

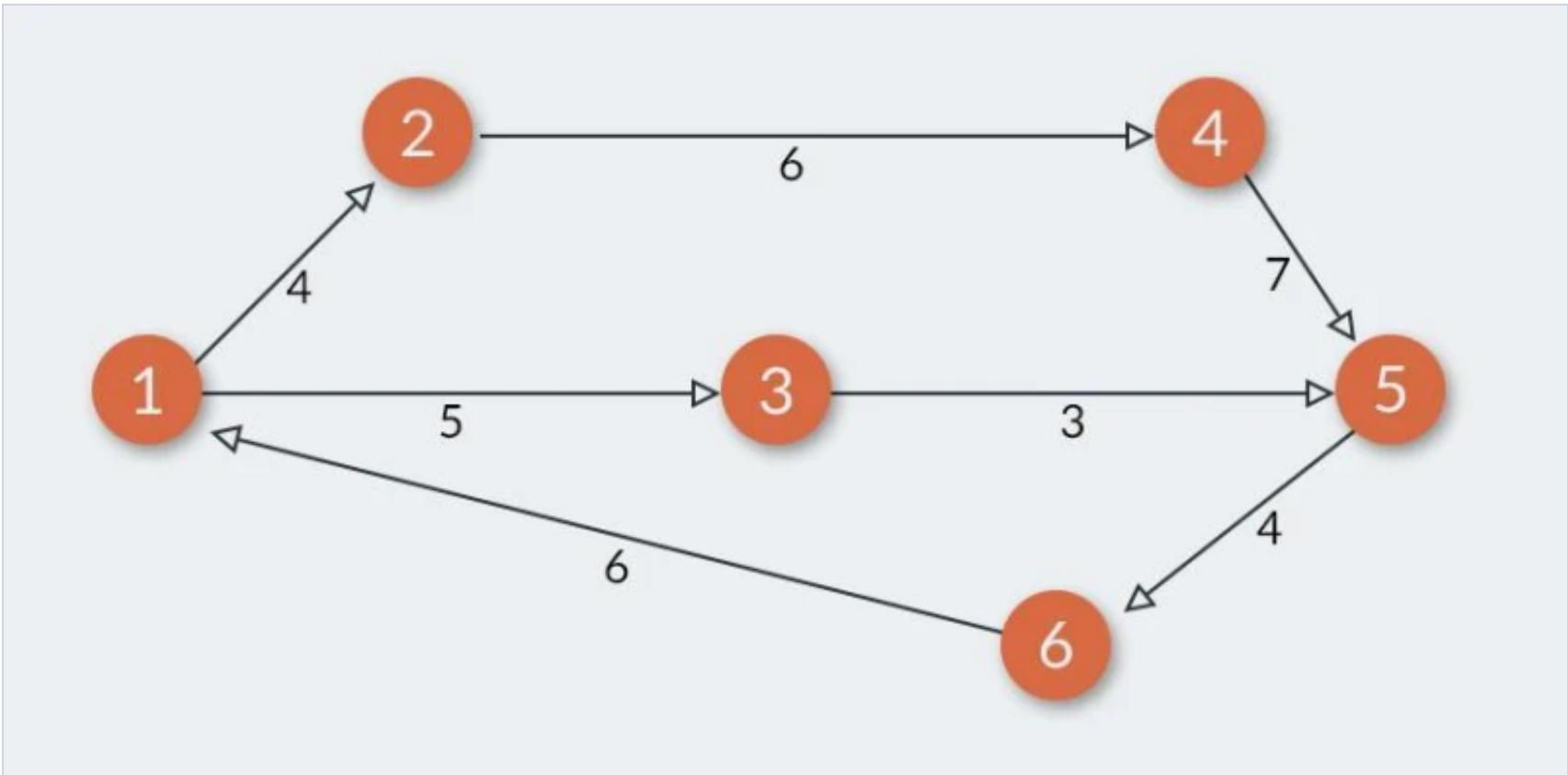
# Деревья, красно-черные деревья

Java Collections  
6 уровень, 7 лекция

ОТКРЫТА

- Привет, Амиго!
- Здорово, Риша!
- Нашел тут свои старые записи и приготовил для тебя немного интересного материала. Думаю, тебе будет интересно послушать.
- Давай. Ты всегда находишь что-то интересное, которое потом становится очень полезным.
- Ладно. Сегодня я хочу тебе рассказать про **деревья**, поэтому начну я с **графов**.

