

## 2019/03 数据科学类 OT Hackerank

OT一共三道题，第三题不算分，计时共三小时。

第一题，写一个**function**，根据已有的  $(x, y)$ ，计算新  $x$  的  $y$  值，题目不难，而且 instruction 里，也把特殊情况说的很清楚，所以做起来是很顺利的。第二题，处理数据，首先读取一个**dataframe**，题目有很多个小问，每个小问考察一个小点，\*\*前两个主要是需要熟悉 pandas/numpy 的计算和操作，后几个问是 linear regression 和 mse 的相关计算\*\* 第三题，我没有做 lol...

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=494746&highlight=two%2Bsigma>

## 2019/01 Data Science Intern

我面的是 QR，有三道题，不过最后一道是 bonus，开始吓我一条，还琢磨怎么多一道。第一道编程序，**game of life**。秒过。很简单。[\(https://leetcode.com/problems/game-of-life/\)](https://leetcode.com/problems/game-of-life/) 第二道是数据分析题。

比较恶心的是，要知道怎么读取数据，如果这一关过不来就挂了。我在这上面花了挺长时间，所幸最后弄对了。数据分析倒不难，就是要准备一些数据分析的技巧，如果用 python，pandas，sklearn 肯定要熟的。最后通过了所有的 case。

<https://www.1point3acres.com/bbs/thread-493467-1-1.html>

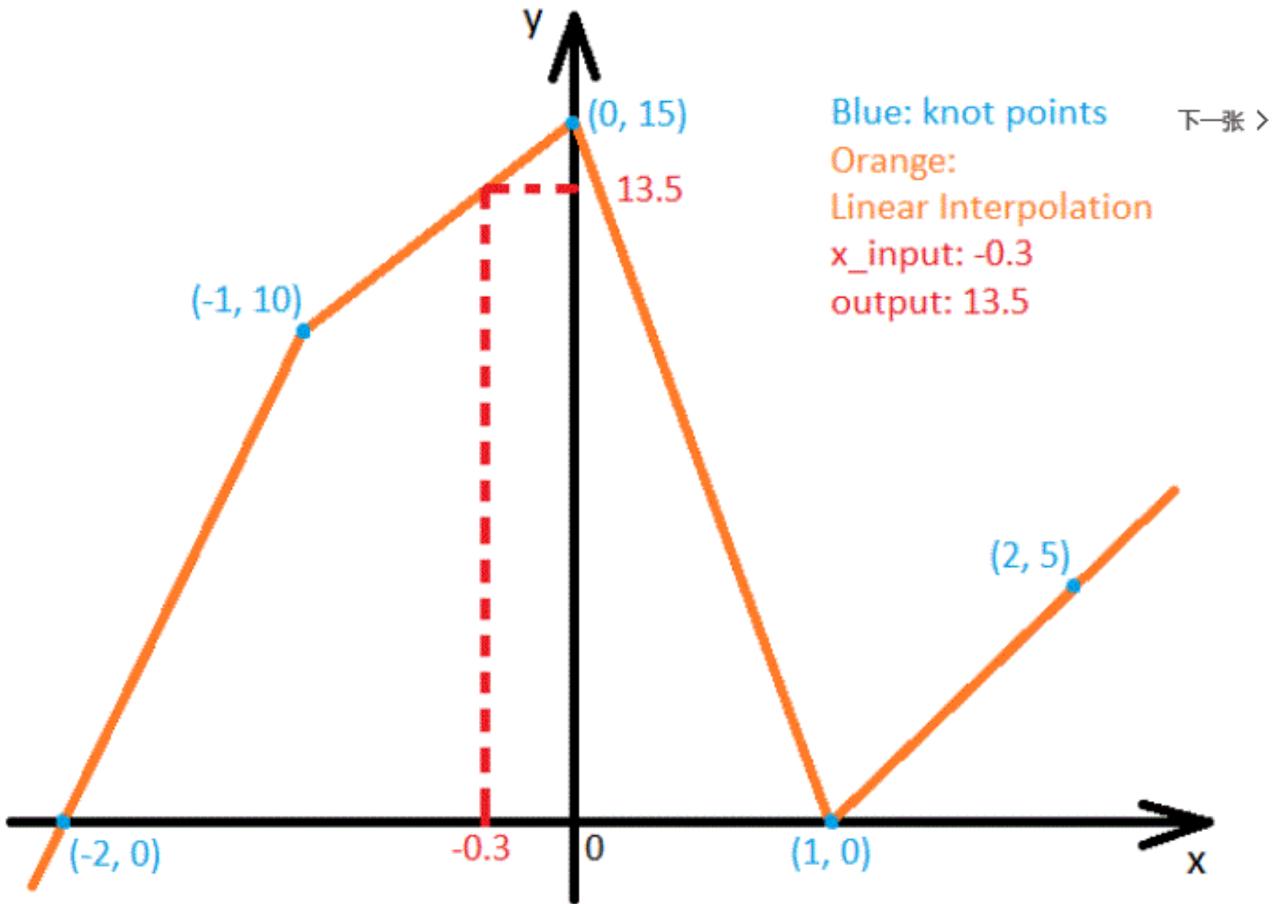
## 2019/03 Quantitative Researcher HackerRank

