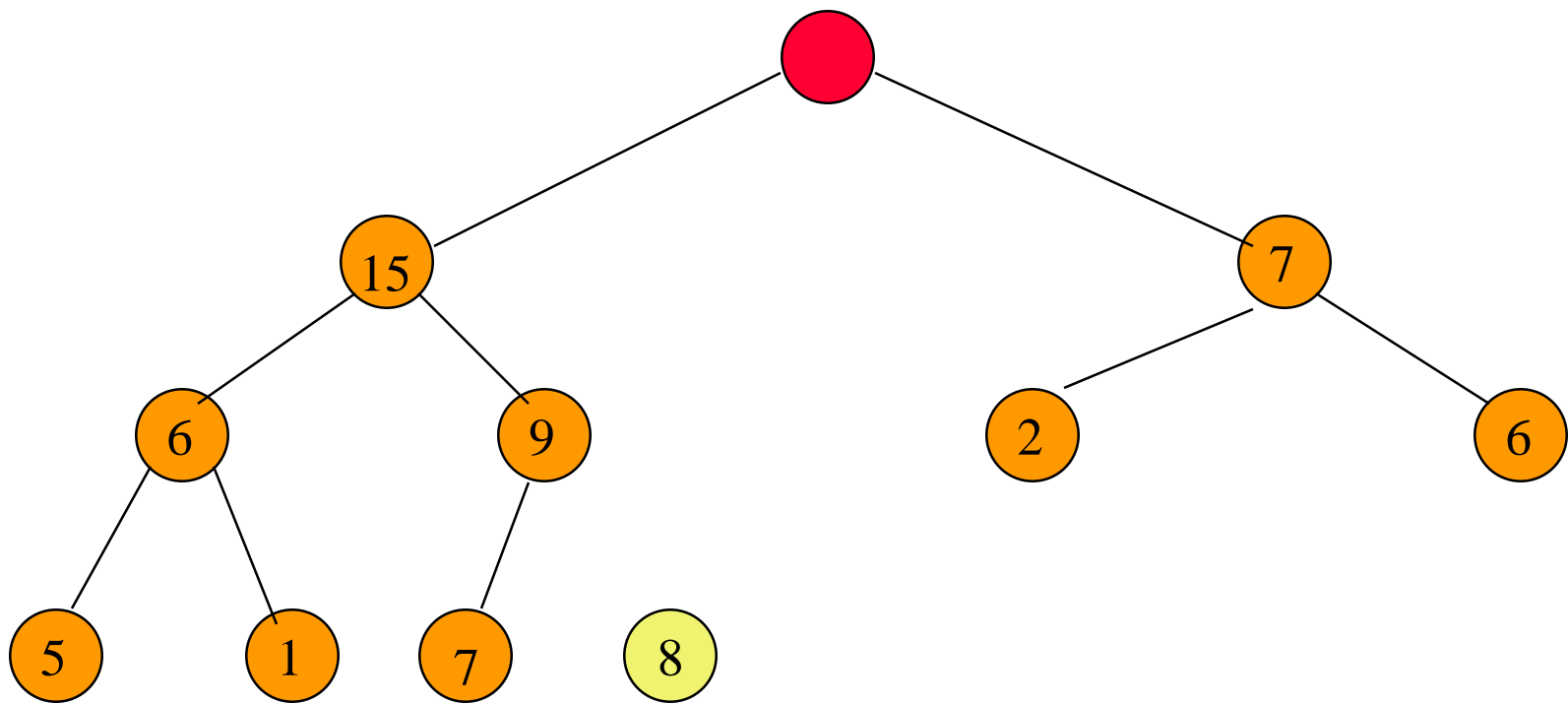
Мах элемент нь үндэс.

Мах элементийн устгах



тах элементийг устгасны дараа.

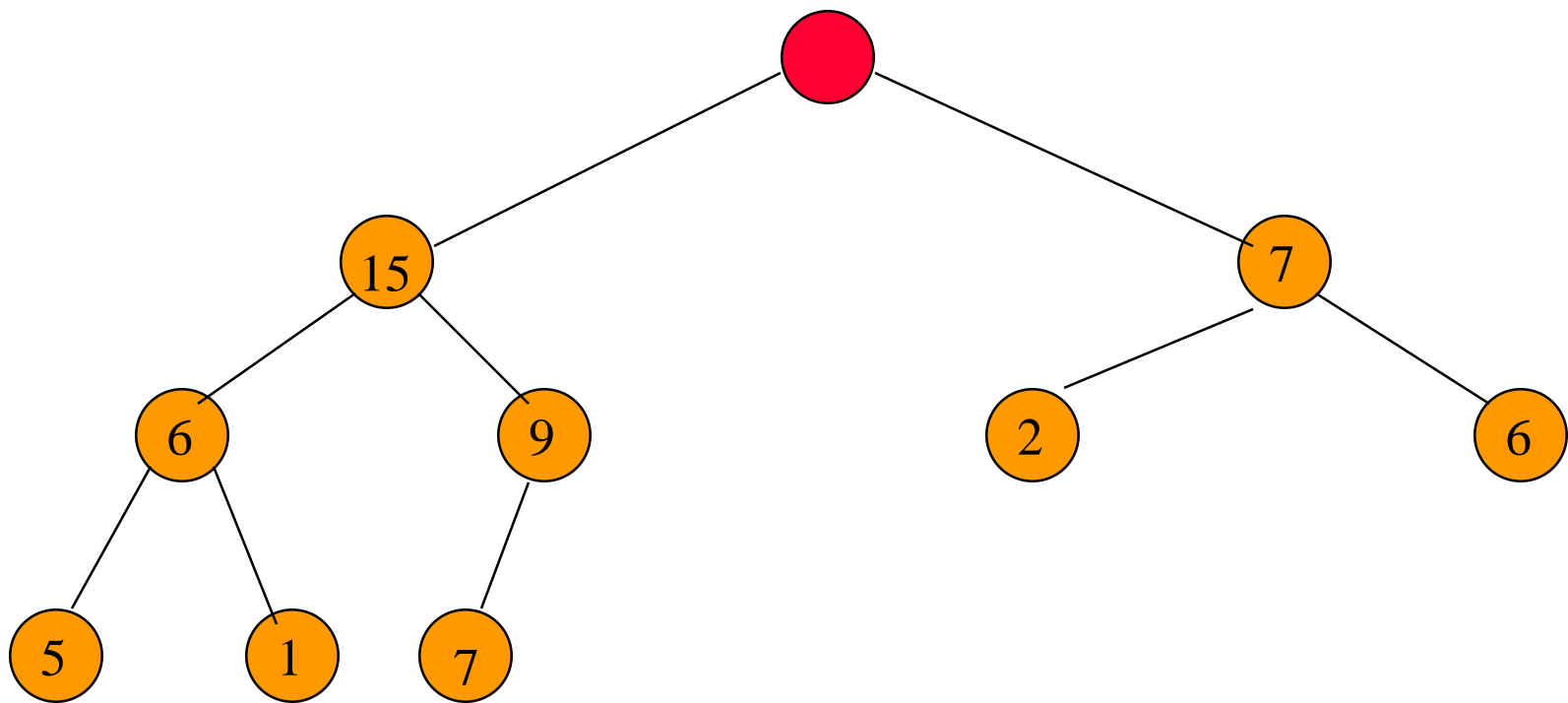
Мах элементийн устгах



10 зангилаатай овоолго.

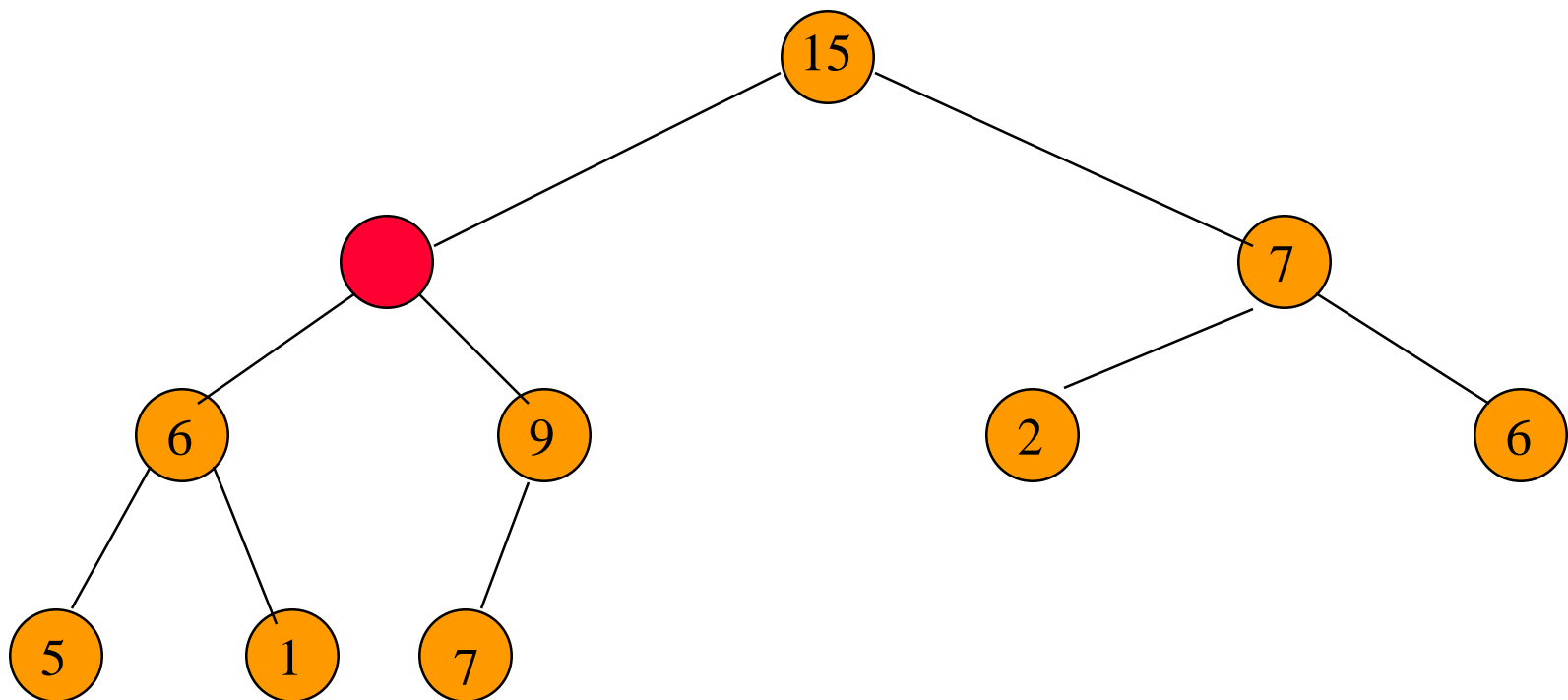
Овоолгод 8 –г дахин хийх.

Мах элементийн устгах



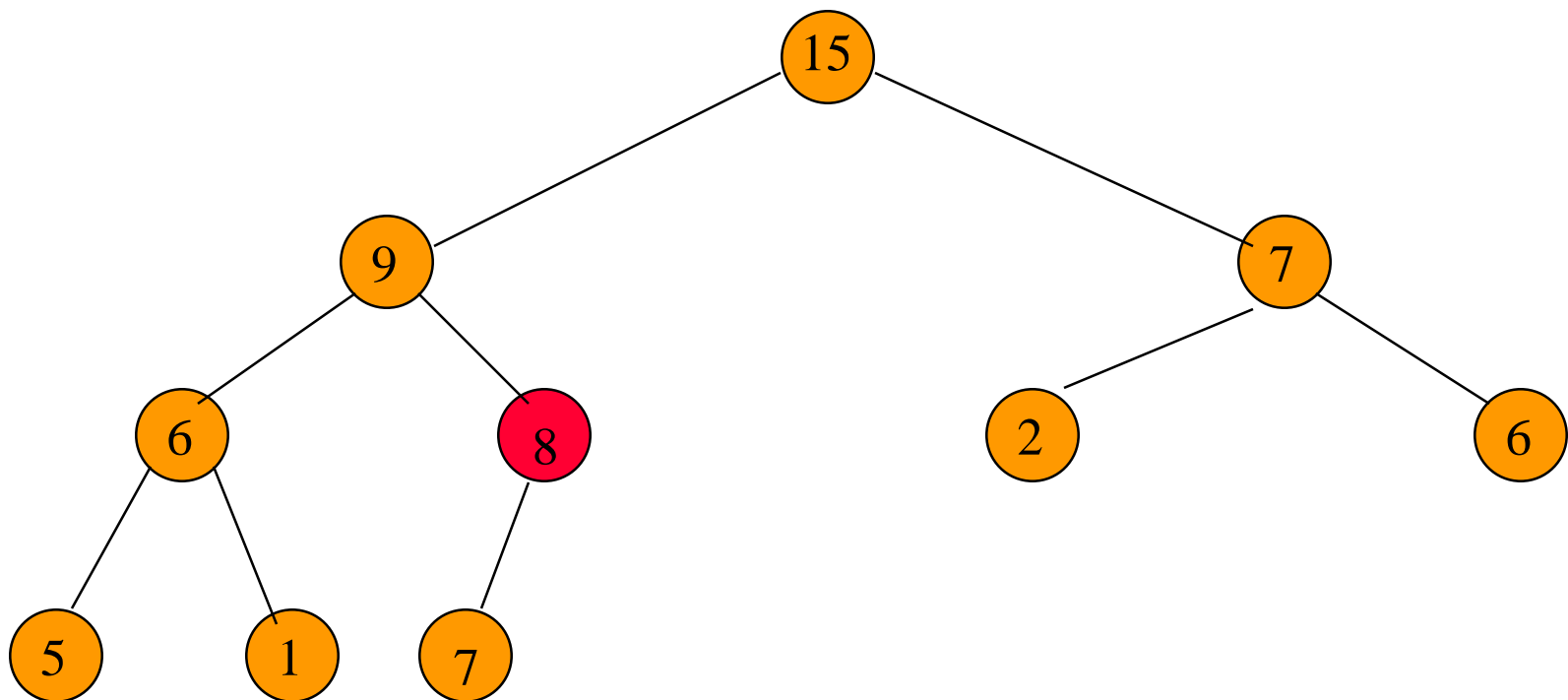
Овоолгод 8 –г дахин хийх.

