ПРОГРАММИРОВАНИЕ

Семинар 11.
Графы: деревья

Новосибирский государственный университет, 2019

Определения

$(a, b) = \{a, \{a, b\}\}$ — упорядоченная пара.

$(a, b) = (c, d) \Leftrightarrow a = c \,\&\, b = d$

$\{a, b\} = \{b, a\}$

$A \times B = \{(a, b) \mid a \in A \,\&\, b \in B\}$ — декартово произведение.

Пример

$A = \{1, 2\}$

$B = \{2, 3, 4\}$

$A \times B = \{(1, 2), (1, 3), (1, 4), (2, 2),$
$(2, 3), (2, 4)\}$

Отношения

Произвольное подмножество $A \times B$ называется отношением из A в B.

A — область определения.

B — область значений.

Если $A = B$, то отношение задано на A.

$(a, b) \in R \Leftrightarrow aRb$

Свойства отношений

Пусть на множестве A задано отношение R:

- рефлексивное, если $aRa \forall a \in A$;
- симметричное, если $aRb \Rightarrow bRa \forall a, b \in A$;
- транзитивное, если $aRb \ \& \ bRc \Rightarrow aRc \forall a, b, c \in A$.

Рефлексивное, симметричное и транзитивное отношение называется отношением эквивалентности.

$A = U_{a \in A} A_a$ — разбиение:

$A_a = \{b \mid aRb\}$ — класс эквивалентности;

$A_a \cap A_b = \emptyset, a \neq b$.

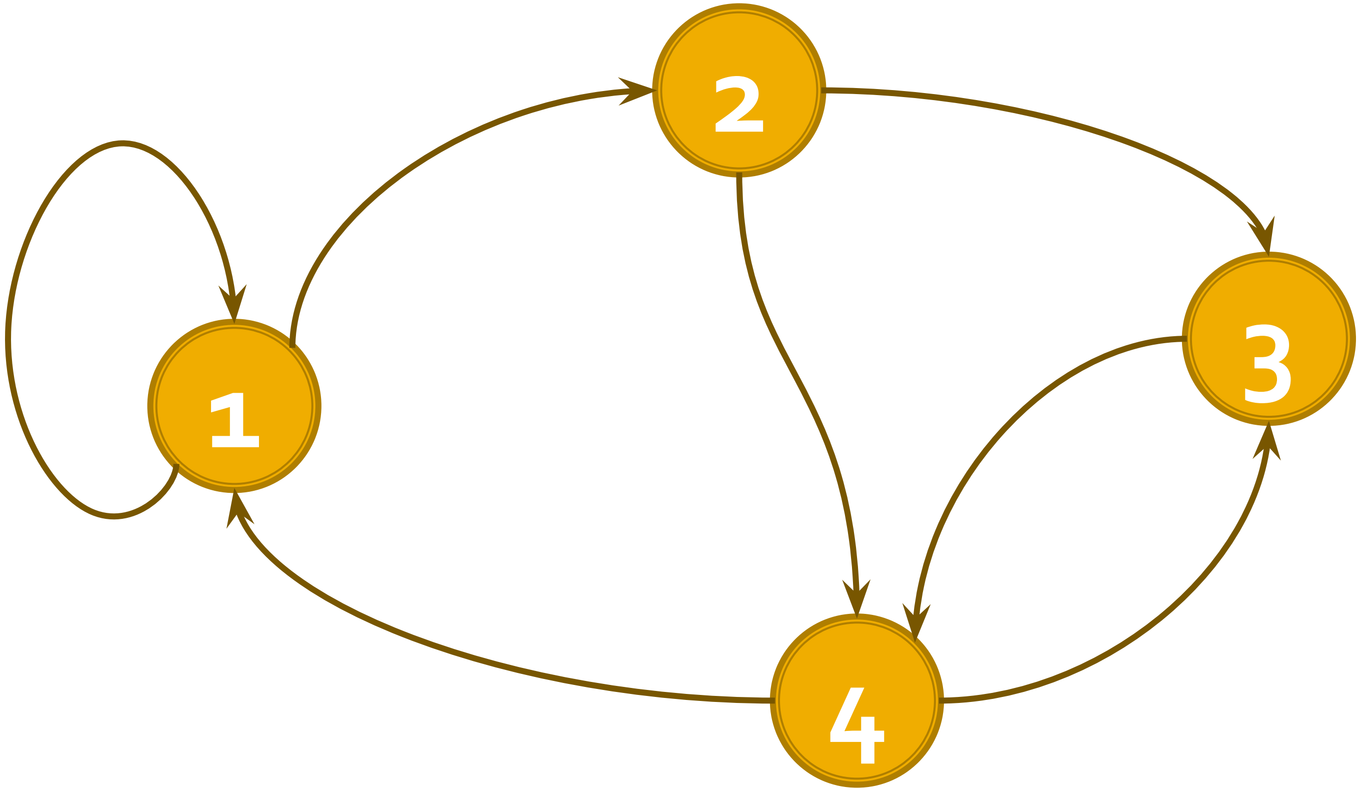
Графы

Неупорядоченный граф $G = (V, E)$, где:

- V — множество вершин;
- E — отношение на V.

Граф G ориентированный (орграф), если E — асимметричное, иначе — неориентированный.

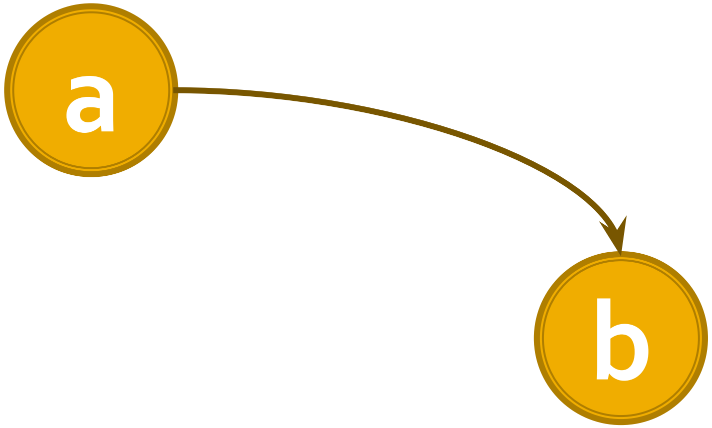
Пример



$V = \{1, 2, 3, 4\}$

$E = \{(1, 1), (1, 2), (2, 3), (2, 4), (3, 4), (4, 1), (4, 3)\}$

Графы



$(a, b) \in R$ — дуга (ребро):

- дуга выходит из a и входит в b;
- a предшествует b;
- b следует за a;
- b смежна с a.

Пути в графах

Последовательность вершин $(a_0, a_1, a_2, \ldots, a_n)$, $n \geq 1$ называется путём (маршрутом) длины n из a_0 в a_n, если $(a_{i-1}, a_i) \in E \; \forall i \in \{1, 2, \ldots, n\}$, при этом, говорят, что a_n достижима из a_0.
Путь, в котором $a_0 = a_n$, называется циклом.
Орграф называется сильно связным, если для любых его двух вершин существует путь из одной в другую.

Степень вершины

Степень по входу (полустепень входа) вершины — число входящих в неё дуг.
Степень по выходу (полустепень выхода) вершины — число исходящих из неё дуг.
Если граф неориентированный, то степень вершины — количество связанных с ней рёбер.