<<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=475845>

### ★ Linear Interpolator

You are asked to implement a linear interpolator. Namely you are given  $n$  points on a 2-dimensional coordinate system (known as the knot points). When these points are sorted by their  $x$  coordinates and then connected together using straight lines (in their sorting order), they define a piecewise-linear function  $L(x)$  known as the linear interpolation. When  $x$  is outside the range of all knot points,  $L(x)$  is defined by extrapolation i.e. extending the straight line connecting its two nearest knot points.

You have to complete a function `linear.interpolate(n: int, x_knots: List[float], y_knots: List[float], x_input: float) -> float` where `x_knots` and `y_knots` give you  $x$ - and  $y$ -coordinate of knot points. Your function should return  $L(x_{\text{input}})$ . Below is a graph of this calculation for Sample Case 0.



In case multiple knot points share the same x-coordinate  $x$ , when  $x_{\text{input}} \leq x$ ,  $\text{Li}(x_{\text{input}})$  is defined to be the  $y$ -value of the knot point with the smallest  $y$ -coordinate.  
When  $x_{\text{input}} > x$ ,  $\text{Li}(x_{\text{input}})$  is defined to be the  $y$ -value of the knot point with the largest  $y$ -coordinate.

#### Constraints

- $1 \leq n \leq 100,000$
- $x_{\text{knots}}$  and  $y_{\text{knots}}$  are guaranteed to have the same length
- No guarantee that  $x_{\text{knots}}$  and  $y_{\text{knots}}$  are pre-sorted in any way

#### Input Format

Your function signature (in Python) should look like: `linear_interpolate(n, x_knots, y_knots, x_input)`. The arguments are:

- $n$  (int): number of knot points
- $x_{\text{knots}}$  (List[float]):  $x$ -coordinates of the knot points
- $y_{\text{knots}}$  (List[float]):  $y$ -coordinates of the knot points
- $x_{\text{input}}$  (float): input to interpolation i.e. your function should return  $\text{Li}(x_{\text{input}})$

Note: The function signature in other languages will look slightly different due to grammar and naming conventions.