Мах элементийн устгах



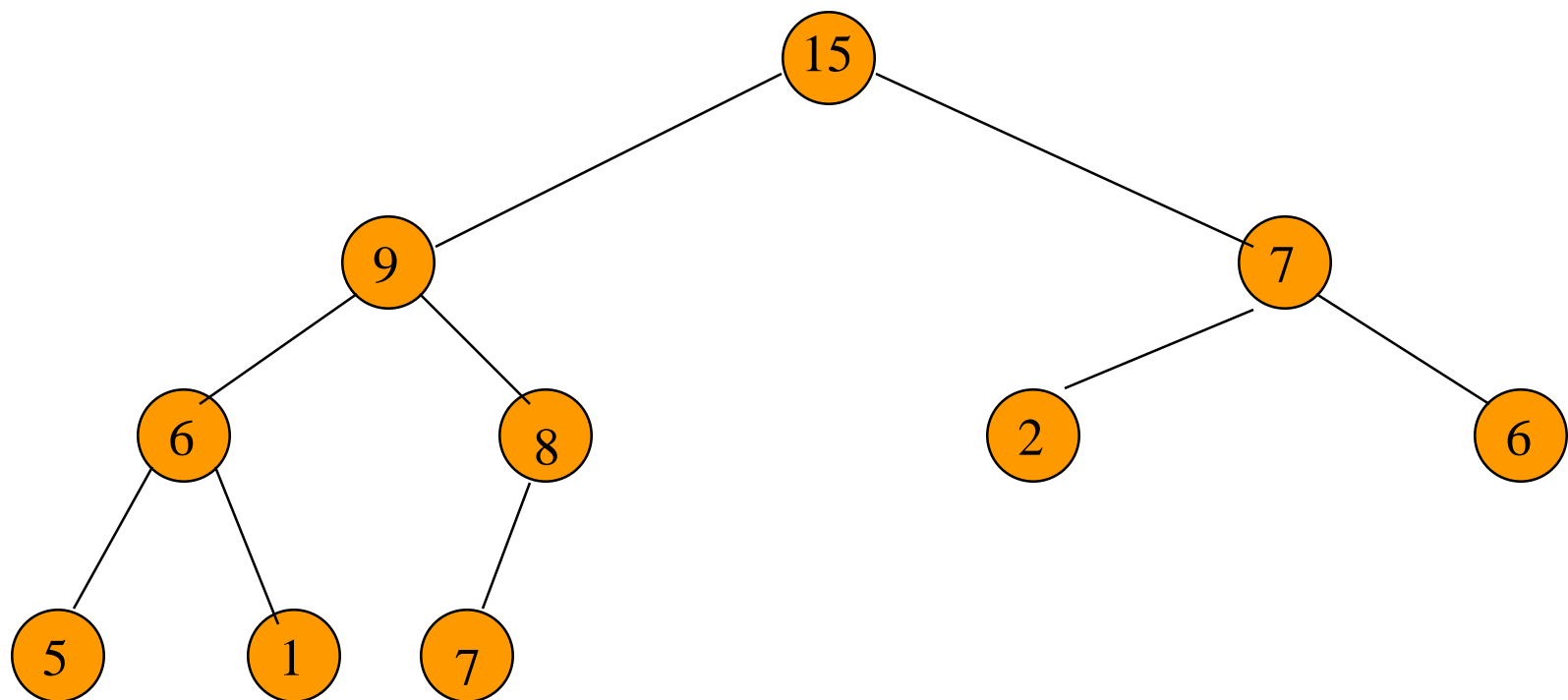
Овоолгод 8 –г дахин хийх.

Мах элементийн устгах



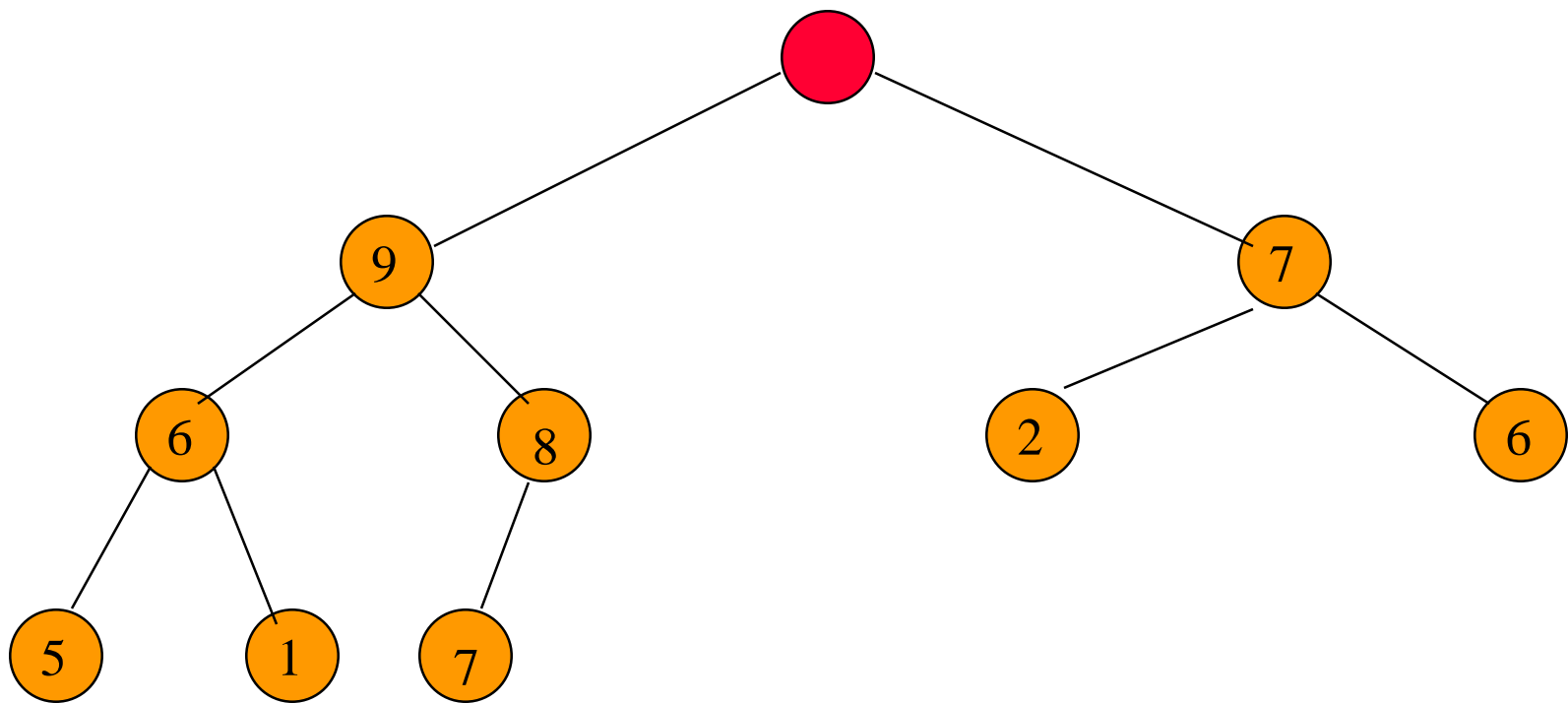
Овоолго 8 —г дахин хийх.

Мах элементийн устгах



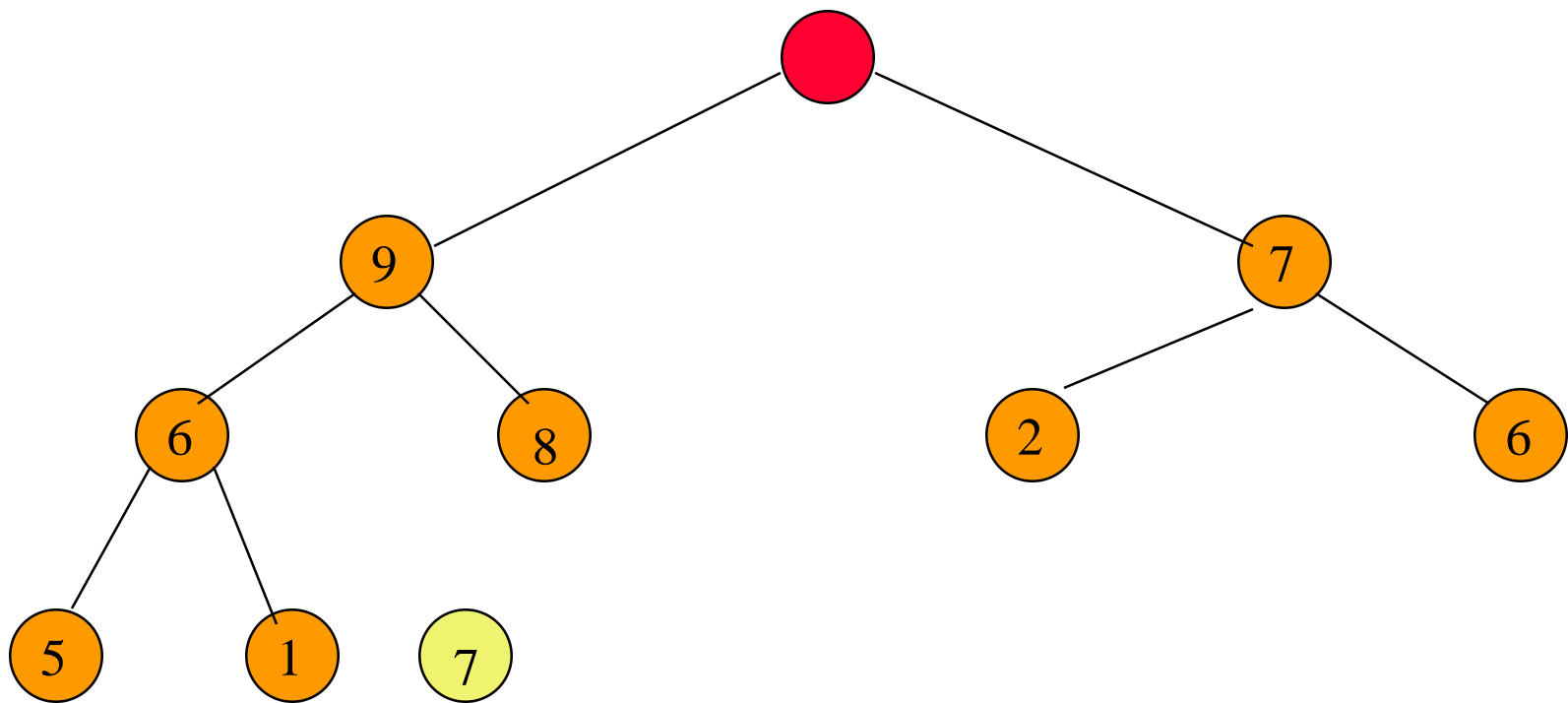
Мах элемент 15.