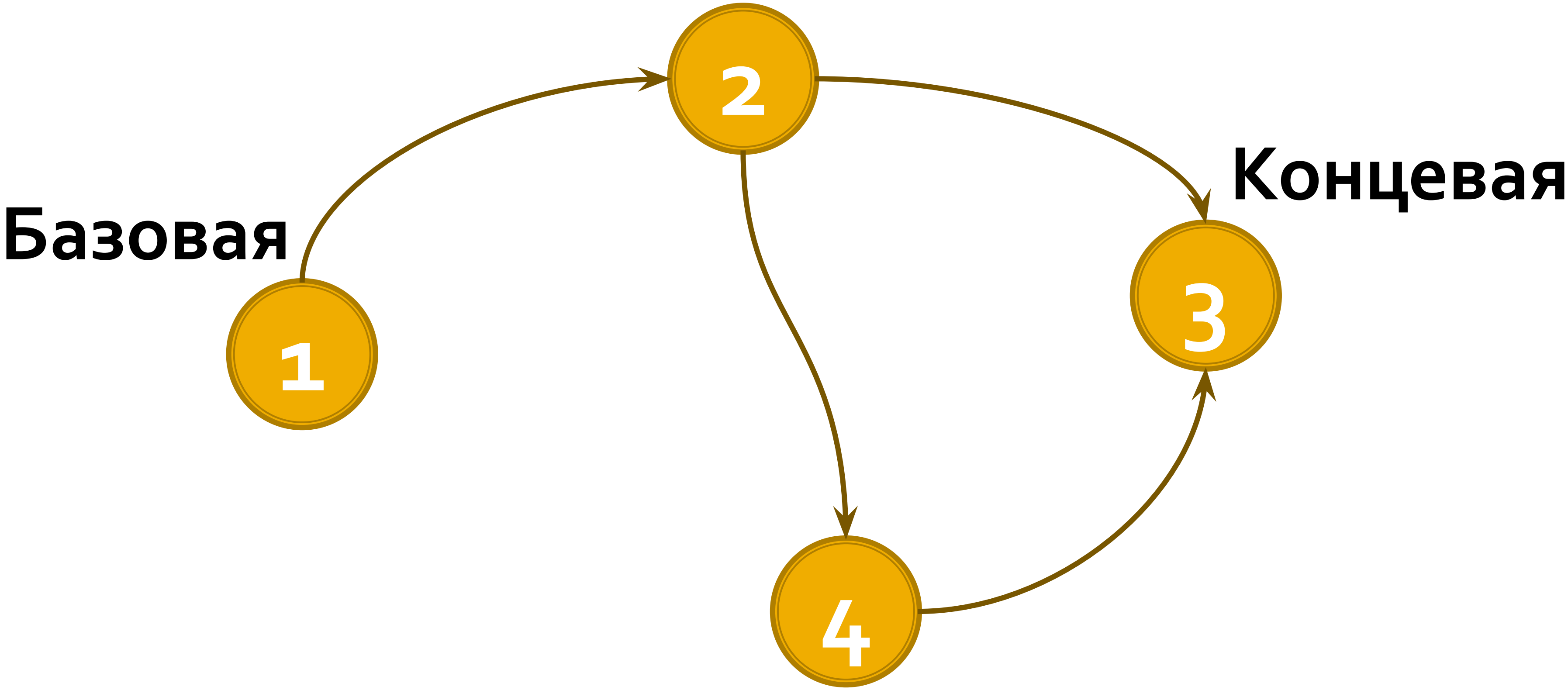
Ациклические графы

Ациклический граф — орграф без циклов.

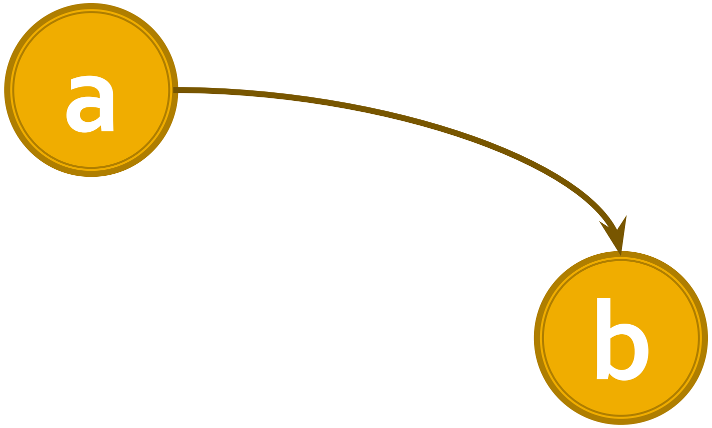
Вершина с полустепенью входа *0* — базовая.

Вершина с полустепенью выхода *0* — лист (концевая).

Пример



Ациклические графы



$(a, b) \in R$ — дуга:

a — прямой предок b;

b — прямой потомок a.

Если существует путь из a в b, то:

a — предок b;

b — потомок a.

Деревья

(Ориентированное) дерево — (ориентированный) граф со специальной вершиной *r* (корнем):

- полустепень по входу равна *0*;
- полустепень по входу всех остальных равна *1*;
- каждая вершина достижима из корня.

Свойства:

1. Дерево — ациклический граф.
2. Для каждой вершины дерева существует единственный путь в неё из корня.

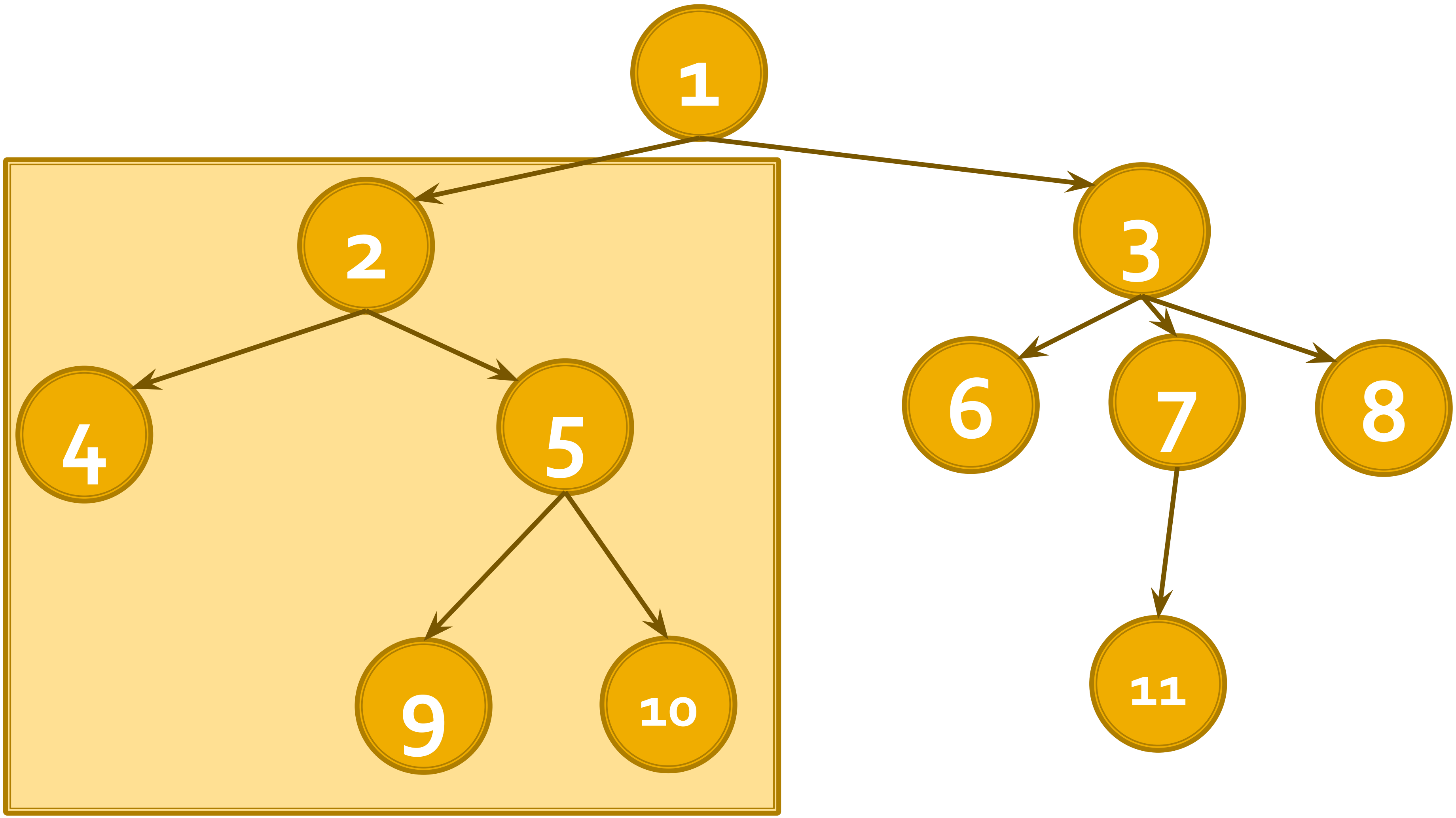
Поддерево

Поддерево дерева $T = (V, E)$ — любое дерево $T' = (V', E')$:

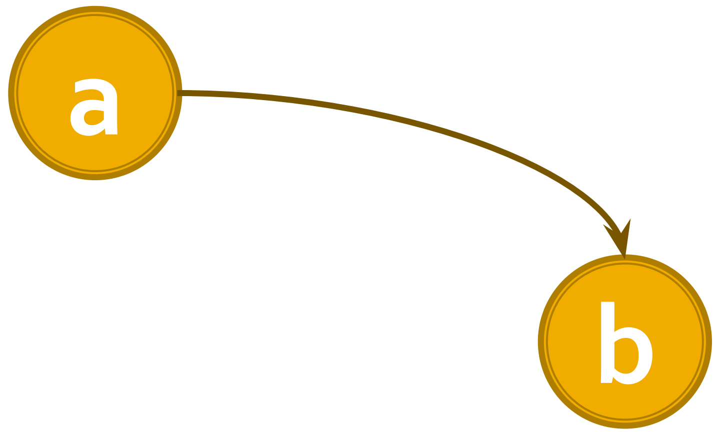
1. $V' \neq \emptyset \;\&\; V' \subseteq V$.
2. $E' = (V' \times V') \cap E$.
3. Ни одна вершина из $V \setminus V'$ не является потомком вершины из V'.

