* 1. **High five**

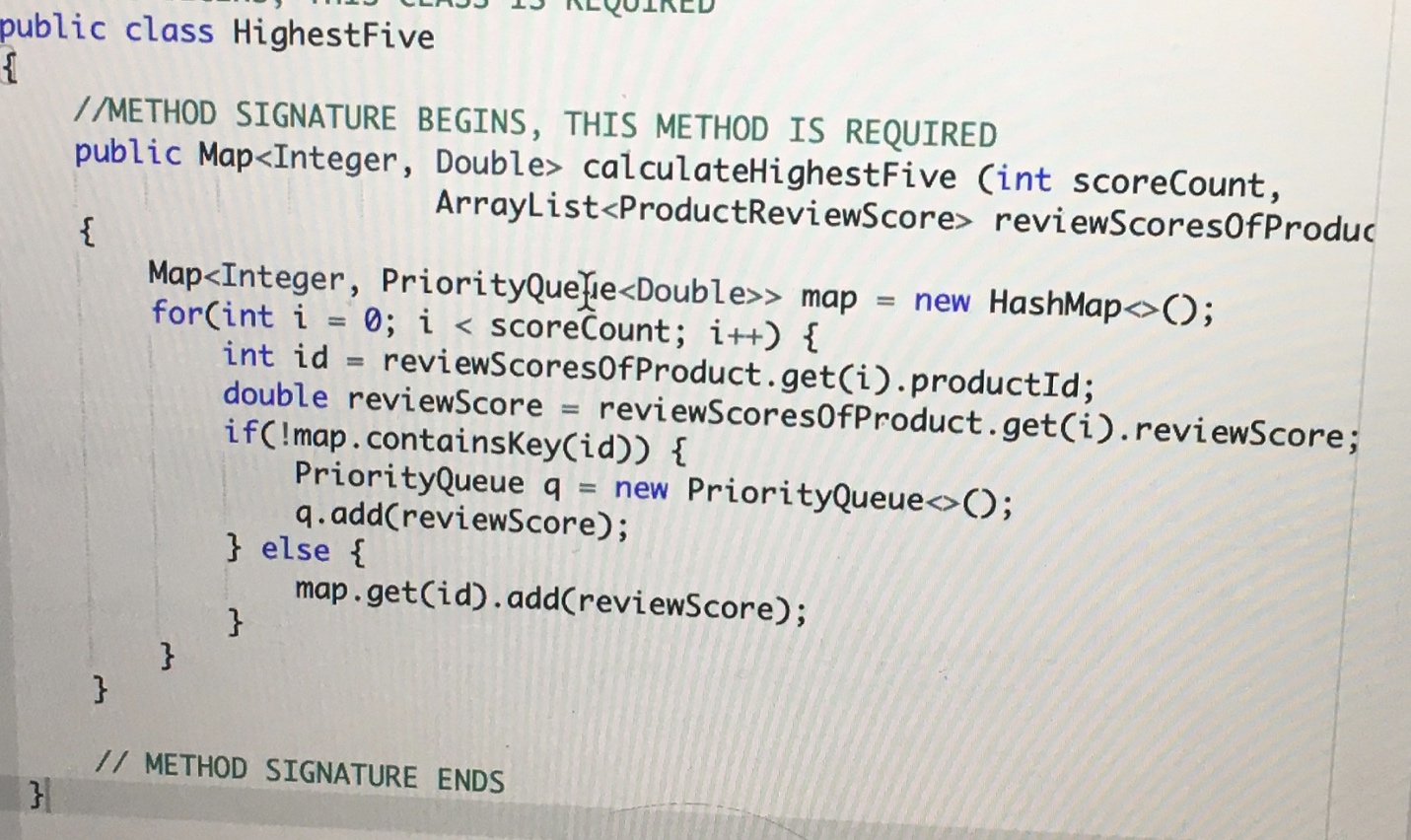
本地VS code <https://www.lintcode.com/problem/high-five/description>