Мах элементийн устгах



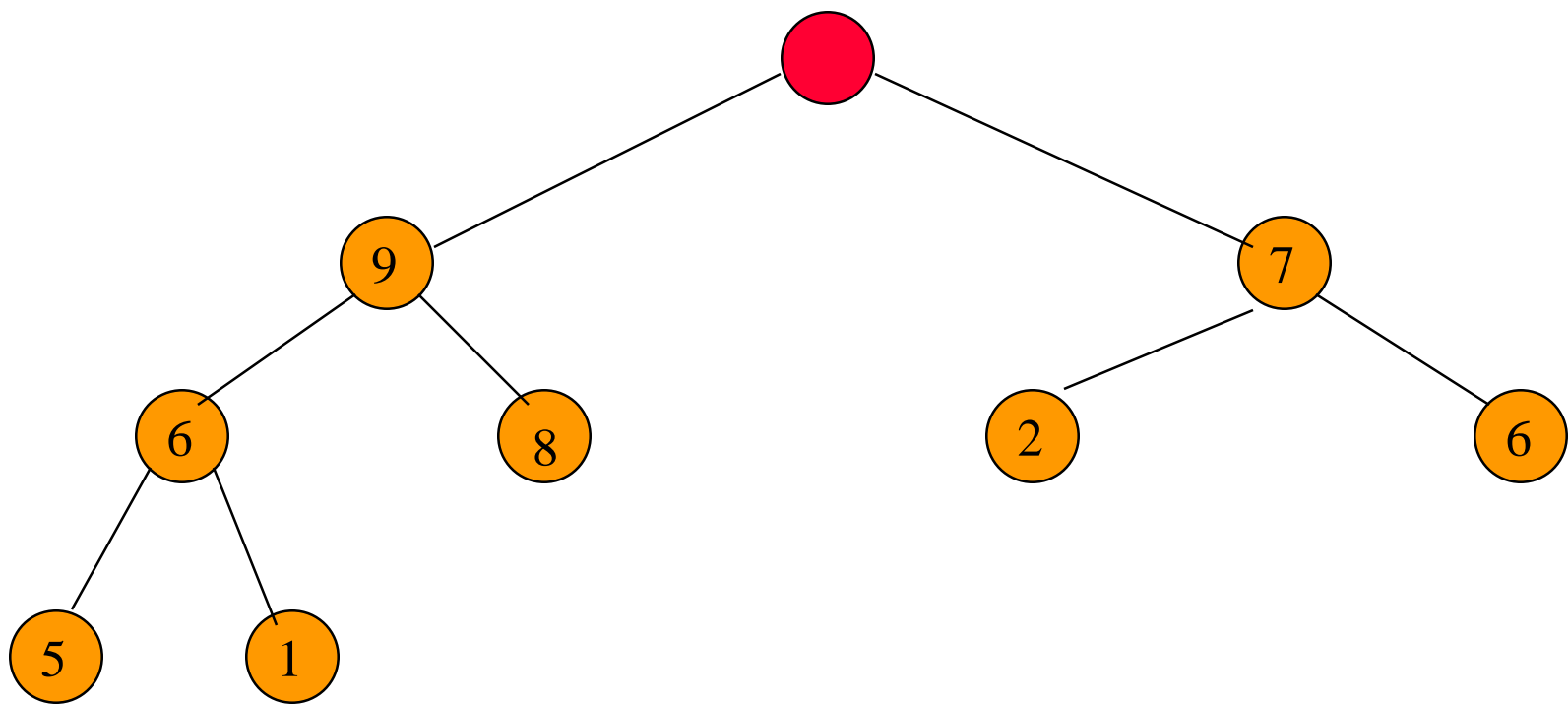
тах элементийг устгасны дараа.

Мах элементийн устгах



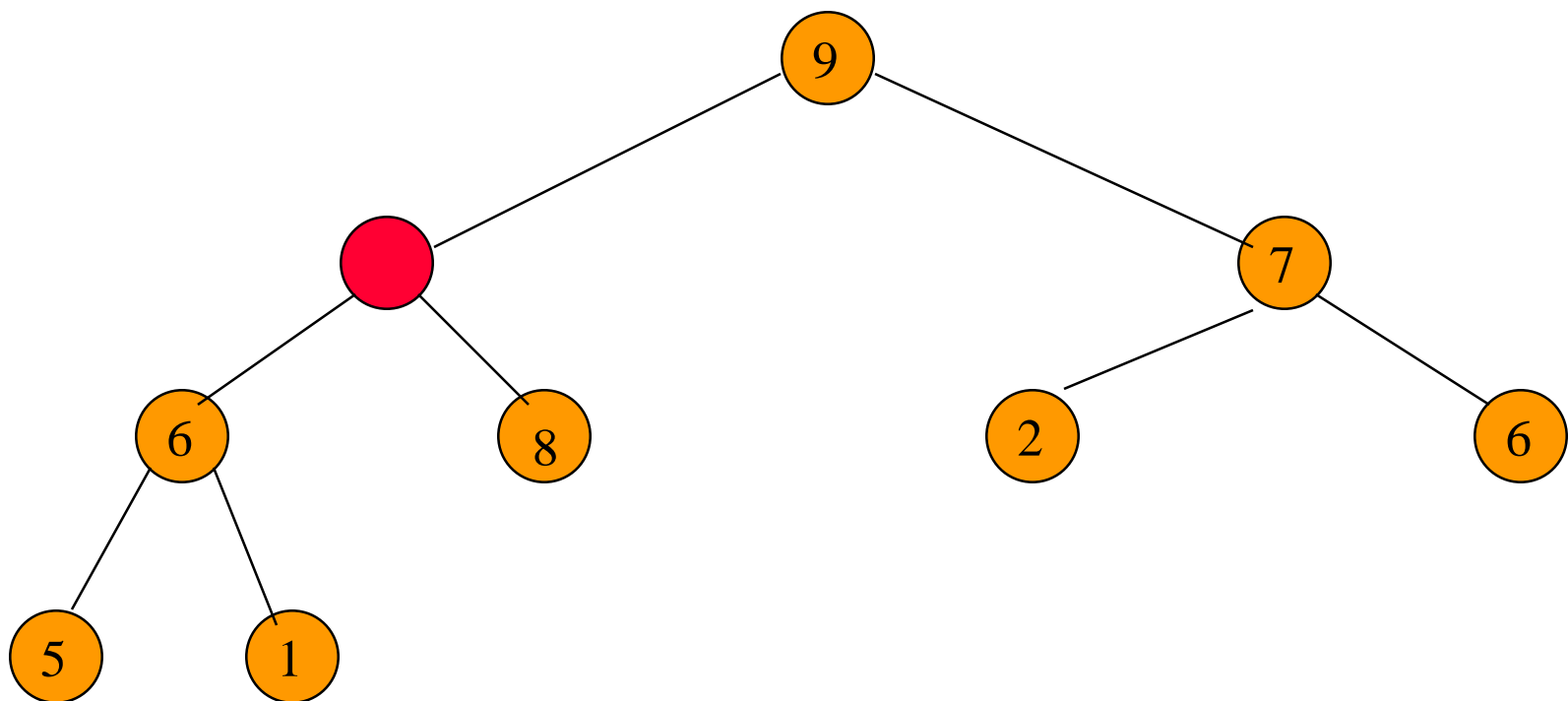
9 зангилаатай овоолго.