Орграф из нескольких деревьев — лес.

Пример



Деревья



(a, b)∈ R — дуга:

a — отец *b*;

b — сын *a*.

Глубина (уровень) вершины — длина пути до неё из корня.

Высота вершины — длина максимального пути от неё до листа.

Высота дерева — длина максимального пути от корня до листа.

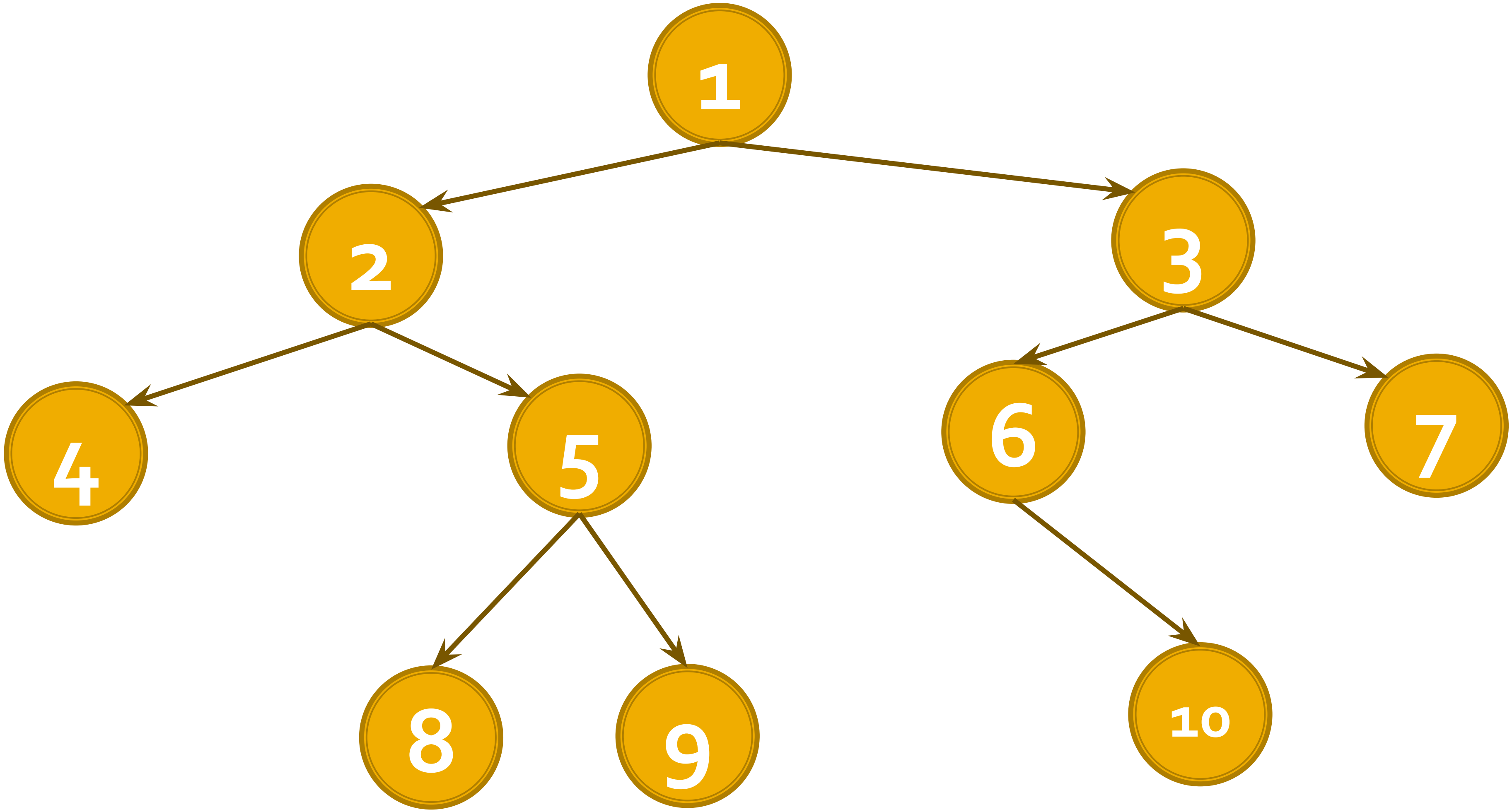
Бинарные деревья

Упорядоченное дерево — дерево, в котором множество сыновей каждой вершины упорядочено слева направо.

Бинарное дерево — это упорядоченное дерево:

1. Любой сын либо левый, либо правый.
2. Любая вершина имеет не более одного левого и не более одного правого сына.

