

# Arrays

Java Collections  
7 уровень, 7 лекция

ОТКРЫТА

— Привет, Амиго!

Как прошли утренние лекции?

— Да прилично так, я тебе скажу.

Билаабо дал кучу паттернов проектирования, Элли кучу всего по коллекциям. Непростой выдался денек.

— Ничего, зато я тебя сильно ничем нагружать не стану.

Хочу рассказать тебе про два утилитных класса, с которыми ты уже сталкивался.

Это классы **Arrays** и **Collections**. Все их методы статические и предназначены для работы с коллекциями и массивами.

Начну с более простого – **Arrays**. Вот его методы:

Методы		Пояснение
1	List<T> <b>asList</b> (T... a)	Возвращает неизменяемый список, заполненный переданными элементами.
1	<b>int</b> <b>binarySearch</b> ( <b>int</b> [] a, <b>int</b> fromIndex, <b>int</b> toIndex, <b>int</b> key)	Ищет элемент (key) в массиве a или подмассиве, начиная с fromIndex и до toIndex. Массив должен быть отсортирован! Возвращает номер элемента или fromIndex-1, если элемент не найдет.
1	<b>int</b> [] <b>copyOf</b> ( <b>int</b> [] original, <b>int</b> newLength)	Возвращает подмассив original массива, newLength элементов, начиная с нулевого.
1	<b>int</b> [] <b>copyOfRange</b> ( <b>int</b> [] original, <b>int</b> from, <b>int</b> to)	Возвращает под массив original массива, начиная с from и до to.
1	<b>boolean</b> <b>deepEquals</b> (Object[] a1, Object[] a2)	Глубокое сравнение массивов. Массивы считаются равными, если равны их элементы. Если элементы сами являются массивами, для них тоже выполняется глубокое сравнение.

		массивом, для него также вызывается deepHashCode.
1	String <b>deepToString</b> (Object[] a)	Глубокое преобразование к строке. Для всех элементов вызывается toString(). Если элемент является массивом, для него тоже выполняется глубокое преобразование к строке.
1	<b>boolean equals</b> (int[] a, int[] a2)	Сравнивает два массива поэлементно.
1	<b>void fill</b> (int[] a, int fromIndex, int toIndex, int val)	Заполняет массив (или подмассив) заданным значением.
1	<b>int hashCode</b> (int a[])	Вычисляет общий hashCode всех элементов массива.
1	<b>void sort</b> (int[] a, int fromIndex, int toIndex)	Сортирует массив (или подмассив) по возрастанию.
1	String <b>toString</b> (int[] a)	Преобразовывает массив к строке. Для всех элементов вызывается toString();

— А что, очень полезные методы, многие мне бы пригодились.

— Хочу еще заметить, что я тут привел не все методы. Почти все методы из таблицы дублируются для всех примитивных типов. Т.е. если написано String toString(int[] a), то также есть методы String toString(**boolean**[] a), String toString(**byte**[] a), String toString(**long**[] a), String toString(**float**[] a), String toString(**double**[] a), String toString(**char**[] a).

— Так это же меняет дело. Теперь это просто незаменимый класс.

— Рад, что тебе понравилось. Тогда продолжим после перерыва.

 +39 

Комментарии (20)

популярныеновыестарые

JavaCoder

Введите текст комментария

Igor Petrashevsky Уровень 4727 августа, 03:25

...

было в первом квесте, снова 27 в топку

НАЧАТЬ ОБУЧЕНИЕ

Anonymous #2413783

Java Developer

17 декабря 2020, 12:28

...

почему-то ранее не обращал на это внимание, может для кого-то будет новой такая деталь:  
[Разница между Arrays.asList \(массив\) и новым ArrayList <Integer> \(Arrays.asList \(массив\)\).](#)

Ответить

+20

LuneFox

инженер по сопровождению в BIFIT

EXPERT

28 февраля, 18:53

...

Каким-то странным языком задан вопрос и написан ответ, очень напоминает машинный перевод.

Ответить

0

ram0973

Уровень 41, Набережные Челны, Россия

19 августа 2020, 08:13

...

Интересно за [преподавателями](#) стоят реальные люди?

Ответить

0

Zlata

Уровень 49, Ольштын, Польша

30 апреля, 14:08

...

да Бобров это [Бранниган](#)

Ответить

0

hidden #1281202

Уровень 41

10 августа 2019, 18:31

...

**Что такое бинарный поиск?**  
Этот вид поиска использует подход «Разделяй и властвуй», требует предварительной сортировки набора данных.  
Алгоритм делит входную коллекцию на равные половины, и с каждой итерацией сравнивает целевой элемент с элементом в середине. Поиск заканчивается при нахождении элемента. Иначе продолжаем искать элемент, разделяя и выбирая соответствующий раздел массива. Целевой элемент сравнивается со средним.  
Вот почему важно иметь отсортированную коллекцию при использовании двоичного поиска.  
**Визуализация бинарного поиска в сравнении с обычным линейным:**

Ответить

+2

Андрей Святогоров

Уровень 38, Москва, Россия

11 августа 2019, 18:32

...