Мах элементийн устгах



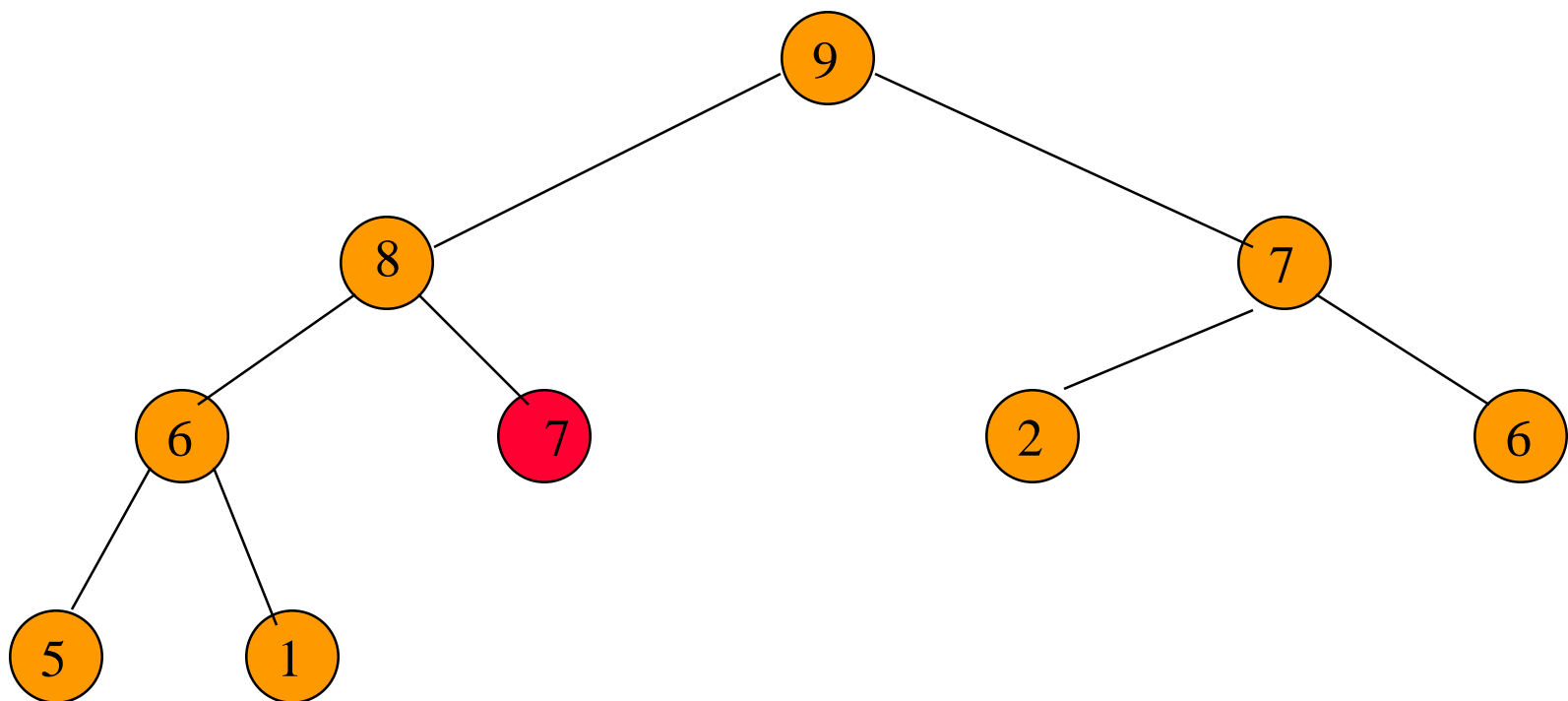
7 -г дахин хийх

Мах элементийн устгах



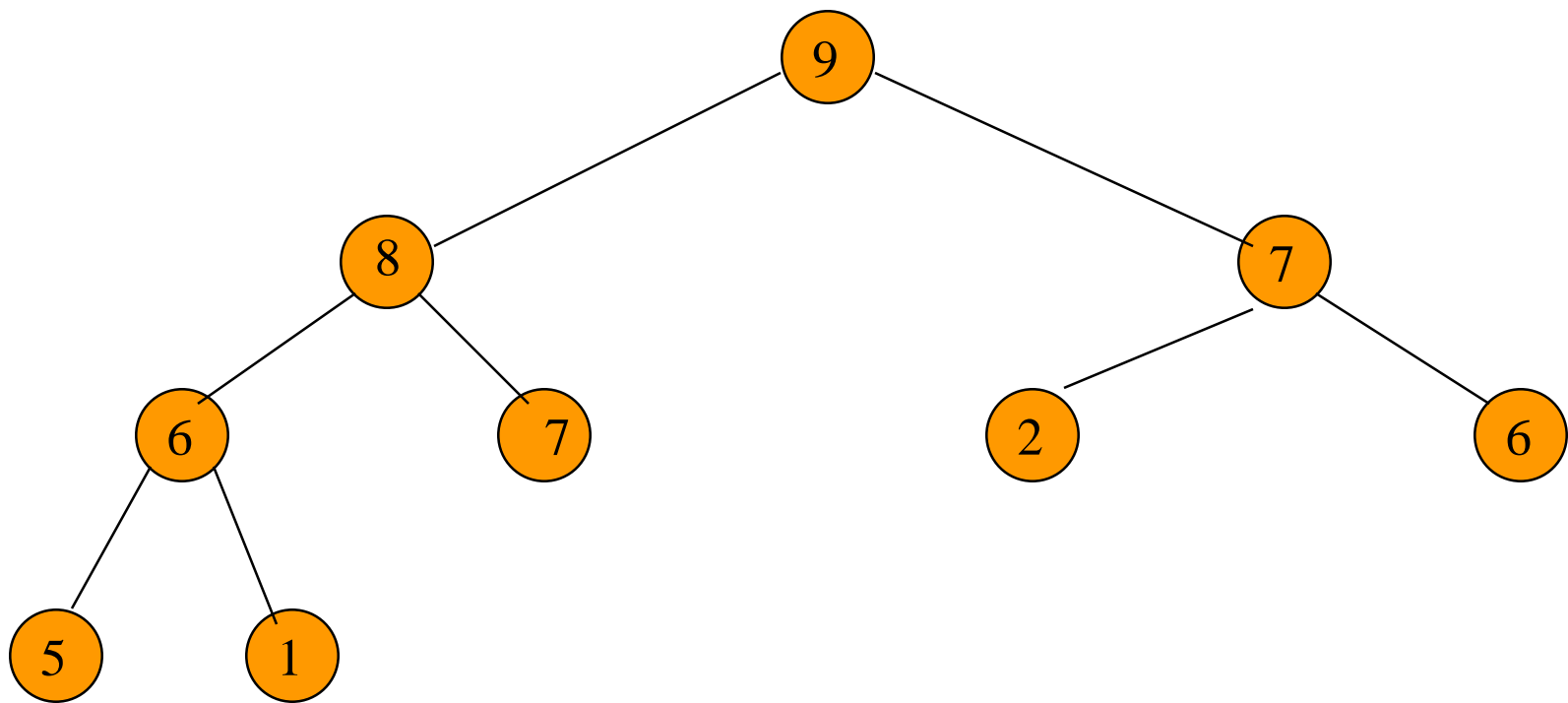
7 -г дахин хийх

Мах элементийн устгах



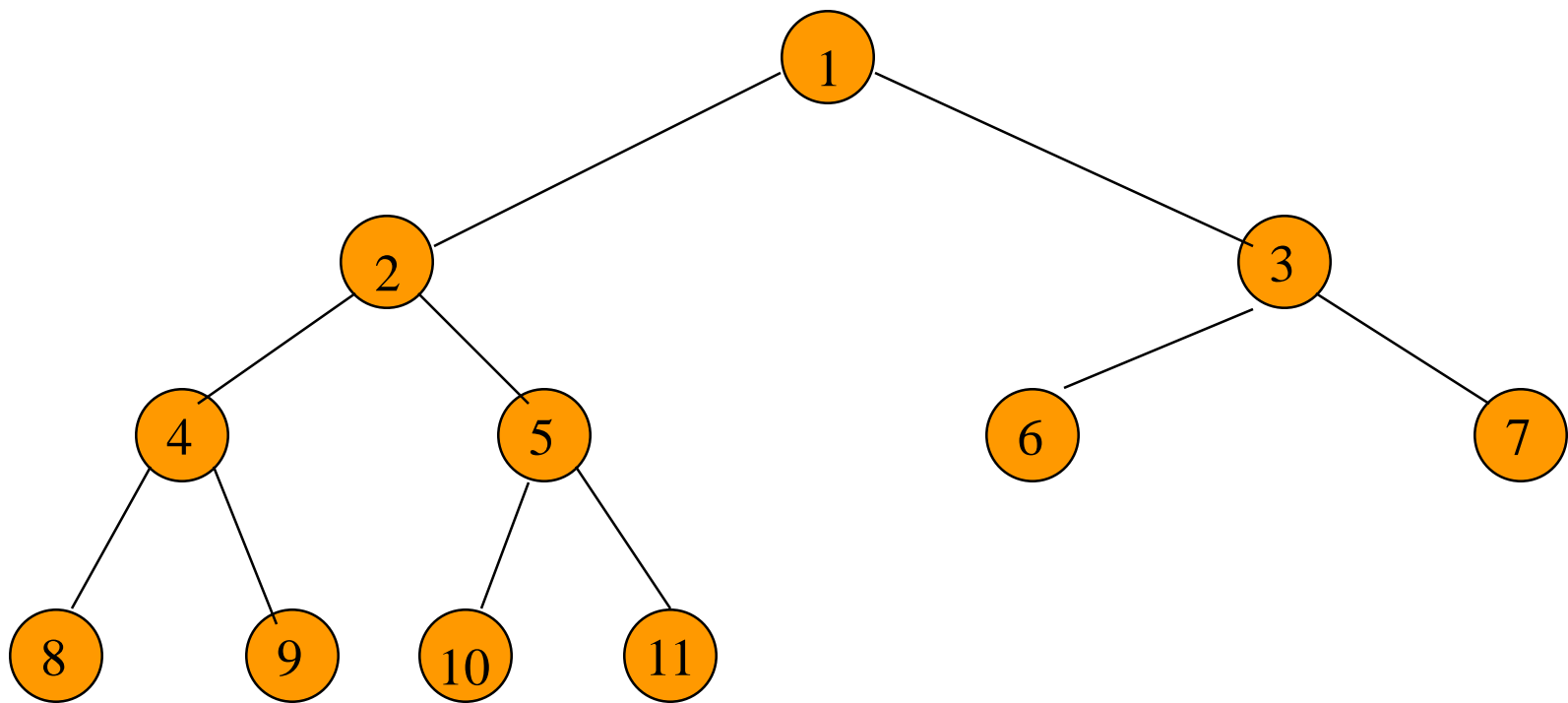
7 -г дахин хийх

Remove Max үйлдлийн хугацаа



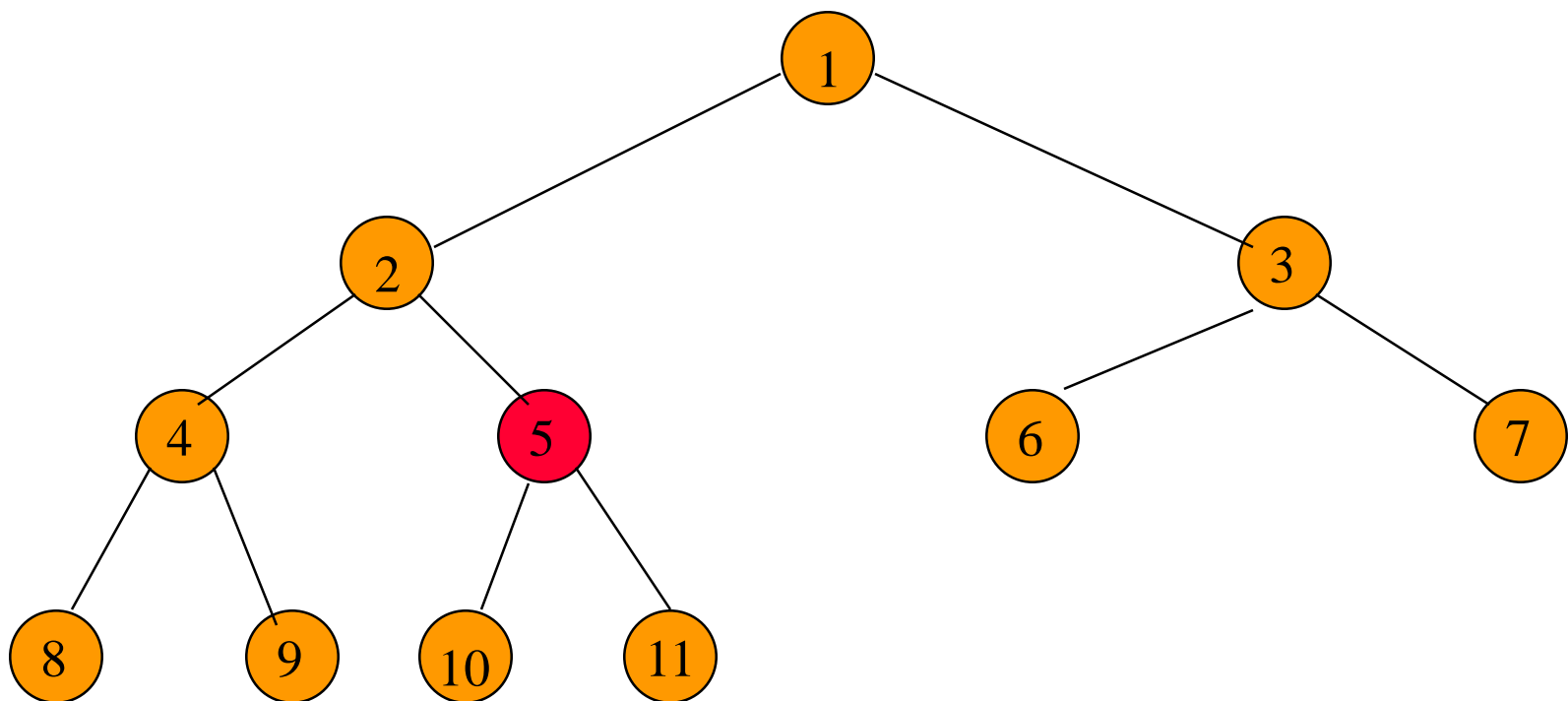
Хугацаа $O(\log n)$.

Мах овоолгыг идэхижүүлэх



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

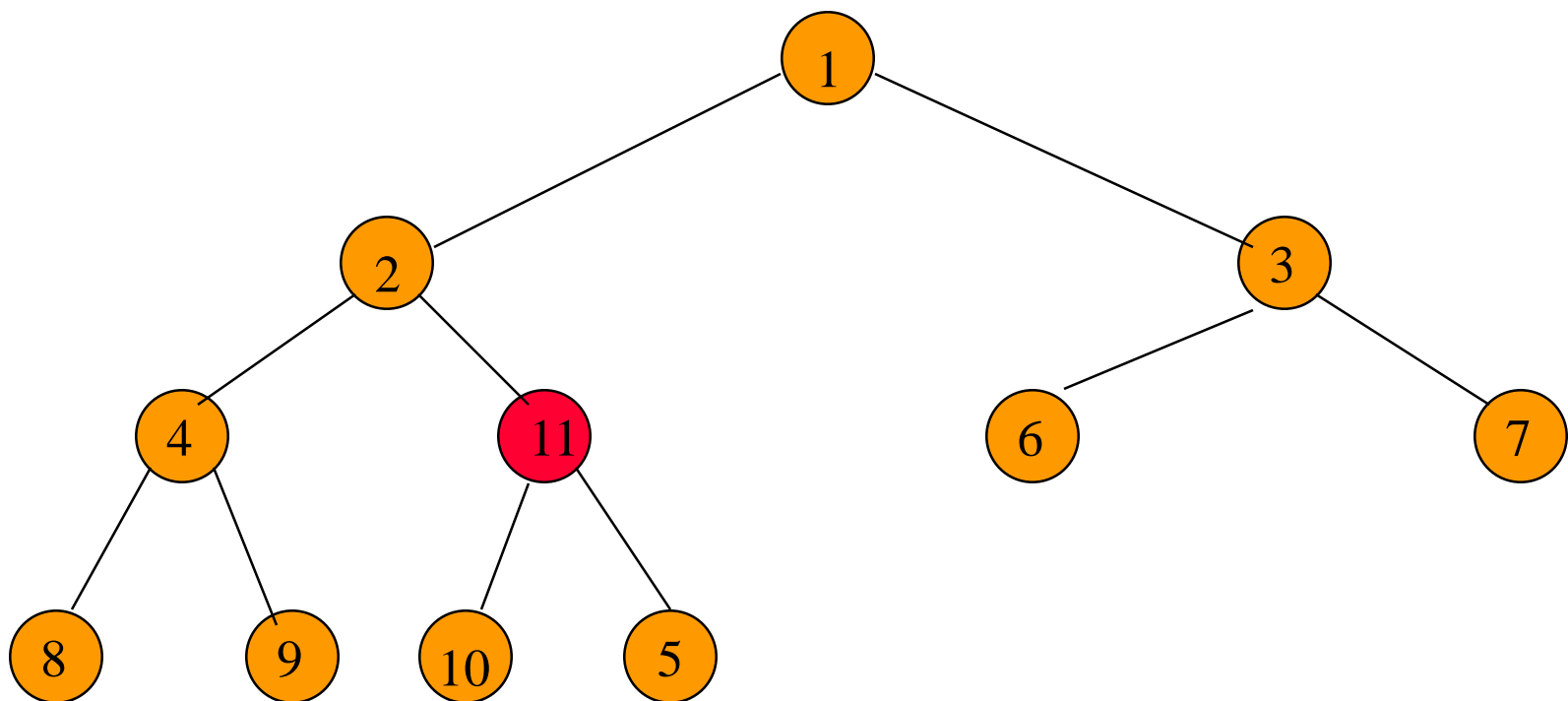
Мах овоолгыг идэхижүүлэх



Массивын хамгийн баруун байршилтай хүүхэдтэй
зангилаанаас эхэлнэ.