**Граф** – это система, состоящая из точек и линий, которые их соединяют. Точки называются вершинами графа, а линии – ребрами графа. Пример:



Граф очень удобно использовать как математическую модель для различных реальных процессов и задач. Для графов придумано очень много различных задач и алгоритмов, поэтому их довольно часто используют.

Например, вершины – это города, а ребра – это дороги. Тогда поиск самой короткой дороги между городами превращается в задачу «дан граф, найти кратчайший путь между двумя вершинами».

Но не всегда путь из А в Б, занимает столько же, как и путь из Б в А. Поэтому иногда желательно бы иметь две различные линии. Для этого линии (ребра графа) заменяют на стрелки. Т.е. граф может содержать две стрелки: одну из А в Б, а вторую из Б в А.

Если в графе используются стрелки, его называют **ориентированным графом**, если просто линии – **неориентированным графом**.

У каждой вершины может быть свое количество ребер. Также вершина может не иметь ребер вообще. Или наоборот, быть соединена ребрами со всеми остальными вершинами. Если в графе каждая вершина соединена ребром с каждой – такой граф называют **полным**.

Если в графе по ребрам можно добраться до любой вершины, такой граф называют **связным**. Граф состоящий из трех

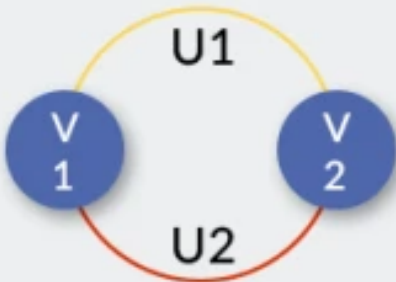
НАЧАТЬ ОБУЧЕНИЕ

# Некоторые типы графов

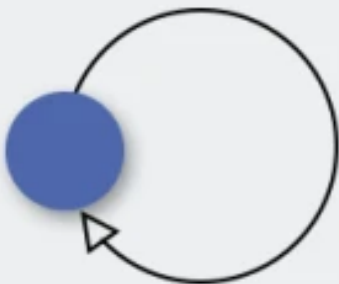
Нуль-граф (граф без рёбер)



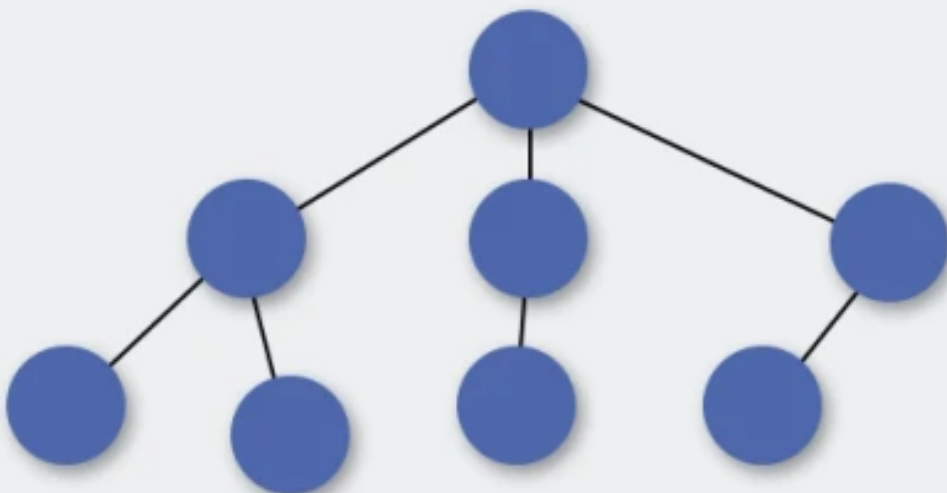
Мультиграф (граф, содержащий кратные рёбра).  
**U1, U2** — кратные рёбра или мультирёбра