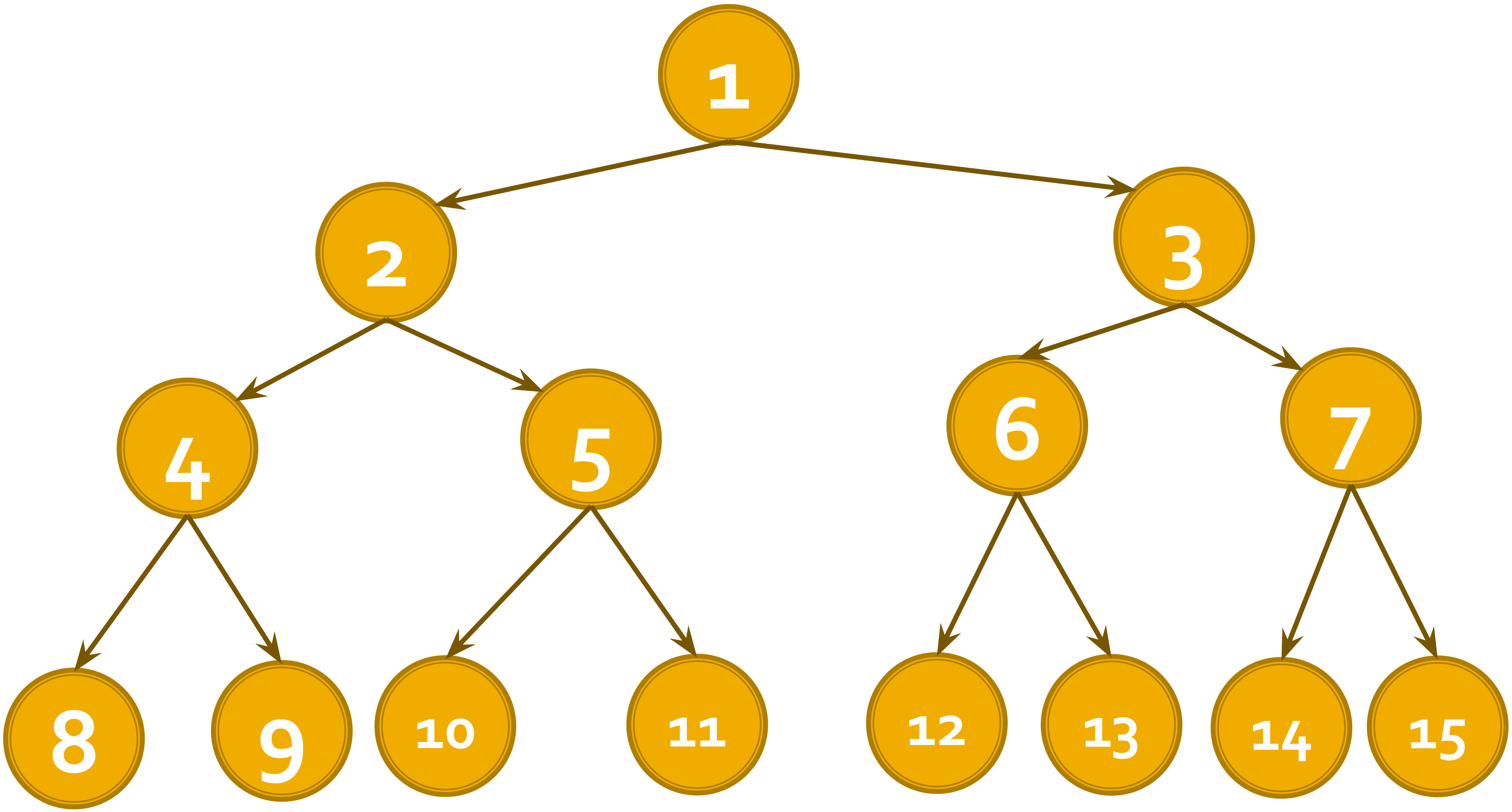
Пример



Бинарные деревья

Бинарное дерево полное, если существует целое k, такое что любая вершина глубины меньше k имеет и левого, и правого сыновей, а любая вершина глубины k — лист.

Пример



Представление полных бинарных деревьев

Массив *T[2^k - 2]*:

T[0] — корень;

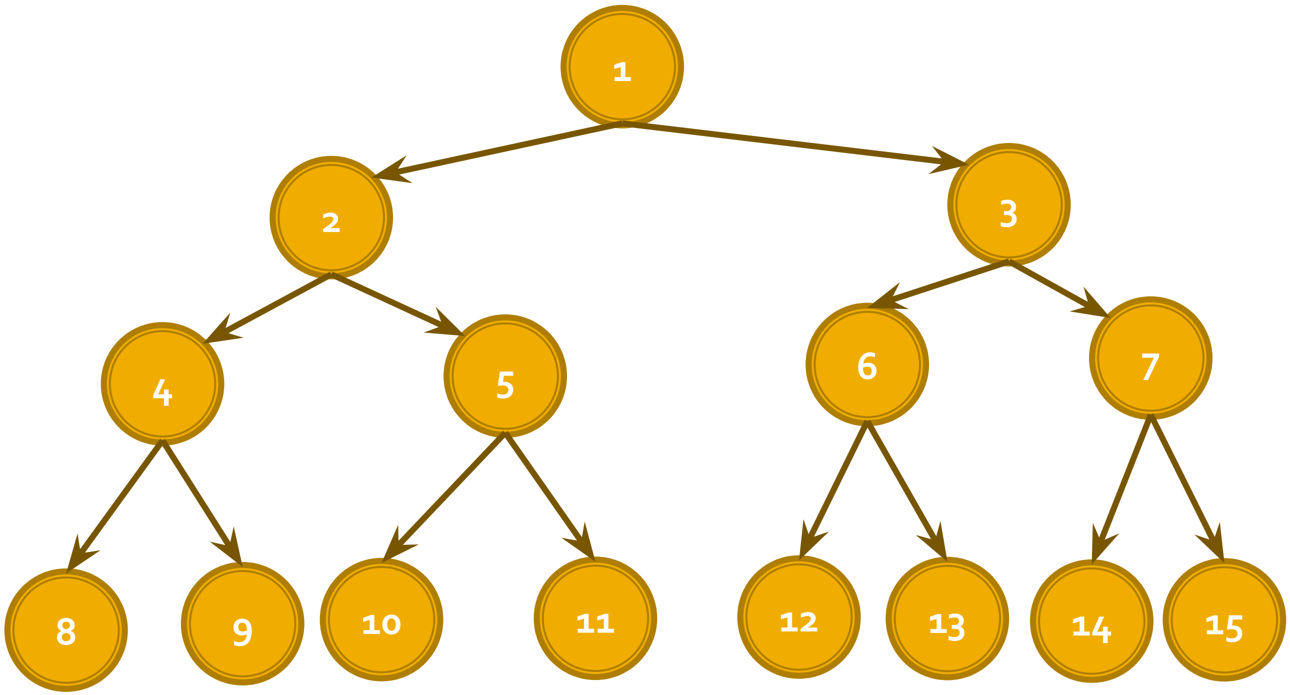
левый сын вершины *i* — *T[2 * i - 1]*;

правый сын вершины *i* — *T[2 * i]*.

Отец вершины в позиции *i > 0*

расположен в позиции *[i / 2] - 1*.

Пример



T == {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}

Обходы деревьев

Обход дерева — способ исследования вершин дерева, при котором каждая вершина проходится только один раз.

Виды:

1. В глубину.
2. В ширину.

Обходы деревьев в глубину

Пусть T — дерево с корнем r, а $v_1, v_2, \ldots, v_n$ — сыновья r.

Прямой (префиксный) обход:

1. Посетить корень r.
2. Посетить в прямом порядке поддеревья с корнями $v_1, v_2, \ldots, v_n$.

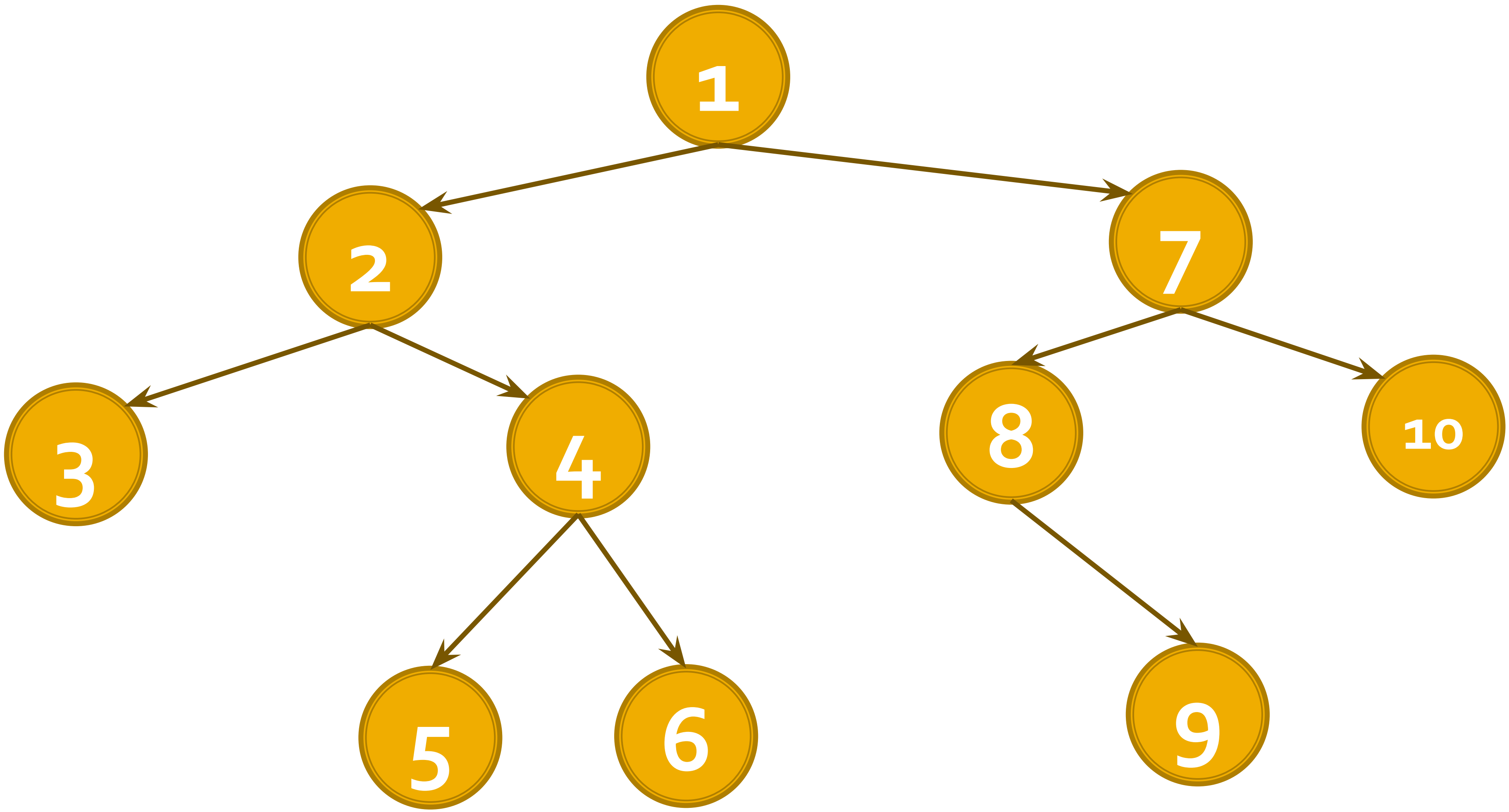
Обратный (постфиксный) обход:

1. Посетить в обратном порядке поддеревья с корнями $v_1, v_2, \ldots, v_n$.
2. Посетить корень r.