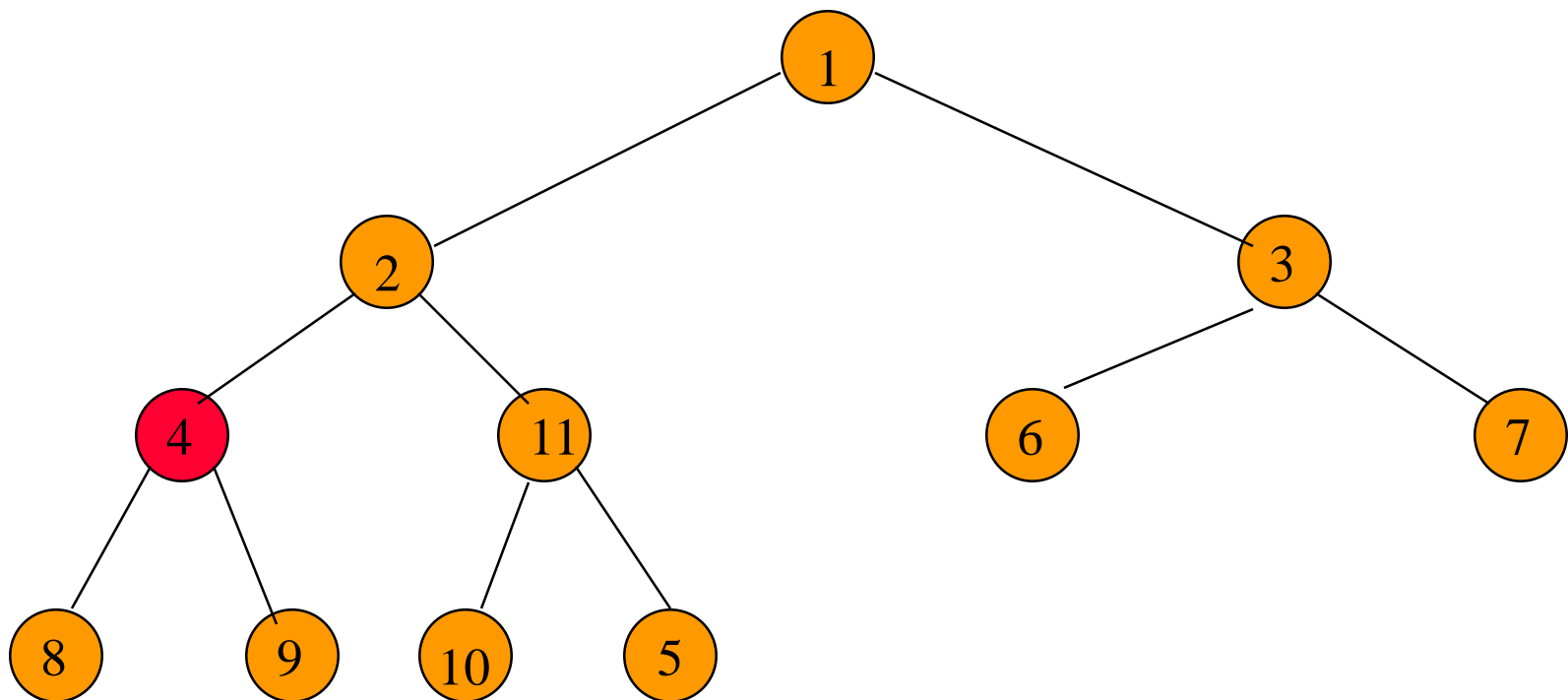
Индекс нь $n/2$.

Мах овоолгыг идэхижүүлэх

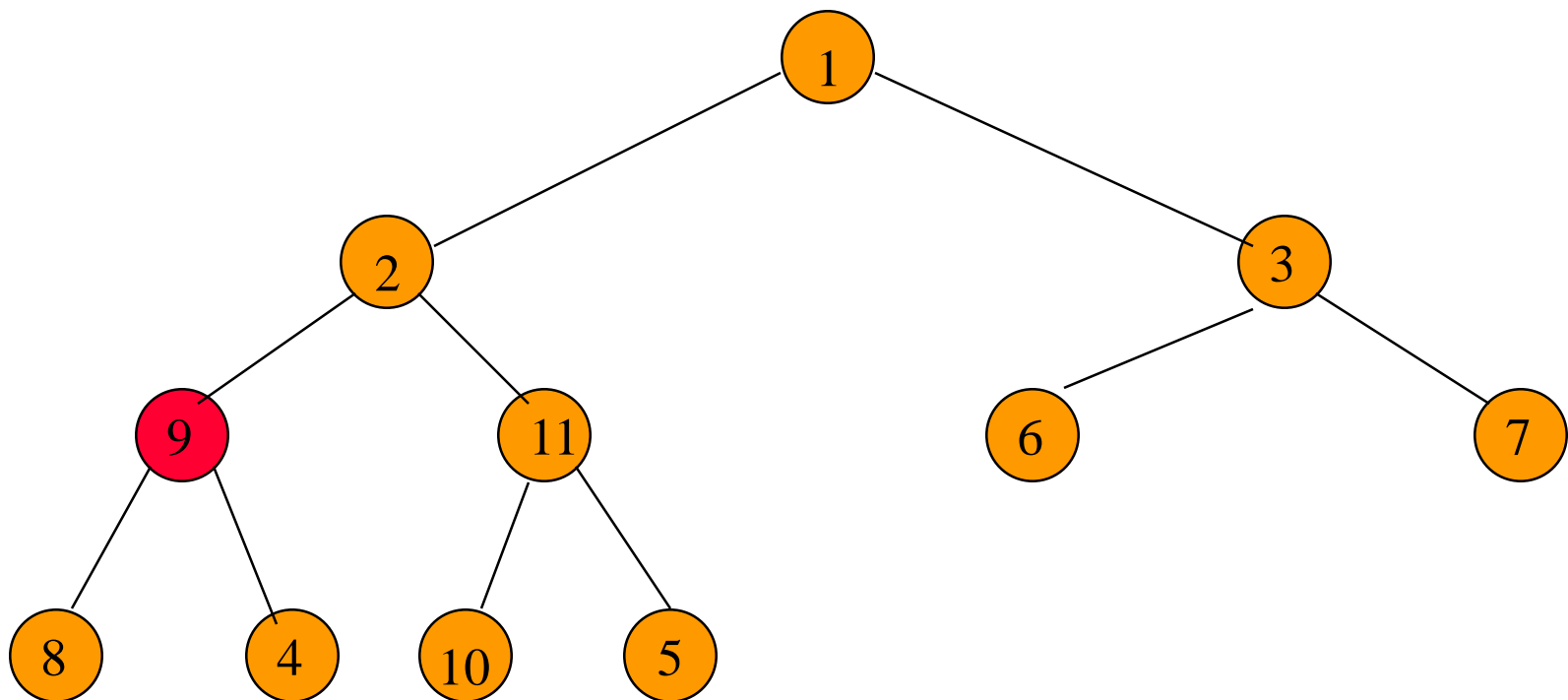


Массивын дараачийн доод байршил руу явна.