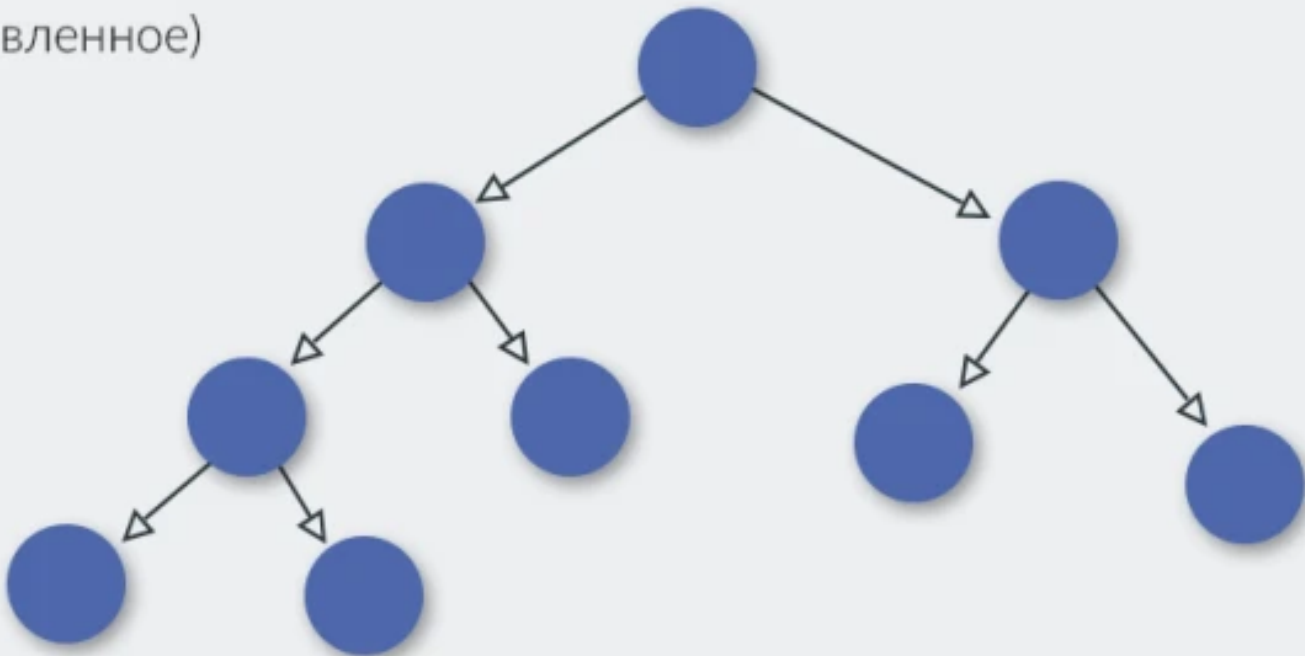
Петля



Дерево — связный граф без циклов, то есть без петель и кратных рёбер

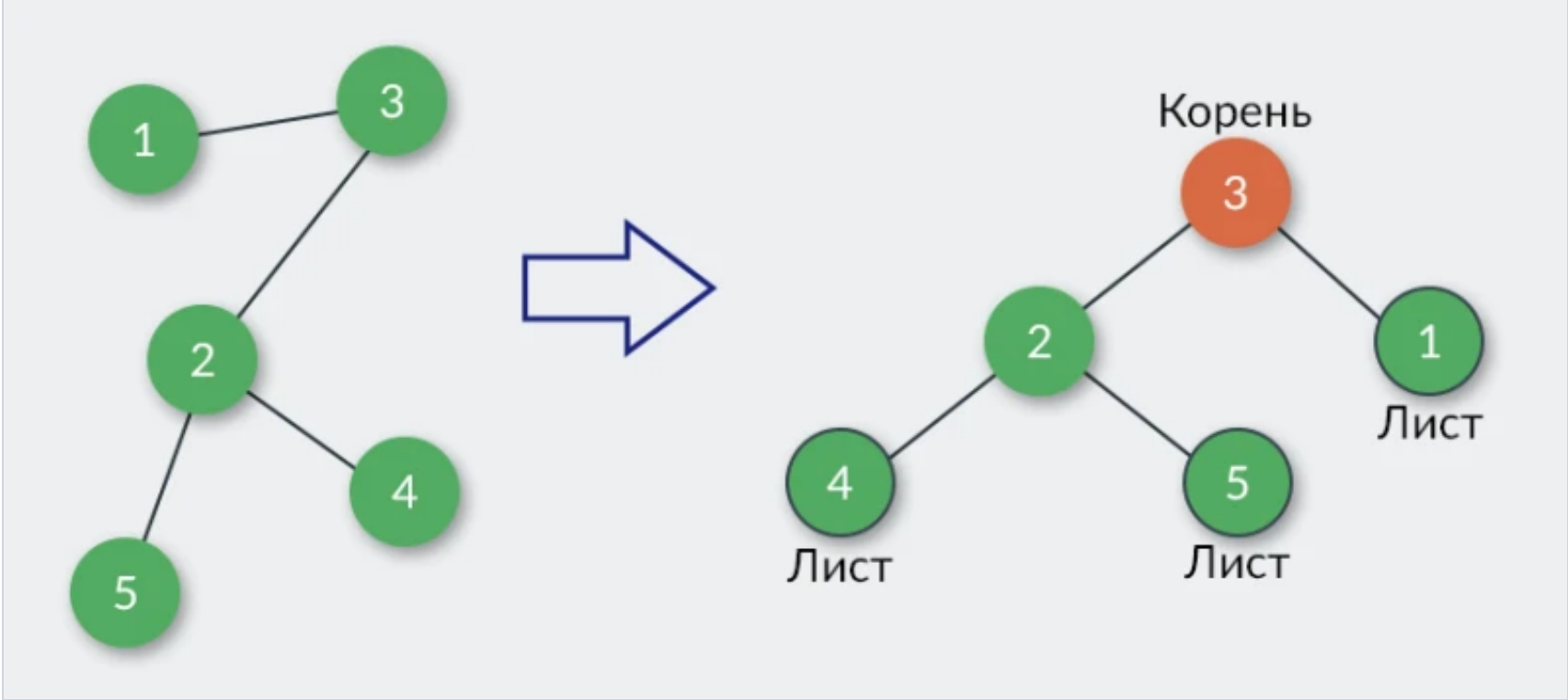


Ориентированное (направленное) дерево



Чтобы соединить в связный граф N вершин, надо минимум