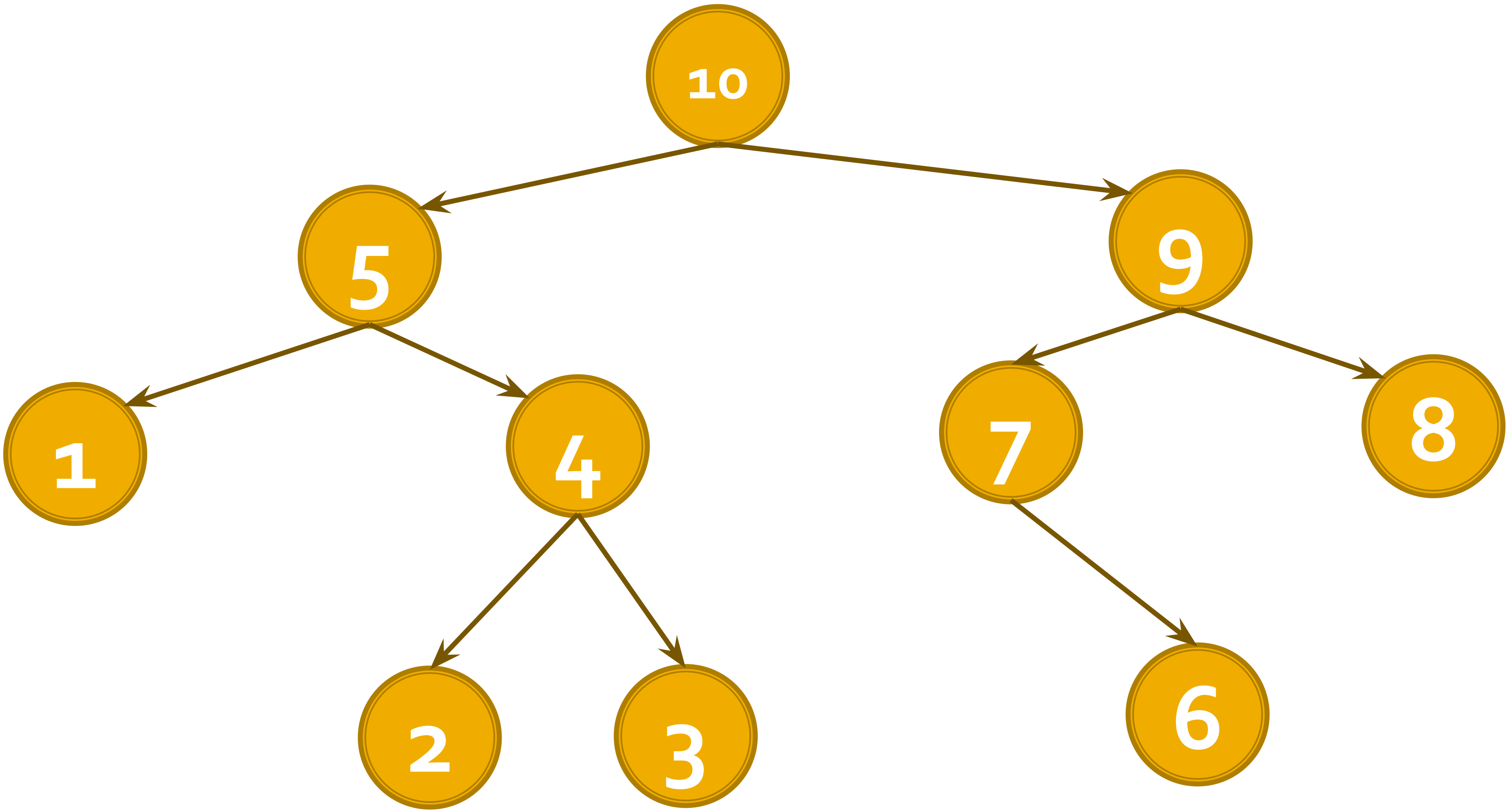
Внутренний (инфиксный) обход:

1. Посетить во внутреннем порядке левое поддерево корня r.
2. Посетить корень r.
3. Посетить во внутреннем порядке правое поддерево корня r.