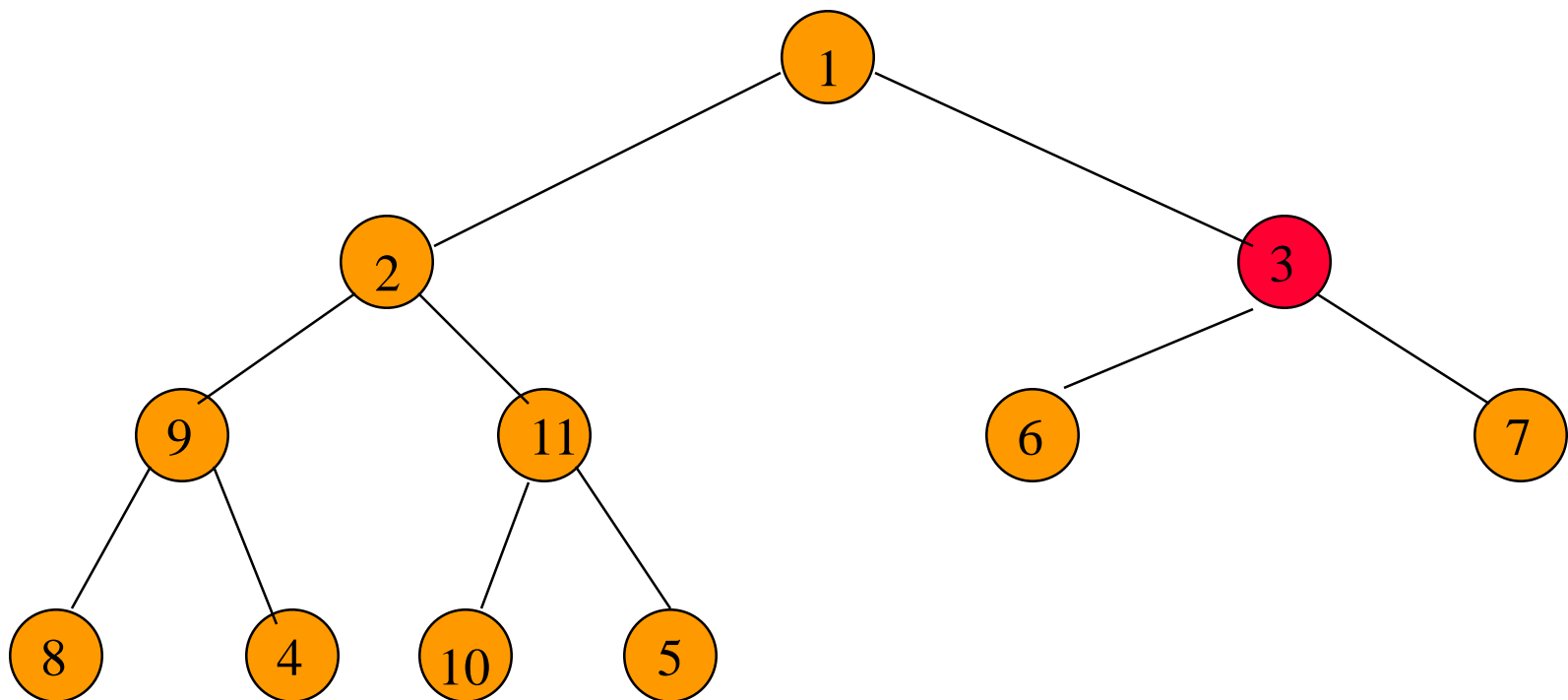
Мах овоолгыг идэхижүүлэх



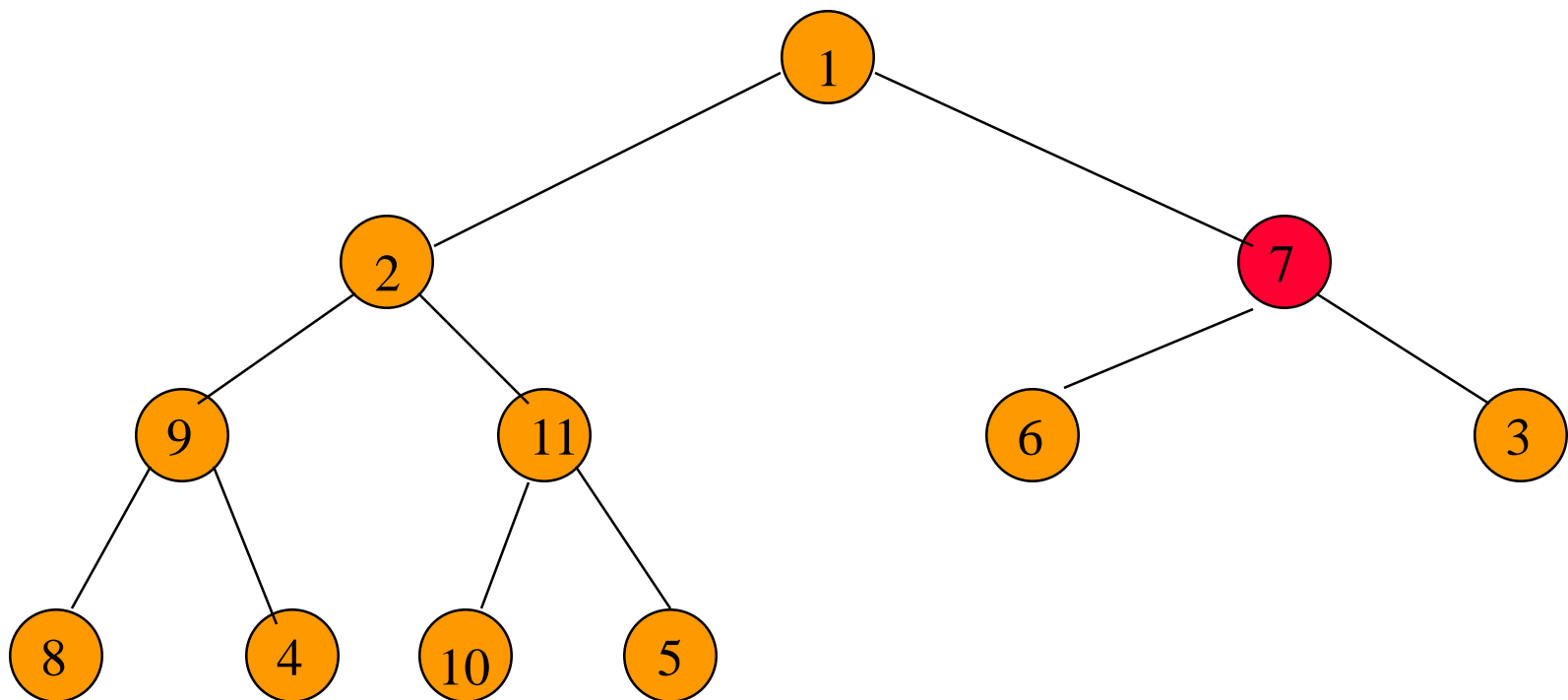
Мах овоолгыг идэхижүүлэх



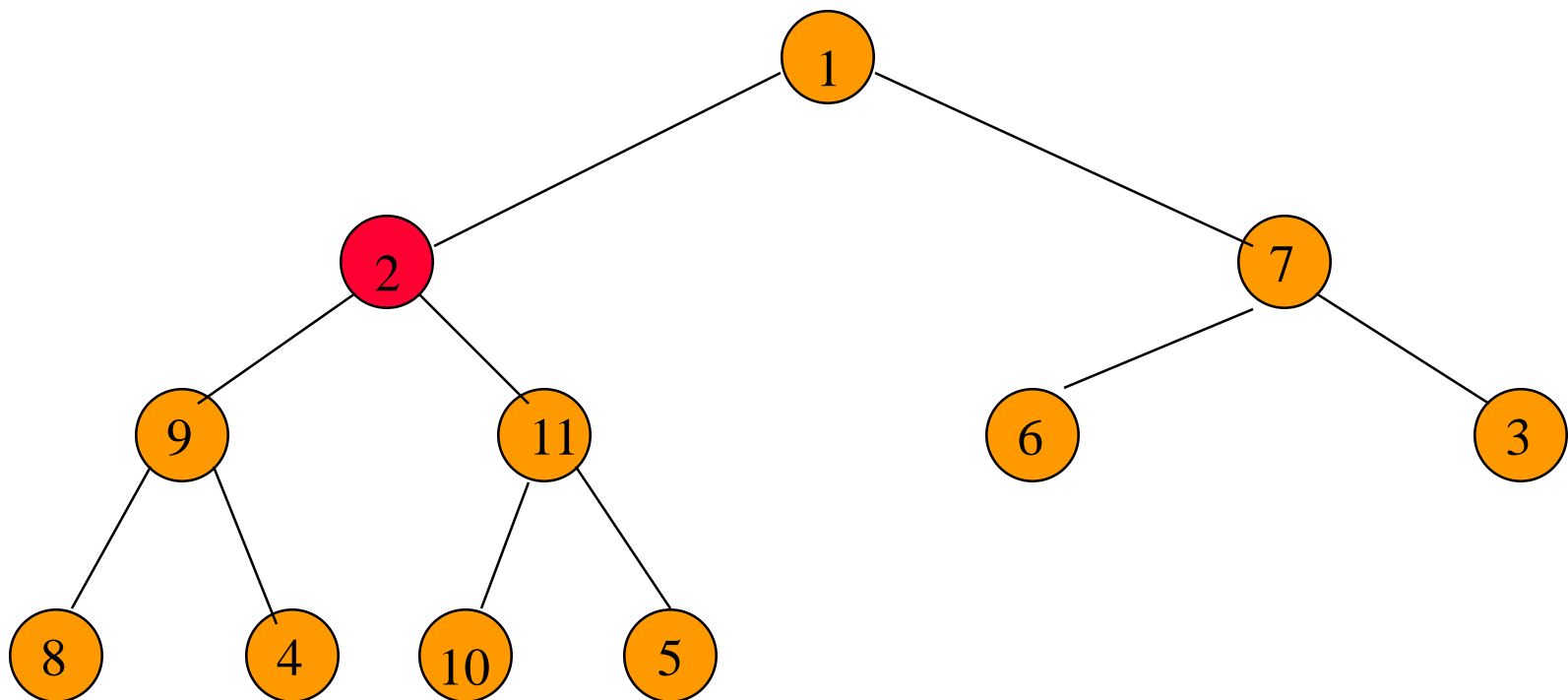
Мах овоолгыг идэхижүүлэх



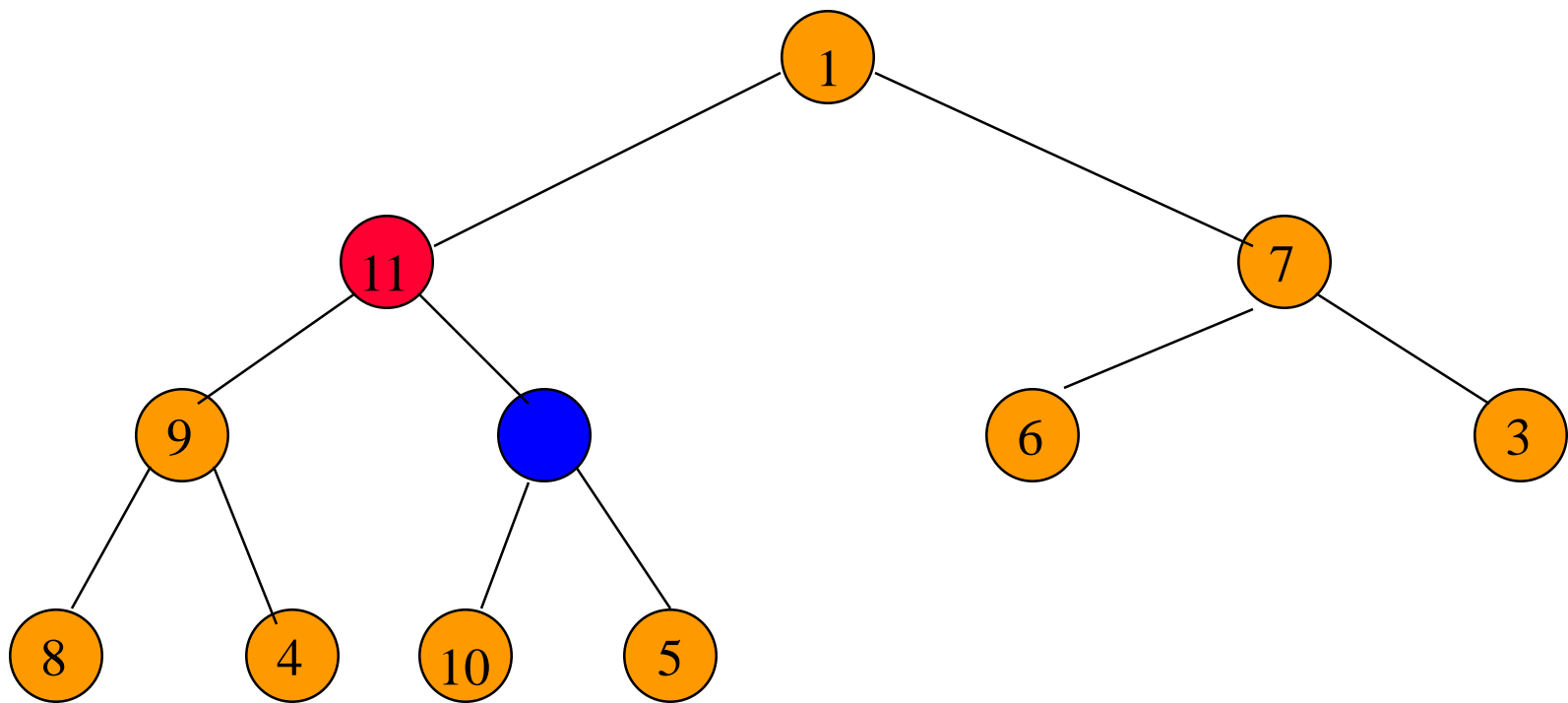
Мах овоолгыг идэхижүүлэх



Мах овоолгыг идэхижүүлэх