#### Sample Case 0

#### Sample Input

```

n = 5
x_knots = [-2.0, -1.0, 0.0, 1.0, 2.0]
y_knots = [0.0, 10.0, 15.0, 0.0, 5.0]
x_input = -0.3

```

Note: this sample input is formatted as arguments being passed into your function. If you want to write your own test case, it needs to be formatted as text which contains one number per line, in the order of n, x\_knots, y\_knots and x\_input. In this example it will look like the following (for simplicity for all subsequent samples will still use the short format):

```
5
-2.0
-1.0
0.0
1.0
2.0
0.0
10.0
15.0
0.0
5.0
-0.3
```

#### Sample Output

```
13.5
```

#### Explanation

x\_input -0.3 lies between -1.0 and 0.0. In this interval,  $LI(\cdot)$  is defined by the line segment connecting (-1.0, 10.0) and (0.0, 15.0).  $LI(-0.3) = 13.5$

#### Sample Case 1

#### Sample Input

4.GIF

鼠标滚轮缩放图片



#### Sample Input

```
n = 5
x_knots = [1.0, 2.0, -2.0, -1.0, 0.0]
y_knots = [0.0, 5.0, 0.0, 10.0, 15.0]
x_input = -0.3
```

#### Sample Output

```
13.5
```

#### Explanation

This is identical to example #00 except that the knot points are not pre-sorted

#### Sample Case 2

#### Sample Input

```
n = 5
x_knots = [-2.0, -1.0, 0.0, 1.0, 2.0]
y_knots = [0.0, 10.0, 15.0, 0.0, 5.0]
x_input = -3.0
```

#### Sample Output

```
-10.0
```

#### Explanation

When x\_input is outside the range of all knot points,  $LI(x_{input})$  is defined by extrapolation from the two nearest points (-2.0, 0.0) and (-1.0, 10.0)

#### Sample Case 3

#### Sample Input

```
n = 6
x_knots = [-2.0, -1.0, -1.0, 0.0, 1.0, 2.0]
y_knots = [0.0, 10.0, 12.0, 15.0, 0.0, 5.0]
x_input = -0.5
```

#### Sample Output

```
13.5
```

#### Explanation

Note that two knot points (-1.0, 10.0) and (-1.0, 12.0) share the same x-coordinate. x\_input is on the right-hand side of this point, hence it is interpolated based on the knot point with the largest y-coordinate. Namely it is interpolated between (-1.0, 12.0) and (0.0, 15.0)

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour.

[Start tour](#)



## ★ Daily Temperature By Town

You are given the daily temperature reading of  $P$  towns and the daily temperature reading of New York City (NYC) for  $N$  days. Write a function that returns the answer to the following five questions:

- Q1: The name of the place (either a town or NYC) with the largest variation in the daily temperature. Use the standard deviation to measure the variation.
- Q2: The median daily temperature of NYC when the daily temperature of Town2 is between 90 and 100 degrees (inclusive of 90 and 100). Round your answer to the nearest integer.
- Q3: Fit  $P$  simple linear models with intercept using least squares to predict the daily temperature of NYC given each individual town. Find the **sum of the absolute values** of the regression coefficients, rounded to the nearest integer.
- Q4: For the given data, find the town that is most predictive of the daily temperature of NYC. By most predictive, we mean the town that leads to the lowest mean squared error (MSE) on the given data when fit using a linear model with intercept.
- Q5: For the given data, find the **two** towns that are **jointly** most predictive of the daily temperature of NYC. As before, we mean the two towns that lead to the lowest MSE when fit using a linear model with intercept.

### Function Description

Complete the function `main()` to read the data from `stdin` and answer the six questions. The input data is in CSV format with a header row indicating the town names; use this header rather than making assumptions about column order. Depending on the language you choose we may provide boilerplate code for reading the data - you are welcome to use this if you wish.

7.GIF



鼠标滚轮缩放图片

The function `main` should return a string with the answer to the six questions below, delimited by a comma. For example, suppose your answers to the five questions are as follows:

- Q1: Town1
- Q2: 95
- Q3: 3
- Q4: Town2
- Q5: Town1 and Town2

You should return "Town1,95,3,Town2,Town1,Town2".

### Constraints

- $N > 2$  and  $4 < P < 60$
- There exists at least one data point for Town2 where the highest temperature is between 90 and 100 degrees.