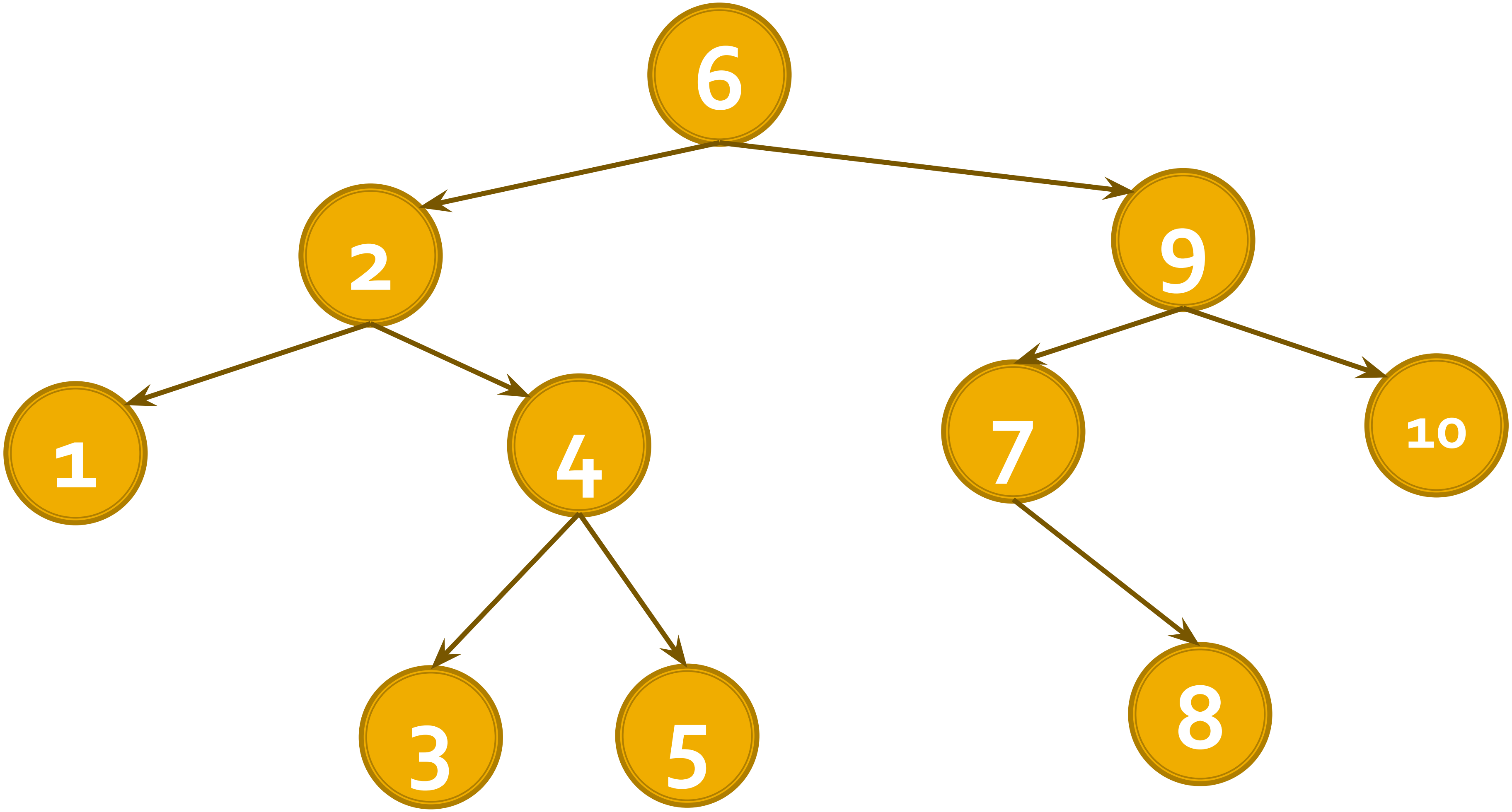
Пример: прямой обход



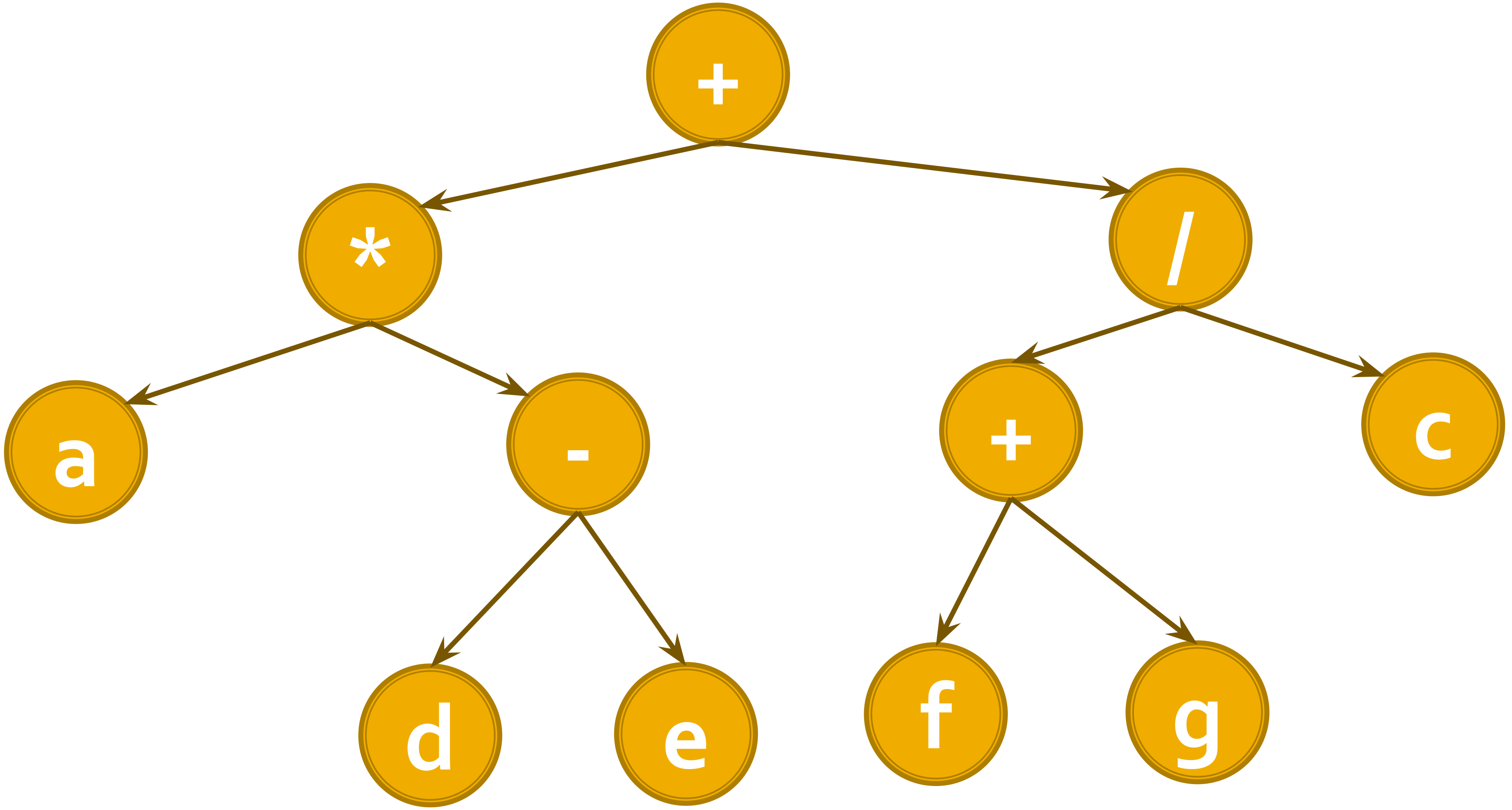
Пример: обратный обход



Пример: внутренний обход



Пример



Пример

Префиксный: *+ * a - d e / + f g c*
Постфиксный: *a d e - * f g + c / +*
Инфиксный: *a * (d - e) + (f + g) / c*

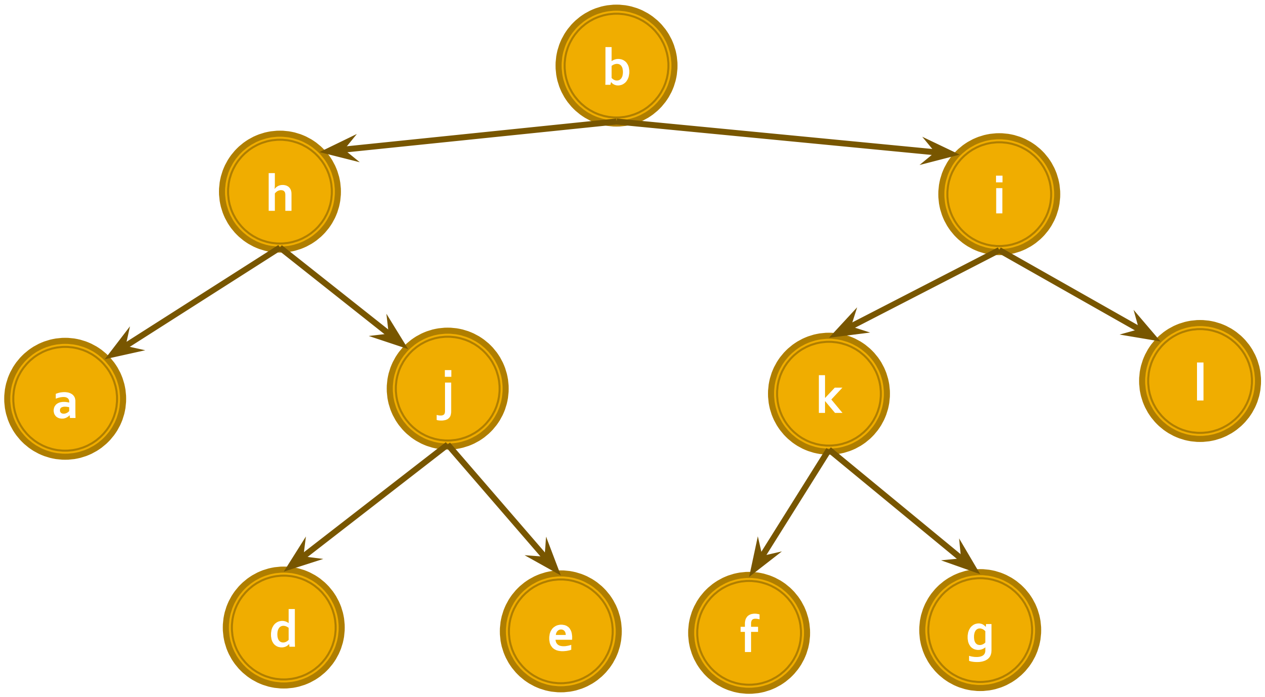
Обход в ширину

Это обход по уровням, начиная от корня, слева направо (или справа налево).

Алгоритм:

0. Поместить в очередь корень.
1. Взять из очереди очередную вершину. Поместить в очередь всех её сыновей слева направо (справа налево).
2. Если очередь пуста — конец. Иначе на шаг 1.