N-1 ребер. Такой граф называется деревом.

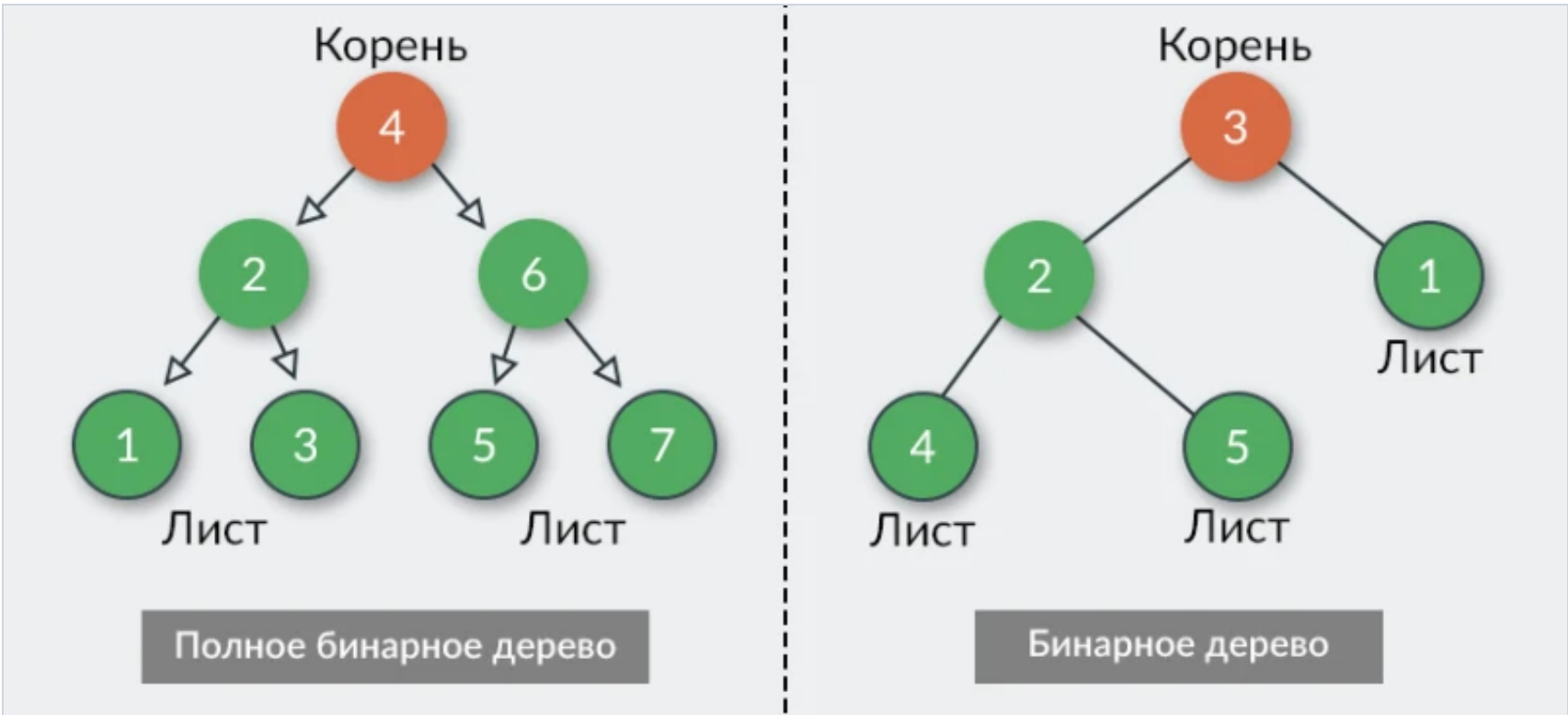
При этом обычно одну вершину выбирают **корнем** дерева, а остальные объявляют ее ветвями. Ветви дерева, которые не имеют своих ветвей, называют **листьями**. Пример:



**Дерево называют бинарным**, если у каждого элемента дерева не более двух потомков. Т.е. их может быть 0, 1 или 2. Выше справа как раз **изображено бинарное дерево**.

**Дерево называют полным бинарным деревом**, когда у каждой ветви 2 потомка, а все листья (без потомков) находятся в одном ряду.

Пример:



- А зачем нужны такие деревья?
  - О, деревья применяются много где. Бинарные деревья поиска так вообще являются отсортированной структурой данных.
  - Это как?
  - Да очень просто. В каждой вершине мы храним некоторое значение. А для каждого элемента вводится правило – значение, которое хранится в потомке справа, больше, чем значение в вершине, а значение, которое хранится в потомке слева – меньше чем значение в вершине. Такое упорядочивание позволяет очень быстро находить нужные элементы в дереве.
  - А можно поподробнее.
  - Сортировка элементов дерева обычно выполняется добавлением. Вот, допустим, у нас есть 7 элементов: 13, 5, 4, 16, 8, 11, 10
- Вот как добавляются элементы в такое дерево.

Шаг 1 – добавляем 13

13

Шаг 2 – добавляем 5

13

5

Шаг 3 – добавляем 4

13

5

4

Шаг 4 – добавляем 16

13

5

16

4

Шаг 5 – добавляем 8

13

5

16

4

8

Шаг 6 – добавляем 11

13

5

16

4

8

11

Шаг 7 – добавляем 10

13

5

16

4

8

[НАЧАТЬ ОБУЧЕНИЕ](#)

Если мы ищем, например, число 7 в таком дереве, то поиск будет проходить так:

0) Начинаем с корня.

1а) Число 7 равно 13? Нет

1б) Число 7 больше 13? Нет, тогда идем в левое поддерево.

2а) Число 7 равно 5? Нет.

2б) Число 7 больше 5? Да, тогда идем в правое поддерево.

3а) Число 7 равно 8? Нет

3б) Число 7 больше 8? Нет, тогда идем в левое поддерево.

4а) Левого поддерева нет, значит, числа 7 в дереве нет.

— Ага. Т.е. нам надо проверять только вершины на пути от корня до предполагаемого места нужного числа. Да, это действительно быстро.

— Еще бы, если дерево сбалансировано, то для миллиона элементов понадобится обход всего около 20 вершин.

— Да, согласен, что это не много.

А что значит – сбалансированное дерево?

— Дерево без «перекосов» — без длинных ветвей. Ведь если бы мы подавали элементы при строительстве дерева в уже отсортированном порядке, у нас бы получилось длинное-предлинное дерево, состоящее из одной ветви.