### Input Format For Custom Testing

The first line contains a comma separated list of strings corresponding to the name of the towns.

Each subsequent line contains comma separated values corresponding to the temperature of each town.

Example:

```
Town1,Town2,Town3,Town4,Town5,NYC
70,95,34,46,10,50
65,88,45,24,32,51
87,91,23,35,10,78
67,101,34,55,15,88
```

**Input Format For Custom Testing**

The first line contains a comma separated list of strings corresponding to the name of the towns.

Each subsequent line contains comma separated values corresponding to the temperature of each town.

下一张 &gt;

Example:

```
Town1,Town2,Town3,Town4,Town5,NYC
70,95,34,46,10,50
65,88,45,24,32,51
87,91,23,35,10,78
67,101,34,55,15,88
```

**Sample Case 0****Sample Input For Custom Testing**

```
Town1,Town2,Town3,Town4,Town5,NYC
70,95,34,46,10,50
65,88,45,24,32,51
87,91,23,35,10,78
67,101,34,55,15,88
```

**Sample Output**

```
NYC,64,5,Town2,Town1,Town2
```

**Sample Case 1****Sample Input For Custom Testing**

DESKTOP-NV1HJ14 / 1

**Sample Input For Custom Testing**

```
Town1,Town2,Town3,Town4,Town5,Town6,NYC
34,32,53,5,10,29,40
86,91,24,10,12,50,45
56,78,90,23,23,45,23
23,92,45,44,55,67,34
12,99,23,34,64,56,35
22,102,11,23,45,65,43
```

**Sample Output**

```
Town3,35,1,Town3,Town3,Town5
```

**Sample Case 2****Sample Input For Custom Testing**

```
Town3, NYC, Town1, Town4, Town5, Town6, Town2
67,69,60,63,65,69,95
31,48,61,62,46,46,102
63,81,61,73,49,100,94
88,77,71,67,64,90,95
63,66,61,77,43,82,81
```

**Sample Output**

```
Town6,77,4,Town6,Town5,Town6
```

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=475845>

**Phase one** :是叫你设计一个**Random Class**, 要求**specify**一个**interval**之后, 它**return interval**中的随机数字。但额外的要求是, 每个数字只能**return**一次, 一旦出现过, 就不会再**return**相同的数字。直到最后把这个inteval里所有的数字**return**完。然后再回到所有数字都可能被**return**的阶段。然后叫自己写**testing**。

**Phase two**是要求支持随时可以改变interval range的feature, 而且range改变之后, 此前**return**过的数字依然不会再**return**, 直到range里的所有数字都被**return**一遍。然后是写更多的**testing**。

楼主的做法:

是用一个ArrayList **candidateList**装可以**return**的candidate, 每次通过call Java的Random.nextInt 来得到to-return element的index。**return**了这个element出去之后, 把它和arrayList的最后一个element交换位置, 再**remove**。这样removal就是O (1)。支持update range的方法是maintain另外一个List, 记录所有被**remove**掉了的element, 每次update之后重建ArrayList, 并把所有被记录了**remove**的element从重建的list里去除。有这第二个array的好处是, 当range里所有element都进了removed list之后, 可以直接调换两个array的reference, 重新开始而不需要重新**populate**。

按照这样的做法, initialize需要O (N) , UpdateRange O(N), getRandom O(1). 30分钟内要做完coding, 自己写test和跑test时间还是比较紧张的。不确定是否是最优解, 应该可以再优化, 但在big O time上似乎已经没法做到更好了。结束之后问我假如用BST或者hashmap的话会有什么区别, 似乎是还有更好的解。但我分析之后认为不管用什么DS, Initialize和UpdateRange似乎没法做到比O (N) 更好。如果地里有想到更好解法的大神, 请务必让我知道。