Пример



b

h i

i a j

a j k l

j k l

k l d e

l d e f g

d e f g

e f g

f g

g

Представления деревьев

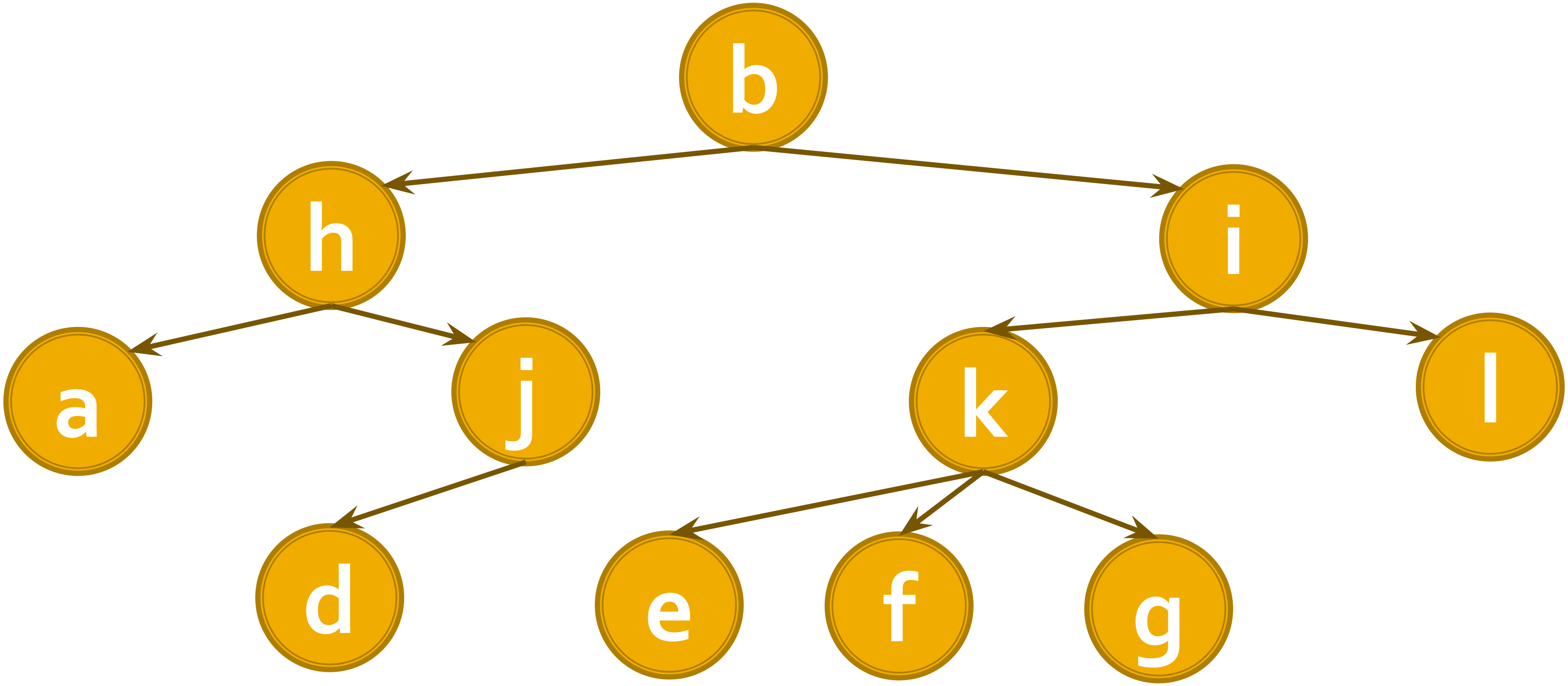
Левое скобочное представление $Lrep(T)$:

1. Если a — корень T с поддеревьями $T_1, T_2, \ldots, T_n$, корни которых — сыновья a, то $Lrep(T) = a(Lrep(T_1), Lrep(T_2), \ldots, Lrep(T_n))$.
2. Если у a сыновей нет, то $Lrep(T) = a$.

Правое скобочное представление $Rrep(T)$:

1. Если a — корень T с поддеревьями $T_1, T_2, \ldots, T_n$, корни которых — сыновья a, то $Rrep(T) = (Rrep(T_1), Rrep(T_2), \ldots, Rrep(T_n))a$.
2. Если у a сыновей нет, то $Rrep(T) = a$.

Пример



Lrep(T) = b(h(a, j(d)), i(k(e, f, g), l))

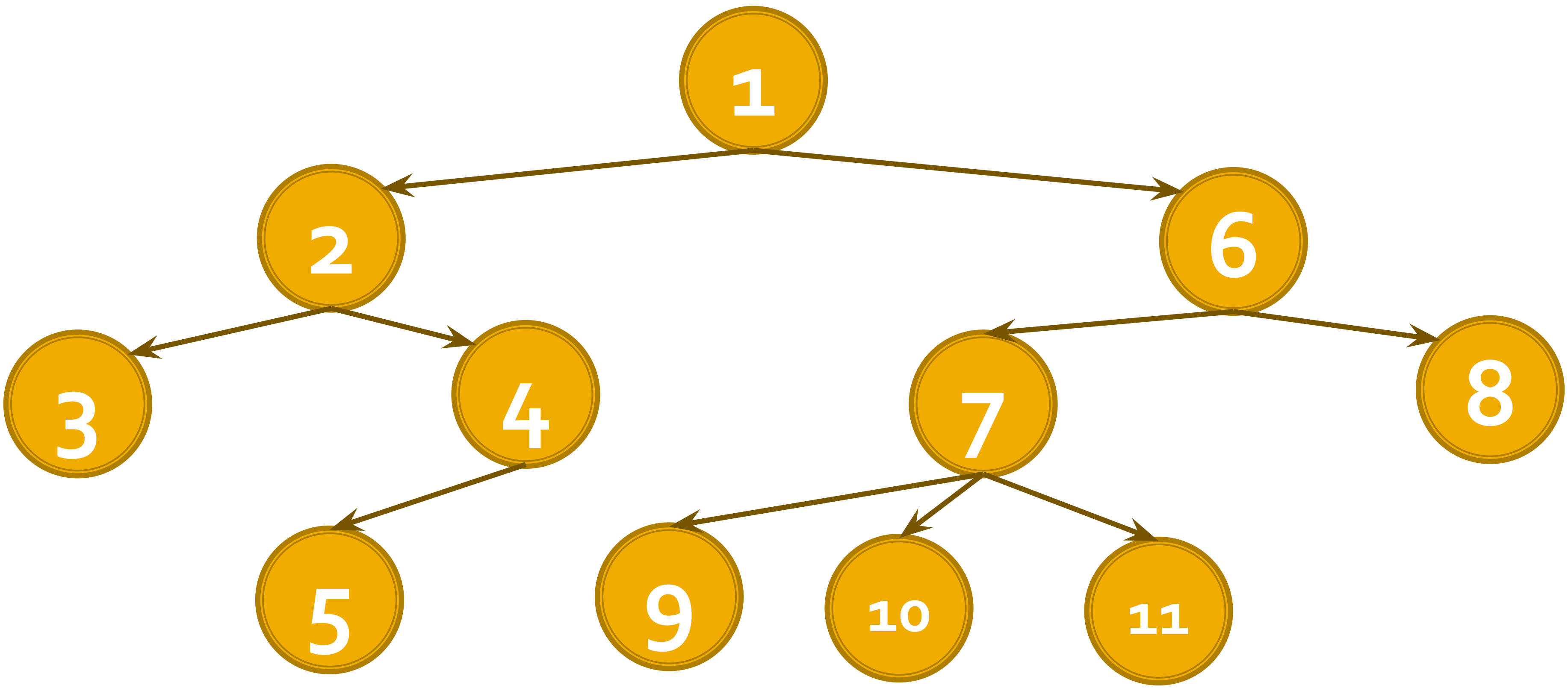
Rrep(T) = ((a, (d)j)h, ((e, f, g)k, l)i)b

Представления деревьев

Список прямых предков:

Составляется список прямых предков для вершин дерева с номерами *1, 2, ..., n*. Для корня полагаем, что предок имеет номер *0*.