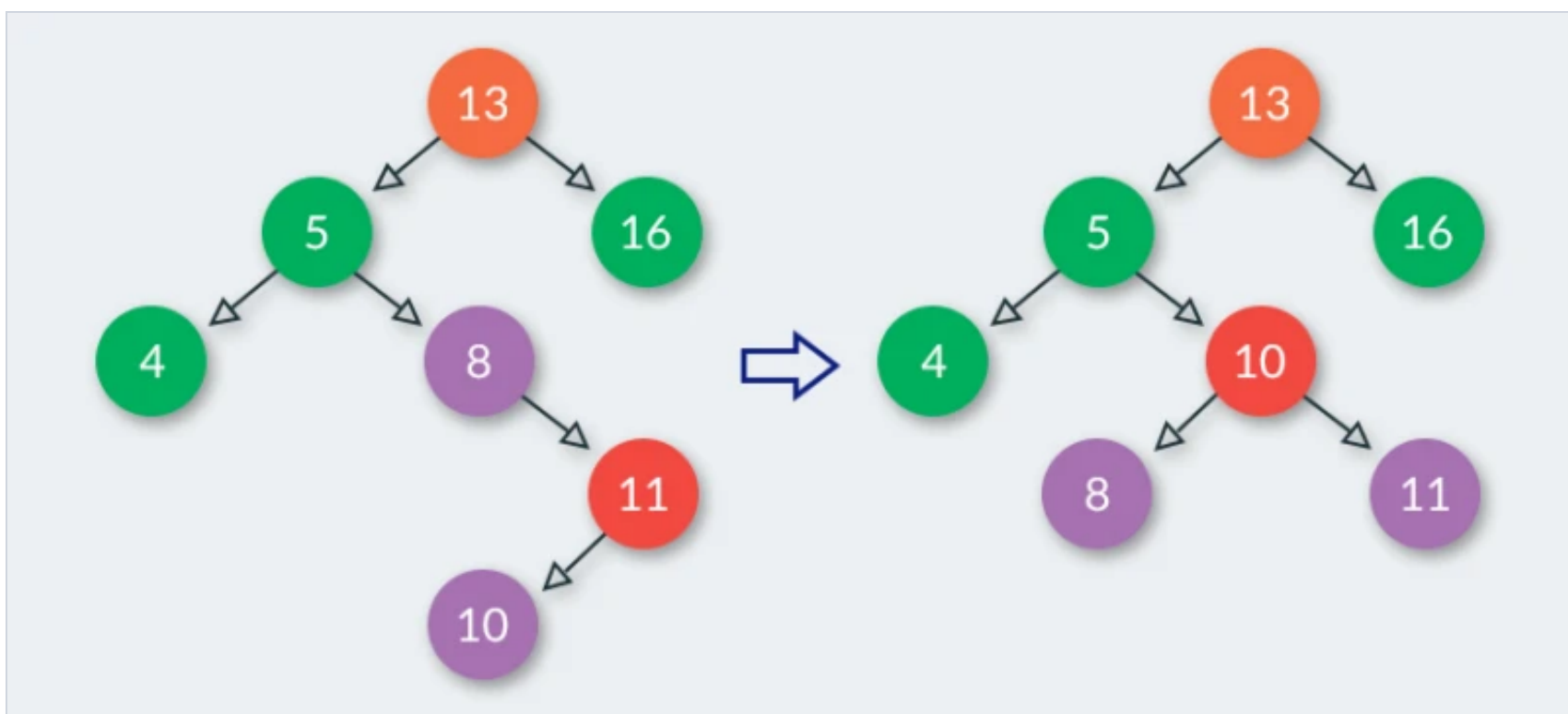
— Гм. Действительно. И как тогда быть?

— Как ты уже, наверное, догадался, самым эффективным будет дерево, которое имеет ветви примерно равной длины. Тогда при каждом сравнении отбрасывается наибольшая часть поддерева из оставшегося.

— Т.е. нужно переделать дерево?

— Ага. Его нужно «сбалансировать» — сделать максимально похожим на полное бинарное дерево.