第二个list仅是记录, 不存在lookup操作, 所以并不一定要用set。调换只是一个小优化, 也不影响BigOTime, 做不做都无所谓的。

最后说下感受, 总的来说感觉2s面试的professionalism还是差了点, 尤其是和谷歌比。大多数问题都是问了很多年的原题, 可以看出他们的题库也就这么点而且没动力更新。面试官全程基本没什么反馈, 我跟他说我的想法他就只会说ok那你写写看, 很多时候干脆不说话, 估计是在干别的, 于是后来楼主也不说话了。写出来了以后, 感觉他也无法判断对错, 只能通过跑test的结果。有点好笑的是面试结束之后楼主没有关页面, 看到他自己留在那个hackarank页面上各种写和跑test。我觉得一个有足够水平的面试官, 熟悉问题、肉眼判断解法对错是基本的吧。

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=496638&highlight=two%2Bsigma>

## 2019/03 Software Engineer

老题: 1)**friend circle** 和 2)**longest chain**. 对没见过这两题的朋友, **Friend circles**在leetcode 547. **Longest chain** 在 <https://www.cnblogs.com/EdwardLiu/p/6177843.html>

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=494286&highlight=two%2Bsiga ma>

## 2019/03 Software Engineer OA

---

还是原来的配方，还是熟悉的味道，Friend Circles + longest chain

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=504459&highlight=two%2Bsiga ma>

## 2016/12 超全

---

(太长了，直接看链接)

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=216941&highlight=two%2Bsiga ma>

大概就是：

- Behavioral
- 电面6题加答案
- onsite 三套，每套三轮，附答案

## 2018/12 Quantitative Research Intern

---

1，以下俩linear regression模型 $Y \sim X$ ,  $X \sim Y$ , 那么他们的R2、系数Beta、以及noise (epsilon) 是什么关系？2，如果你是个taxi driver, 如何规划一天让自己受益最大？如果你可以选择做Uber和普通Cap driver, 应该如何利用数据做出选择？

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=467937&highlight=two%2Bsiga ma>

## 2019/01

---

题目如下：

一个void generator (int ms, int bufferSize), 每隔 m (ms) 会generate 一个token 到queue 里面。另外一个int getter(int desiredAmount) 去queue 里面fetch 它最大可以拿的数目，就是如果queue 里面的size 大于它想要的，那么就返回那个desiredAmount,不然就直接返回当前queue 所有的数目。

我用了个不熟的synchronousQueue来做，其实应该直接用锁就好了，对queue的size 来增加减少，那时候也没想出来。。。

楼主写的（不一定对）： import queue import threading import time

```
class tokenGenerator: def __init__(self): self.q = queue.PriorityQueue() self.count = 0 . check  
1point3acres for more. def generator(self, time, buffer_size): self.buffer_size = buffer_size self.time =  
time self.enqueue_token(). 1point3acres  
  
def enqueue_token(self): print('ENQUEING...', self.count) threading.Timer(self.time,  
self.enqueue_token).start() self.count+=1 if self.count < self.buffer_size + 1: self.q.put((self.count,  
"Task"))  
  
def getter(self, amount): res = [] for _ in range(amount): if self.q.empty(): return res  
res.append(self.q.get()) self.count-=1 return res
```

不是的话 算法题不外乎就是

1. **Random Mod 5 Iterator** 可参考：<https://github.com/sreeprasad/TomMod5Iterator.java>
2. **weight random class**
3. **implement a random number generator in a range without repeats**(我拿到这题 等我确定我有没有onsite 我会把详细的code放上来)

## 2019 Summer Intern

---

还是两道原题，第三道题没写，一共三小时。

**Q1 implement a linear interpolate.** 有几种特殊情况列的很清楚。我是先排序，按x排，x一样按y排，然后用binary search去找区间。

**Q2 数据处理，好像是5个小问。找median, standard deviation, least square。** 主要用了pandas和sklearn里的linear regression

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=507865&highlight=two%2Bsigma>