Пример



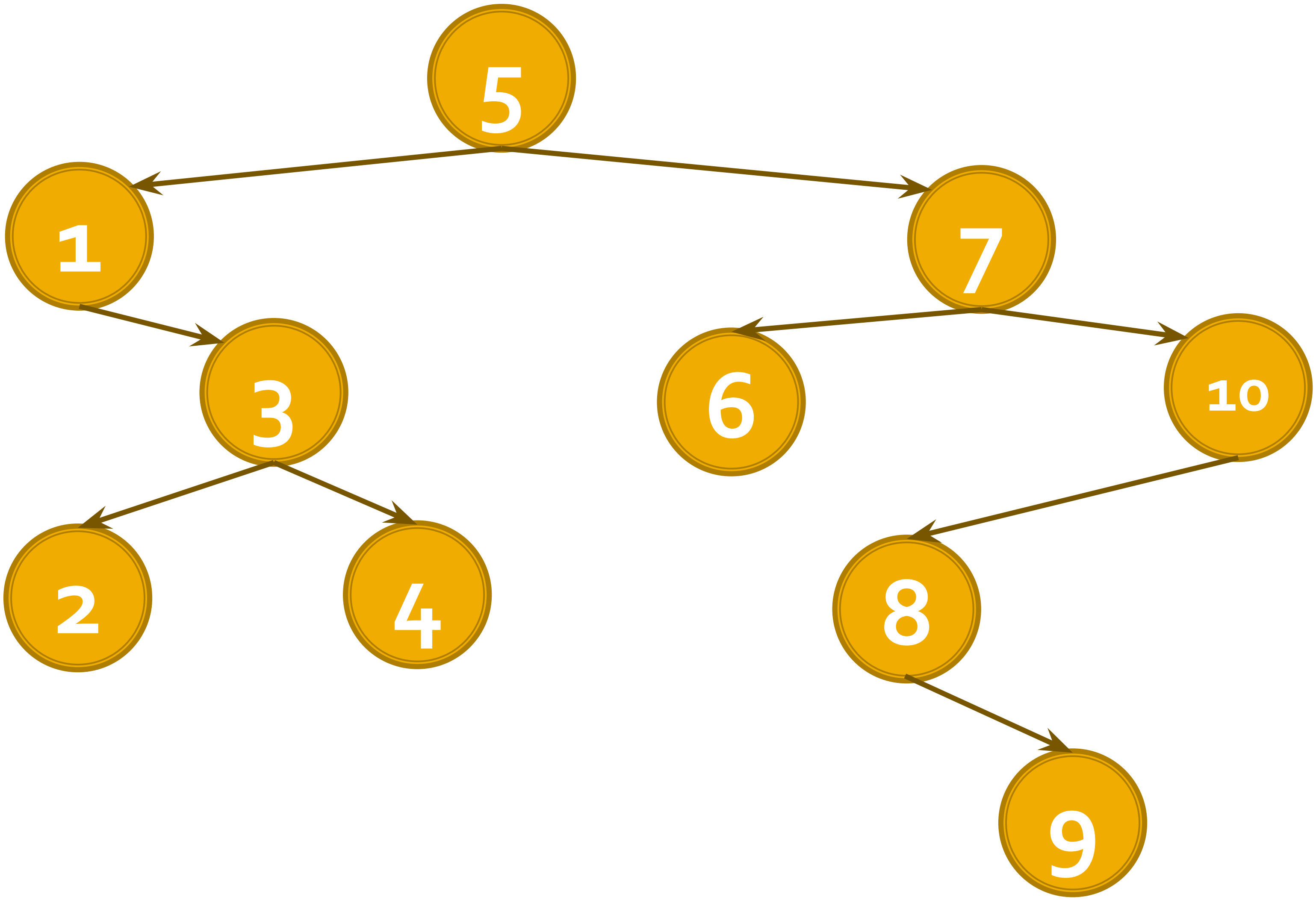
0 1 2 2 4 1 6 6 7 7 7

Дерево двоичного поиска

Деревом двоичного поиска для множества S называется помеченное двоичное дерево, каждая вершина v которого помечена элементом $l(v) \in S$:

1. $l(u) < l(v)$ для всех вершин u из левого поддерева вершины v.
2. $l(u) > l(v)$ для всех вершин u из правого поддерева вершины v.
3. $\forall a \in S\ \exists! v: l(v) = a$.

Пример



Просмотр дерева двоичного поиска

Вход: дерево *T* двоичного поиска для множества *S*, элемент *a*.
Выход: истина, если $a \in S$, ложь — иначе.
Метод: Если $T = \emptyset$, то выдать ложь, иначе выдать *ПОИСК(a, r)*, где *r* — корень дерева *T*.

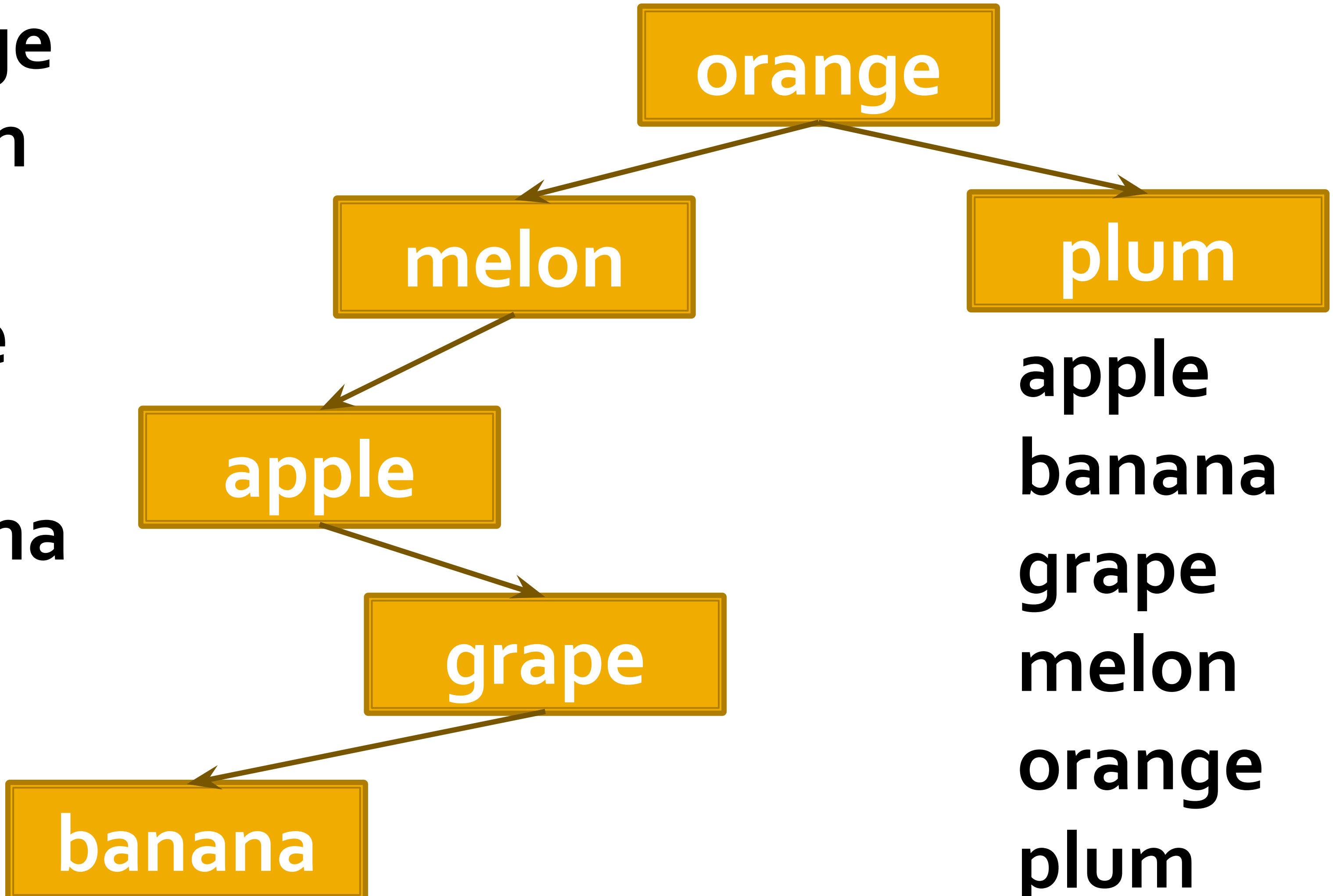
```
функция ПОИСК(a, v)
{
    если a = l(v) то выдать истина
    иначе
        если a < l(v) то
            если v имеет левого сына w
            то выдать ПОИСК(a, w)
            иначе выдать ложь
        иначе
            если v имеет правого сына w
            то выдать ПОИСК(a, w)
            иначе выдать ложь
}
```

Построение дерева двоичного поиска

Вход: последовательность слов произвольной длины.
Выход: введённые слова в лексикографическом порядке.
Метод: каждое слово при вводе помещается в вершину дерева двоичного поиска. По окончании ввода дерево обходится в инфиксном порядке и слова выводятся.

Пример

orange
melon
apple
grape
plum
banana



apple
banana
grape
melon
orange
plum