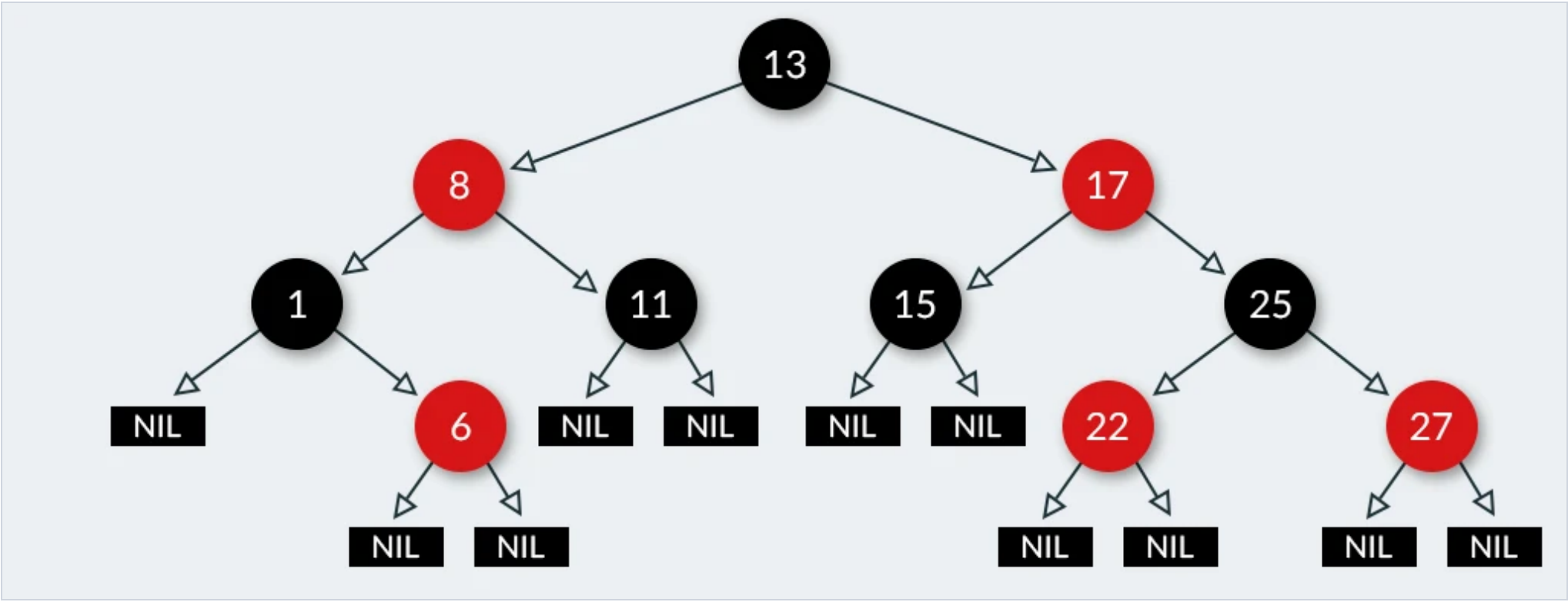
Для решения этой проблемы были придуманы самобалансирующиеся деревья. Когда после добавления элемента в дереве возникает перекос, оно немного меняет порядок элементов, и все становится ок. Пример балансировки:



Одними из таких деревьев есть так называемые «красно-черные деревья».

— Их создатель придумал красить все вершины в два цвета. Один цвет – красный, второй – черный. И разные вершины подчиняются разным правилам. На этом и строится вся балансировка.

Пример:



— А что это за принципы?

- 1) Красная вершина не может быть сыном красной вершины.
- 2) Черная глубина любого листа одинакова (черной глубиной называют количество черных вершин на пути из корня).
- 3) Корень дерева черный.

Я не буду рассказывать тебе, как это работает, у тебя уже небось голова кипит.

— Ага. Процессор греется и не слабо так.

Вот тебе ссылка, если захочешь – почитаешь тут подробнее.

[Ссылка на дополнительный материал](#)

А теперь – иди отдыхай.

< Предыдущая лекция

🌌 x26

– +60 +

НАЧАТЬ ОБУЧЕНИЕ



JavaCoder

Введите текст комментария

KirilX

Уровень 51, Минск, Belarus

17 августа, 14:15

...

как обычно, только затрагиваем важную деталь - на тебе ссылку. Почему тут не расписать?