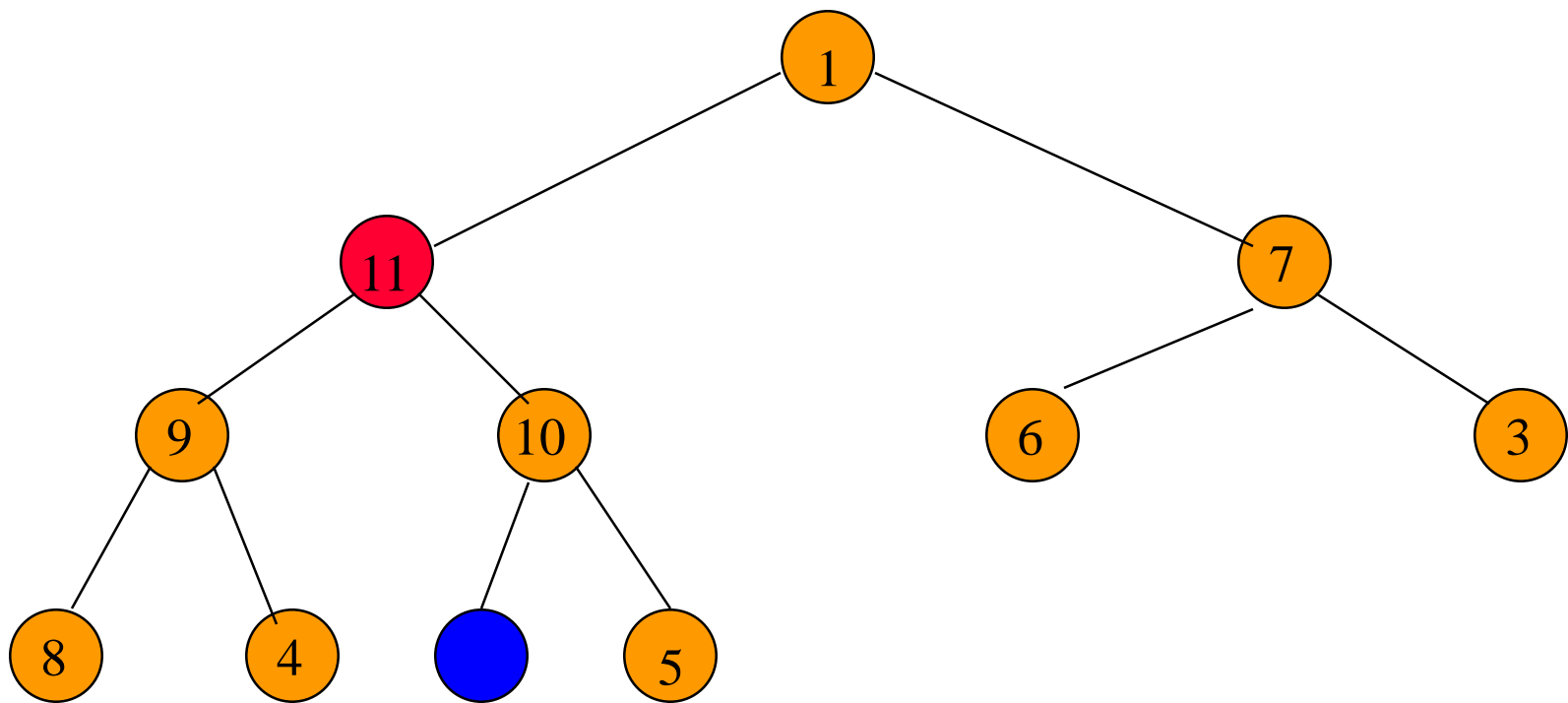


Мах овоолгыг идэхижүүлэх



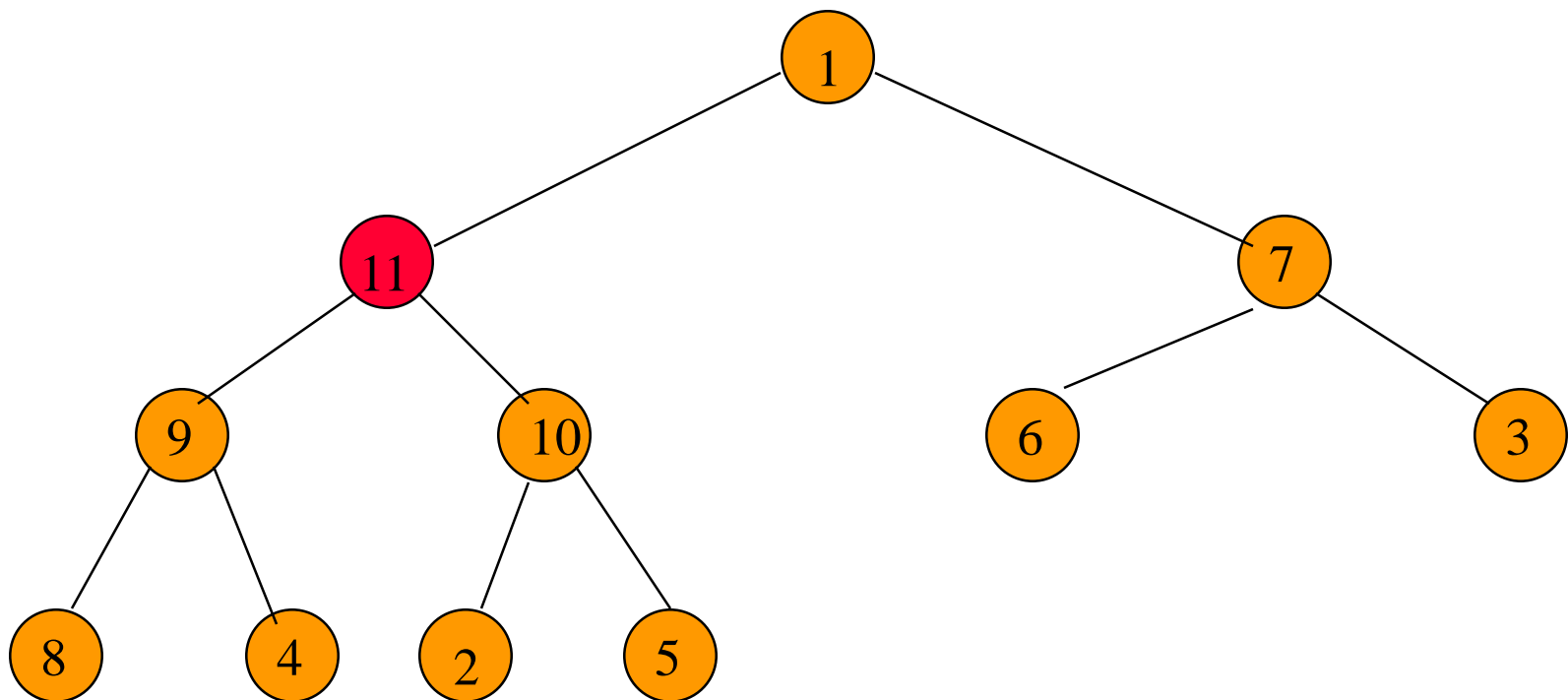
2 -н байрыг олох

Мах овоолгыг идэхижүүлэх



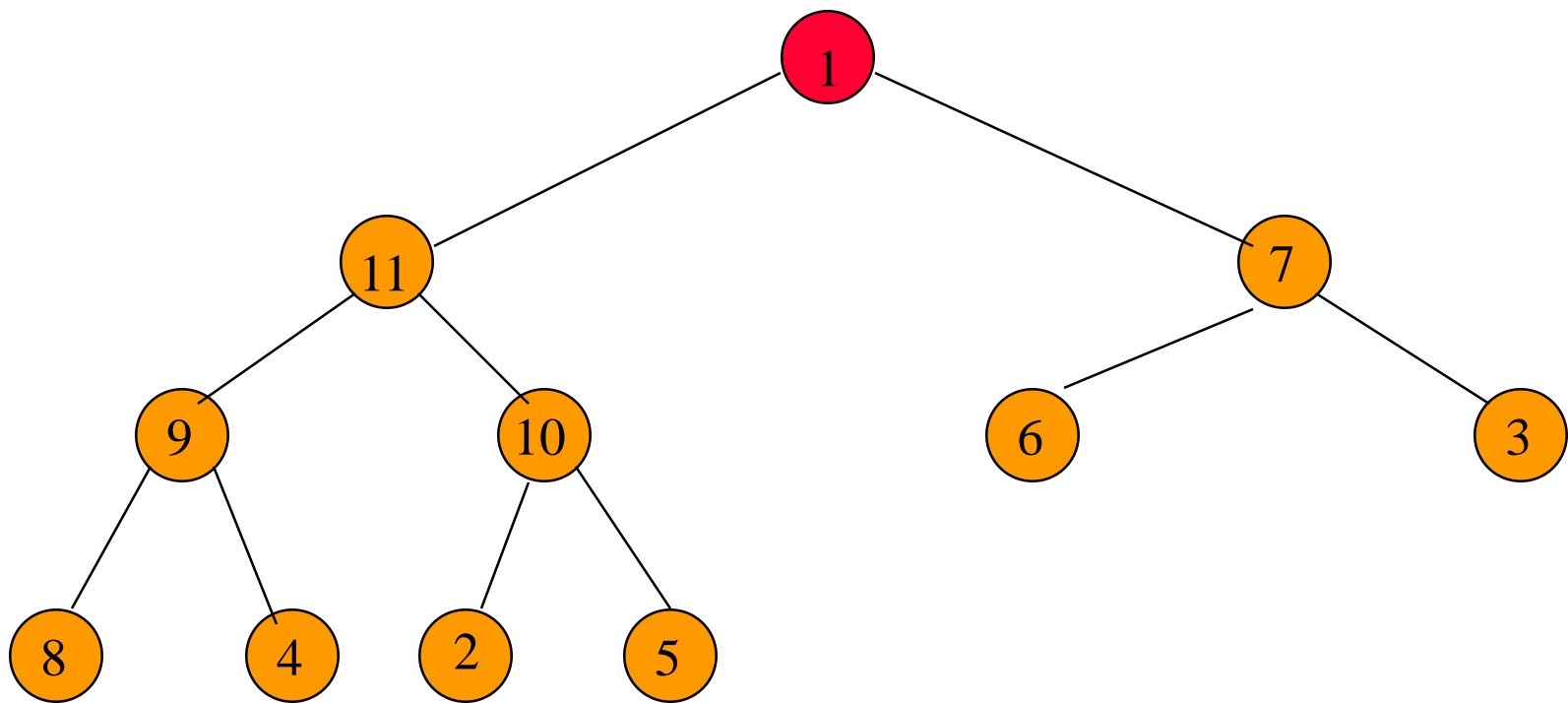
2 -н байрыг олох

Мах овоолгыг идэхижүүлэх



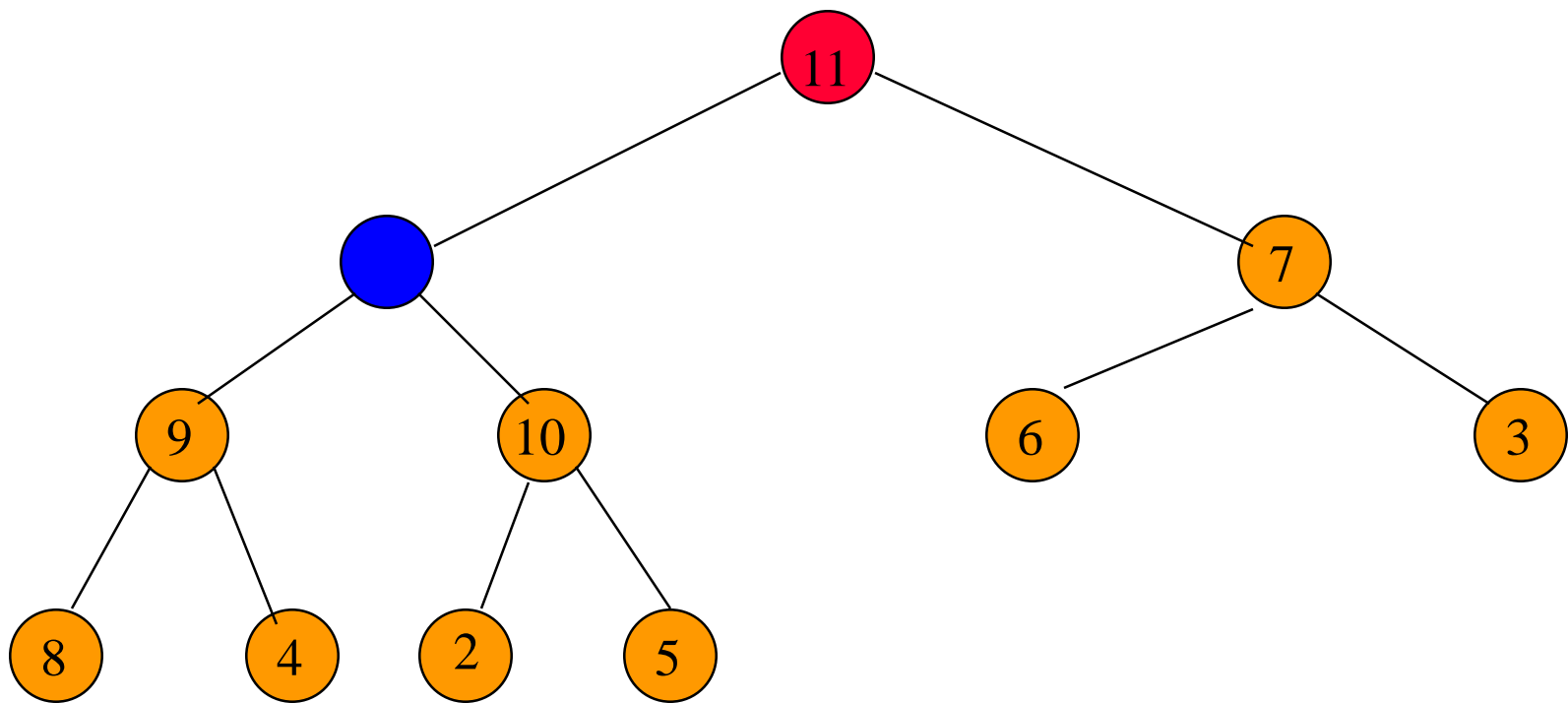
Гүйцлээ, массвын дараачийн доод байршил руу
явна.

