



数据结构和算法 (Python描述)

郭 炜

微信公众号



微博: <http://weibo.com/guoweiofpku>

学会程序和算法，走遍天下都不怕!

讲义照片均为郭炜拍摄



二分算法



北京大学
PEKING UNIVERSITY

信息科学技术学院

北京大学信息学院 郭炜

二分查找



甘肃张掖平山湖大峡谷

二分查找

●A心里想一个1-1000之间的数，B来猜，可以问问题，A只能回答是或否。怎么猜才能问的问题次数最少？

是1吗？是2吗？.....是999吗？ 平均要问500次

大于500吗？大于750吗？大于625吗？每次缩小猜测范围到上次的一半，只需要 10次

二分查找函数

- 写一个函数BinarySeach, 在从小到大排序的列表a里查找元素p,如果找到, 则返回元素下标, 如果找不到, 则返回None。要求复杂度 $O(\log(n))$

```
def binarySearch(a,p,key = lambda x : x):  
    L,R = 0,len(a)-1    #查找区间的左右端点, 区间含右端点  
    while L <= R: #如果查找区间不为空就继续查找  
        mid = L+(R-L)//2 #取查找区间正中元素的下标  
        if key(p) < key(a[mid]):  
            R = mid - 1    #设置新的查找区间的右端点  
        elif key(a[mid]) < key(p):  
            L = mid + 1    # 设置新的查找区间的左端点  
        else:  
            return mid  
    return None
```

复杂度 $O(\log(n))$

二分查找函数

- 写一个函数BinarySeach, 在从小到大排序的列表a里查找元素p,如果找到, 则返回元素下标, 如果找不到, 则返回None。要求复杂度 $O(\log(n))$

```
a = [9, 12, 27, 33, 33, 41, 80]    #a有序
print(binarySearch(a,33))    #>>3
print(binarySearch(a,57))    #>>None
a.sort(key = lambda x: x % 10)    #按个位数从小到大排序
print(a)    #>>[80, 41, 12, 33, 33, 27, 9]
print(binarySearch(a,57,key = lambda x: x %10))    #>>5
```

●写一个函数lowerBound，在从小到大排序的列表a里查找比给定元素p小的，下标最大的元素。找到则返回其下标，找不到则返回None

```
def lowerBound(a,p,key = lambda x : x):    #复杂度 $O(\log(n))$   
    #找小于p的最靠右元素的下标  
    L,R = 0,len(a)-1  
    result = None  
    while L <= R:  
        mid = L+(R-L)//2  
        if key(a[mid]) < key(p):  
            L = mid + 1  
            result = mid  
        else:  
            R = mid - 1  
    return result
```

●写一个函数lowerBound，在从小到大排序的列表a里查找比给定元素p小的，下标最大的元素。找到则返回其下标，找不到则返回None

```
a = [9, 12, 27, 33, 33, 41, 80]
print(lowerBound(a,33)) #>>2
print(lowerBound(a,50)) #>>5
print(lowerBound(a,0)) #>>None
a.sort(key = lambda x: x % 10)
print(a) #>>[80, 41, 12, 33, 33, 27, 9]
print(lowerBound(a,28,key = lambda x: x %10)) #>>5
print(lowerBound(a,13,key = lambda x: x %10)) #>>2
```


- 查找区间起点L,终点R, 循环不可写成:

```
while L < R:
```

```
.....
```

应该是 while L \leq R:

- 设置查找区间新端点时, 新端点一定要比原区间中点大或者小, 不要等于原中点。



北京大学
PEKING UNIVERSITY

信息科学技术学院

北京大学信息学院 郭炜

二分法求方程的根



祁连山风光

二分法求方程的根

求下面方程的一个根: $f(x) = x^3 - 5x^2 + 10x - 80 = 0$

若求出的根是 a , 则要求 $|f(a)| \leq 10^{-6}$

●解法: 对 $f(x)$ 求导, 得 $f'(x) = 3x^2 - 10x + 10$ 。由一元二次方程求根公式知方程 $f'(x) = 0$ 无解, 因此 $f'(x)$ 恒大于0。故 $f(x)$ 是单调递增的。易知 $f(0) < 0$ 且 $f(100) > 0$, 所以区间 $[0, 100]$ 内必然有且只有一个根。由于 $f(x)$ 在 $[0, 100]$ 内是单调的, 所以可以用二分的办法在区间 $[0, 100]$ 中寻找根。



北京大学
PEKING UNIVERSITY

信息科学技术学院

北京大学信息学院 郭炜

例题 寻找指定和的整数对



青海湖

例题 寻找指定和的整数对

输入 n ($n \leq 100,000$)个整数，找出其中的两个数，它们之和等于整数 m (假定肯定有解)。题中所有整数都能用 `int` 表示

例题：寻找指定和的整数对

输入 n ($n \leq 100,000$)个整数，找出其中的两个数，它们之和等于整数 m (假定肯定有解)。题中所有整数都能用 `int` 表示

解法1：用两重循环，枚举所有的取数方法，复杂度是 $O(n^2)$ 的。

```
for(int i = 0; i < n-1; ++i)
    for(int j = i + 1; j < n; ++j)
        if( a[i]+a[j] == m)
            break;
```