Ответить

-

+2

+

Method

Уровень 46, Днепр

23 июля, 13:40

...

Красно-черные деревья не назовешь простыми для понимания. (с) Роберт Лафоре.

Ответить

-

0

+

Фарид Гулиев

Уровень 41, Днепр, Украина

7 июля, 14:07

...

Даже хорошо, что эту лекцию дали немного позже, я бы с таким объяснением ничего не понял)

Ответить

-

0

+

Igor Petrashevsky

Уровень 47

26 августа, 00:53

...

педагогика - не их конёк

Ответить

-

0

+

Тимофей

Уровень 51, Москва, Россия

30 марта, 00:02

...

вспомнил недавно сданную дискретную математику

Ответить

-

+1

+

Галкин Юрий

Уровень 41, Москва

17 января, 20:39

...

Ошибка в лекции.

> Дерево называют двоичным (бинарным), если у каждого элемента дерева два потомка. **Т.е. их может быть 0 или 2.** Выше справа как раз изображено бинарное дерево.

> Дерево называют полным бинарным деревом, когда у каждой ветви 2 потомка, а все листья (без потомков) находятся в одном ряду.

Неверно. Их может быть

1

<дерево> ::= ( <данные> <дерево> <дерево> ) | **null** .

**0, 1 или 2.** Иными словами сколько ссылок на потомков не являются null. При текущем же определении в лекции, похоже, что добавлять и убавлять можно только по 2 элемента, чтобы дерево оставалось двоичным, помимо того, что это страшно неудобно и ужасающе странно, более всего важно, что это неверно.. .

Подтверждающая статья вики [двоичное дерево](#)

=====

> — О, деревья применяются много где. Бинарные деревья так вообще являются отсортированной структурой данных.

Это вообще никак не следует из определения. Ну и потому неверно. Можно без большого труда реализовать алгоритм добавления элемента вместо любого свободного (null) указателя на потомка для некоторого случайно выбранного элемента, у которого такое место есть. Сортировки нет, а дерево есть. Данным же свойством обладает [двоичное дерево поиска](#).

=====

> На этом и стоит вся балансировка.

На этом и строится вся балансировка.

Ответить

-

0

+

Pavlo Plynko

Java-разработчик в JavaRush

EXPERT

14 марта, 18:42

...

Спасибо, исправили.

Ответить

-

+2

+

Aleksandr

Уровень 41

10 марта 2021, 10:33

...

О, можно вернуться к первому уровню и может быть доделать большую задачу с деревом, на которую я положил огромную болтяру.

Ответить

-

+3

+

Flexo

Bending Unit #3370318

18 апреля 2021, 18:15

...

НАЧАТЬ ОБУЧЕНИЕ

Ответить

+1

**Mike Starman**    Уровень 40, Киев, Украина    22 декабря 2020, 09:51   

П - последовательность. Сперва большая задача по бинарному дереву и только спустя несколько уровней лекция ;)

Ответить +12

**LuneFox**    инженер по сопровождению в BIFIT    EXPERT    10 февраля, 14:31   

Поговаривают, что создатели JavaRush рассказывают своим детям про алфавит лет через 15 после того, как они научатся читать и писать :)

Ответить +12

**Валерьян**    Уровень 41, Сыктывкар    14 ноября 2020, 19:17   

[Вот неплохой видеоматериал по теме](#)

Ответить +2

**Артур Жуков**    Уровень 32, Астана    27 февраля 2021, 17:22   

Довольно полезное видео для новичков. Но если стало интересно узнать побольше про красно-чёрные деревья, то вам не сюда

Ответить +1

**Konstantin**    Уровень 41, Одесса    30 июня 2020, 11:09   

Поверхностно.  
Потом видимо пойдёт десять задач с использованием деревьев.  
При этом нужно нагуглить и вычитать всё самому в интернете.

Ответить +13

**Fermion**    Уровень 41, Киев, Украина    21 мая 2020, 16:00   

Одна из немногих действительно внятных статей.

Ответить +2

Показать еще комментарии

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

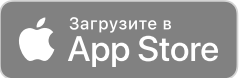
ЯЗЫК ИНТЕРФЕЙСА

Русский



НАЧАТЬ ОБУЧЕНИЕ





"Программистами не рождаются" © 2022 JavaRush