Я одного не понимаю, почему все так любят применять это сраное понятие алгоритм "разделяй и властвуй", неужели это проще и понятнее, чем называть его рекурсивным или разбиением. Не знаю почему, но меня прям коробит, когда используют этот термин. "Разделяй и властвуй" властелины блин. Один переводчик криво перевел, теперь, каждый везде использует этот термин, приехали, мать вашу.

Ответить

+9

Андрей Святогоров

Уровень 38, Москва, Россия

11 августа 2019, 18:43

...

И он не требует предварительной сортировки, это не правильная формулировка. Бинарный поиск требует того, что бы входные данные были отсортированы или должны иметь соответствующую структуру (например, бинарное дерево). Просто, если перед каждым запуском бинарного поиска сортировать входной массив, мы получим менее эффективный алгоритм, чем линейный поиск.  
 $O(n)$  для линейного поиска =  $n$   
 $O(n)$  сортировки (для сортировки слиянием) =  $n \log(n)$   
 $O(n)$  бинарного поиска =  $\log(n)$   
 $O(n)$  бинарного поиска и сортировки слиянием =  $n \log(n) + \log(n)$  (но так как при вычислении верхней границы мы можем пренебречь более низким порядком, можно считать, что бинарный поиск + сортировка имеют  $O(n) = n \log(n)$ , по наибольшему порядку роста)  
  
 $n < n \log(n) \Rightarrow$  линейный поиск эффективнее сортировки + бинарный поиск

Ответить

+3

hidden #1281202

Уровень 41

11 августа 2019, 19:10

...

Походу этот переводчик криво перевёл: [wiki](#)  
Если в вики нажать на english версию, то оригинальное название алгоритма выглядит как: "Divide-and-conquer". Интересно, каков будет наиболее точный дословный перевод на русский язык? Что бы узнать, прошу перейти назад на русскую версию вики.  
  
Насчёт бинарного поиска.  
И так известно, что если мы знаем, что наши данные отсортированы, то не нужно запускать сортировку, так как это не имеет смысла и никакой идиот не будет этим заниматься - это очевидно.

Ответить

+1

Андрей Святогоров

Уровень 38, Москва, Россия

11 августа 2019, 19:29

...

а что делать если входные данные не отсортированные или мы не знаем, что отсортированы они или нет?

Ответить

+2

Андрей Святогоров

Уровень 38, Москва, Россия

11 августа 2019, 19:32

...

Тут дело не в очевидности, а в том, какие ограничения и как накладываются на входные данные. Если данные не отсортированы, то либо эффективнее применить линейный поиск, либо изменить структуру входных данных. Но ни как не сортировать предварительно дабы применить

А про разделяй и властвуй, это так, меня просто очень напрягает этот термин, не знаю почему. А на счет того, что в английской версии используется "Divide-and-conquer", так далеко не всегда дословно переводить правильно. Переведи любой абзац английского текста дословно, любого, получится ересь. Но в любом случае, это разговор в пользу бедных и просто мое субъективное мнение на использование термина. Кто бы какой термин не использовал, Разделяй и властвуй или разбиение или рекурсивный, собеседник вполне поймет о чем речь)))

Ответить

0

+

Это все Цезарь виноват. Он пропиарил "разделяй и властвуй". Не умер бы Атилла так рано было бы "порви и трахни". Эхехх...

Ответить

+6

+

метод половинного деления он звался в 60х. правда для обычных задач на множествах

Ответить

0

+

Не понятна разница между deepEquals, deepToString и обычными методами. А там и там для каждого элемента массива..

Ответить

0

+

в том, что toString вложенные массивы не преобразует к строке, а deepToString преобразует и вложенные массивы.

```
1  int[][] matr = {{1,2,3},{1,2,3},{1,2,3}};
2  System.out.println(Arrays.toString(matr));
3  System.out.println(Arrays.deepToString(matr));
```

ВЫВОД

```
1  [[I@3feba861, [I@5b480cf9, [I@6f496d9f]
2  [[1, 2, 3], [1, 2, 3], [1, 2, 3]]
```

Ответить

+13

+

В binarySearch ключевая фраза "Массив должен быть отсортирован!"  
А то крутишь [массив](#) и не понимаешь: JVM с ума сошла или сам тронулся: не обнаруживает значения, хотя они там точно есть. А сортировочку-то не сделал!

Ответить

+8

+

Вообще логично, поскольку двоичный поиск предполагает, что нам на вход подают отсортированный массив - там потому и сложность оценивается в [O\(logN\)](#), а не O(N), как в линейном (где надо прочесывать с самого начала). Если кто еще не смотрел - [CS50 первая лекция](#), с 17-й минуты примерно.

Ответить

+3

+

Метод ListasList(T... a) превращается в asList(T... a).

Ответить

+4

+

В binarySearch не всё так просто. Если элемент не найден, возвращает величину (-i-1), где i — это индекс элемента, после которого нужно вставить искомый элемент для того, чтобы сохранился порядок сортировки. Отрицание для того, что бы если элемент не найден, результат был меньше нуля.

Ответить

+18

+

- [Курс Java](#)  
[Помощь по задачам](#)  
[Подписки](#)  
[Задачи-игры](#)
- [Статьи](#)  
[Форум](#)  
[Чат](#)  
[Истории успеха](#)  
[Активности](#)
- [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

▼