$100,000^2 = 100$ 亿，在各种OJ上提交或参加各种程序设计竞赛，这样的复杂度都会超时！

例题：寻找指定和的整数对

输入 n ($n \leq 100,000$)个整数，找出其中的两个数，它们之和等于整数 m (假定肯定有解)。题中所有整数都能用 `int` 表示

解法2：

- 1) 将数组排序，复杂度是 $O(n \times \log(n))$
- 2) 对数组中的每个元素 $a[i]$ ，在数组中二分查找 $m - a[i]$ ，看能否找到。复杂度 $\log(n)$ ，最坏要查找 $n-2$ 次，所以查找这部分的复杂度也是 $O(n \times \log(n))$

这种解法总的复杂度是 $O(n \times \log(n))$ 的。

例题：寻找指定和的整数对

输入 n ($n \leq 100,000$)个整数，找出其中的两个数，它们之和等于整数 m (假定肯定有解)。题中所有整数都能用 `int` 表示

解法3：

- 1) 将数组排序，复杂度是 $O(n \times \log(n))$
- 2) 查找的时候，设置两个变量 i 和 j , i 初值是0, j 初值是 $n-1$. 看 $a[i] + a[j]$, 如果大于 m , 就让 j 减1, 如果小于 m , 就让 i 加1, 直至 $a[i] + a[j] = m$ 。

这种解法总的复杂度是 $O(n \times \log(n))$ 的。



北京大学
PEKING UNIVERSITY

信息科学技术学院

北京大学信息学院 郭炜

例题

Aggressive cows



航拍青海湖

例题2 百练 2456: Aggressive cows

<http://bailian.openjudge.cn/practice/2456>

农夫 John 建造了一座很长的畜栏，它包括 N ($2 \leq N \leq 100,000$) 个隔间，这些小隔间的位置为 x_0, \dots, x_{N-1} ($0 \leq x_i \leq 1,000,000,000$, 均为整数, 各不相同)。

John 的 C ($2 \leq C \leq N$) 头牛每头分到一个隔间。牛都希望互相离得远点省得互相打扰。怎样才能使**最近的两头牛之间的距离**尽可能的大，这个最大距离是多少呢？

例题2

●解法1:

先得到排序后的隔间坐标 x_0, \dots, x_{N-1}

从 $1,000,000,000/(C-1)$ 到 1 依次尝试这个 “最大的距离” D ，找到的第一个可行的就是答案。

尝试 D 是否可行的方法:

- 1) 第1头牛放在 x_0
- 2) 若第 k 头牛放在 x_i ，则找到 x_{i+1} 到 x_{N-1} 中第一个坐标位于 $[x_i + D, x_{N-1}]$ 中的 x_j ，第 $k+1$ 头牛放在 x_j 。找不到这样的 x_j ，则 D 不可行

若所有牛都能放下，则 D 即答案

例题2

●解法1:

先得到排序后的隔间坐标 x_0, \dots, x_{N-1}

从 $1,000,000,000/(C-1)$ 到 1 依次尝试这个“最大的最近距离” D ，找到的第一个可行的就是答案。

尝试 D 是否可行的方法:

- 1) 第1头牛放在 x_0
- 2) 若第 k 头牛放在 x_i ，则找到 x_{i+1} 到 x_{N-1} 中第一个坐标位于 $[x_i + D, x_{N-1}]$ 中的 x_j ，第 $k+1$ 头牛放在 x_j 。找不到这样的 x_j ，则 D 不可行

若所有牛都能放下，则 D 即答案

复杂度 $1,000,000,000/(C-1) * N$ ，即 $1,000,000,000$ ，超时!

例题2

●解法2:

先得到排序后的隔间坐标 x_0, \dots, x_{N-1}

在 $[L, R]$ 内用二分法尝试“最大最近距离” $D = (L+R)/2$ (L, R 初值为 $[1, 1,000,000,000/(C-1)]$)

若 D 可行, 则记住该 D , 然后在新 $[L, R]$ 中继续尝试(新 $L = D+1$)

若 D 不可行, 则在新 $[L, R]$ 中继续尝试(新 $R = D-1$)

复杂度 $\log(1,000,000,000/(C-1)) * N$

二分法寻找答案的核心思想

二分法找最优答案的核心思想就是：

**如果一个假设的答案成立，那就跳着试一个更优的假设答案看行不行；
如果一个假设的答案不成立，那就跳着试一个更差的假设答案看行不行。
必须每次验证假设答案，都可以把假设答案所在的区间缩小为上次的一半。**

前提：单调性。一个假设答案不成立，则比它更优的假设答案肯定都不成立。