Мах овоолгыг идэхижүүлэх



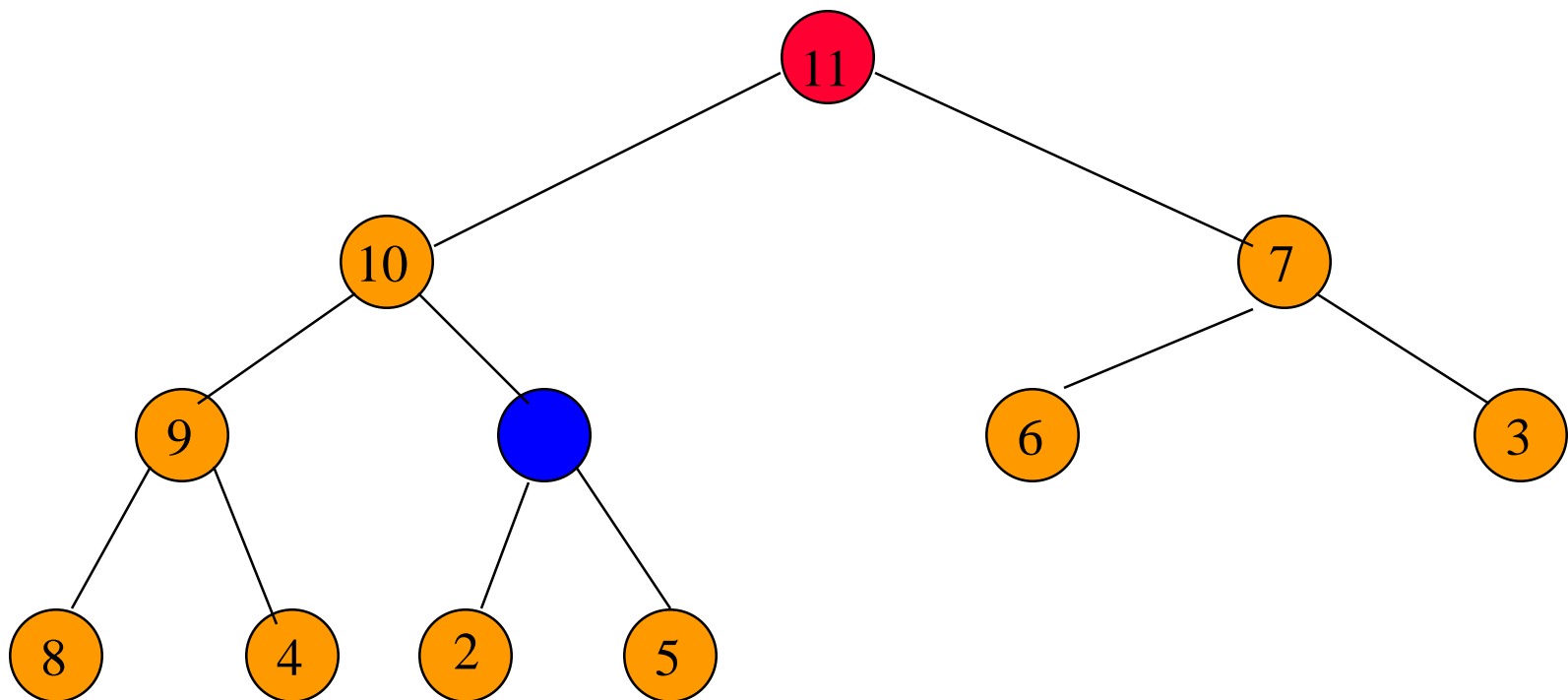
1 -н байрыг олох

Мах овоолгыг идэхижүүлэх



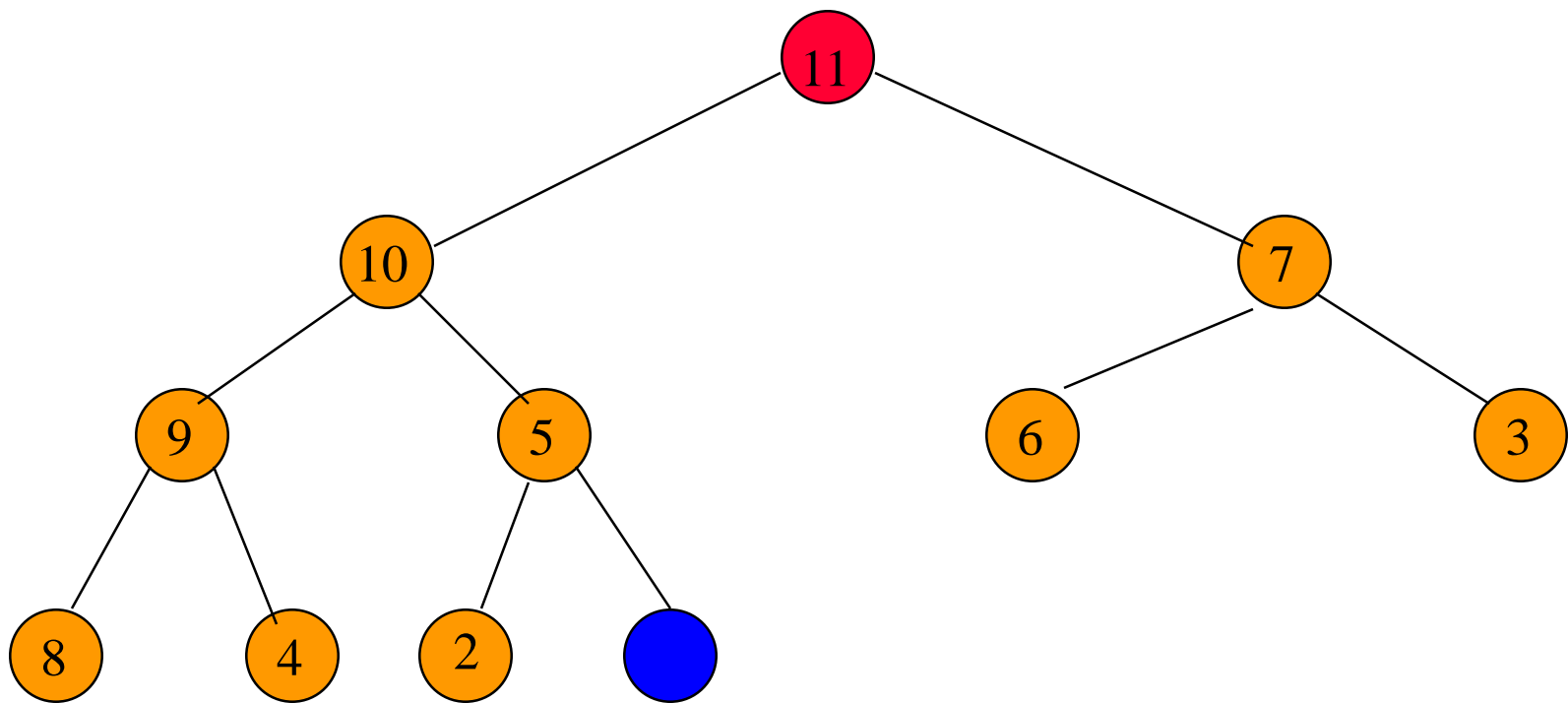
1 -н байрыг олох

Мах овоолгыг идэхижүүлэх



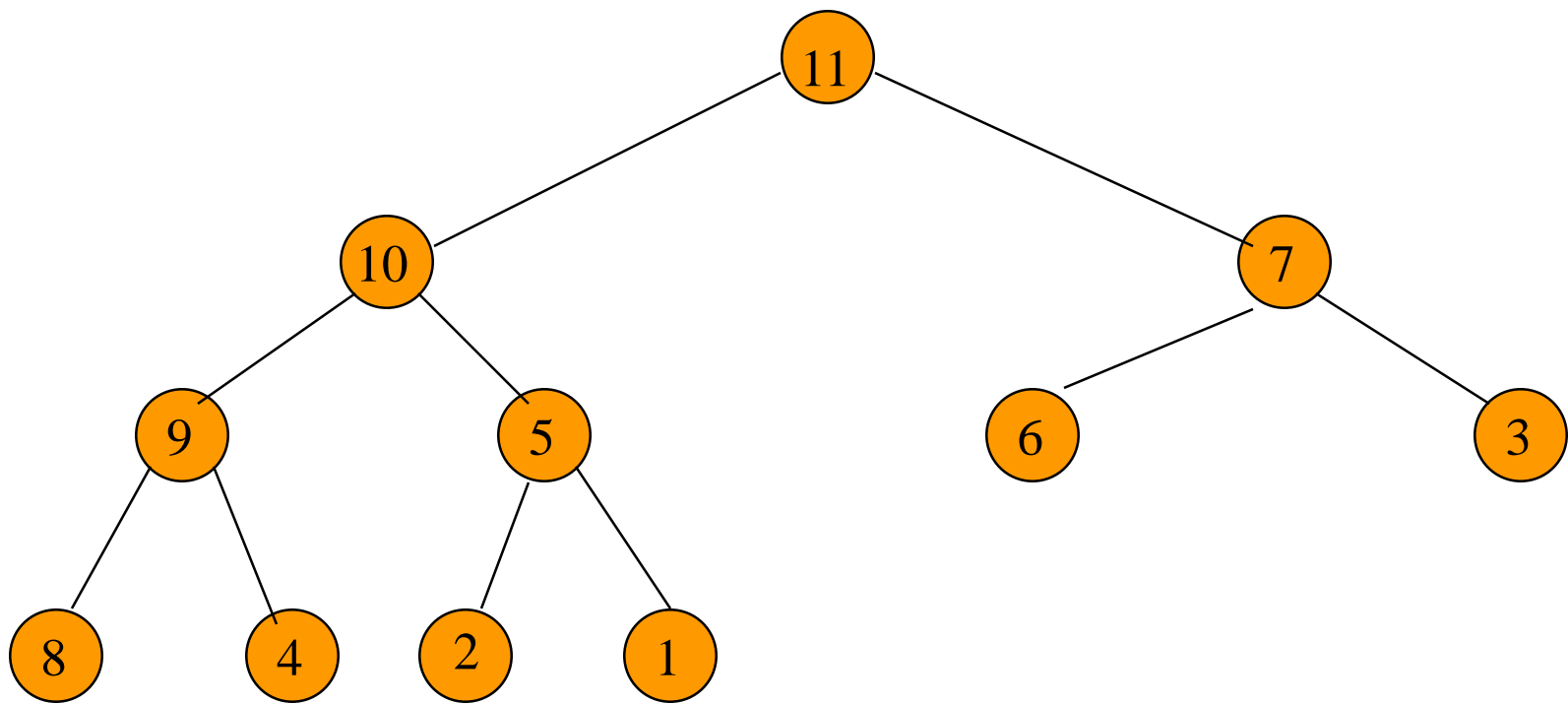
1 -н байрыг олох

Мах овоолгыг идэхижүүлэх



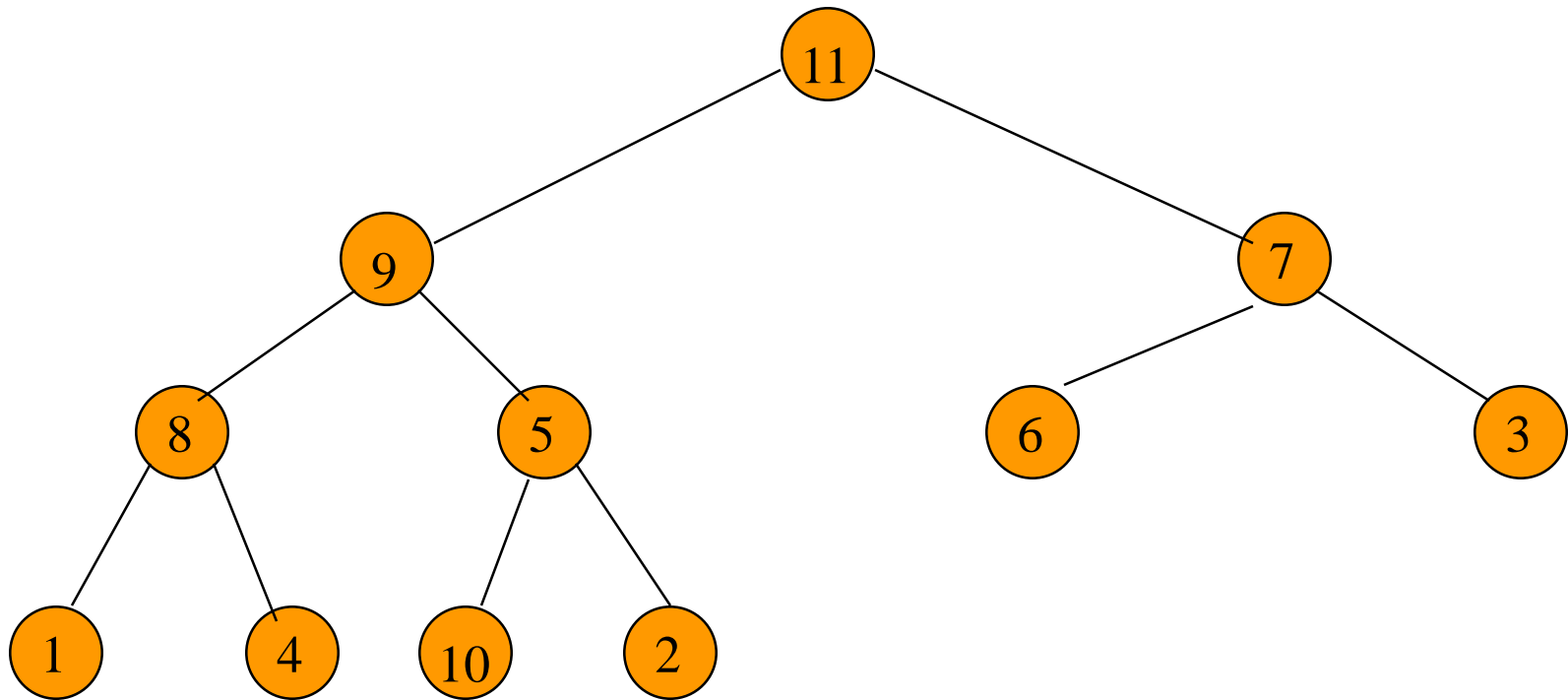
1 -н байрыг олох

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