<https://www.1point3acres.com/bbs/thread-478168-1-1.html>

2. Highest Five/nput: (RfiEGB99[ProductReviewScore] class) , [Product id van, [product id 
id 
[Product id average value Of top 51, [product id, average value Of top 5], ...J 
eg: 
[lid-I, 11, [id-2, 21, 31, [id-2, 41, lid-I, 51, (id-2, 61, lid-I, 71, lid-I, 91, (id=2, 10], 11], 2, 121] 
id=2 v-a/ TO, 70+ 72)/5 
'BG±fiproduct review scorefitypetgedouble. 

<https://www.1point3acres.com/bbs/thread-476958-1-1.html>

Problem l. Test Cases l, Output I 
At Amazon, we focus on the customer, and one important metric is how long a 
customer waits for a page to render. Every time we render a page we record the 
page rendered, date and time the page was rendered, and how long it took to 
render the page. To help assess the customer experience we want to know how lor 
average it is taking to render each page, but we are looking for outliers. To 
calculate these outliers we take the five longest (worst) rendering times for a page 
and average them together. Assume that each page has been rendered at least fi\.a 
times. 
Write an algorithm to find the average of five worst rendering times for each page. 
'Input 
The input to the function/method consists of two arguments- 
renderCount, an integer representing the total number of pages to be rendered 
(includes the repeated rendering of the same page), 
renderTimeOfPages, a list where each element of the list consists of an integer 
representing the page ID and a real number representing the rendering time of the 
page. 
Output 
Return a list where each element of the list consists of an integer representing the 
page ID and a real number representing the average of five worst rendering times c 
each page. 
Example 
Input: 
renderCount = 13 
b renderTimeOfPages 
I[pagelD: 1, renderTime: 40], 
[pagelD: 2, renderTime: 901, 
roaaelD: 1, renderTime•, 501. 



* 1. **K closest point to origin parameters given????**

Leetcode 973 <https://leetcode.com/problems/k-closest-points-to-origin/>

非原点 lintcode 612 <https://www.lintcode.com/problem/k-closest-points/description>

<https://www.1point3acres.com/bbs/thread-467307-1-1.html>

Amazon Fresh is a grocery delivery service that offers consumers the option of purchasing 
their groceries online and schedule future deliveries of purchased groceries. Amazon's 
backend system dynamically tracks each Amazon Fresh delivery truck and automatically 
assigns the next deliveries in a truck's plan. To accomplish this, the system generates an 
optimized delivery plan with X destinations. The most optimized plan would deliver to the 
closest X destinations from the start among all of the possible destinations in the plan. 
Given an array of N possible delivery destinations, implement an algorithm to create the 
delivery plan for the closest X destinations. 
Input 
The input to the function/method consists of three arguments: 
numDestinations, an integer representing the total number of possible delivery destinations 
for the truck (N); 
allLocations, a list where each element consists of a pair of integers representing the x and y 
coordinates of the delivery locations; 
numDeliveries, an integer representing the number of deliveries that will be delivered in the 
plan (X).

Output 
Return a list of elements where each element of the list represents the x and y integer 
coordinates of the delivery destinations. 
Constraints 
numDeliveries s numDestinations 
Note 
The plan starts from the truck's location [O, O . The distance of the truck from a delivery 
destination (x, y) is the square root of x + 
. If there are ties then return any of the 
locations as long as you satisfy returning X deliveries. 
Example 
Input: 
numDestinations = 3 
alll_ocations = [Il, 2], [3, 4], [I , - 
numDeIiveries = 2 
Output: 
[M, [l , 21] 
Explanation: 
The distance of the truck from location [I , 2] is square root(5) 
= 2.236 
The distance of the truck from location [3, 4] is square root(25) = 5 
The distance of the truck from location [I , -I] is square root(2) 
= 1.414 
numDeliveries is 2, hence the output is [I , 
-1] and [1, 2].

<https://www.1point3acres.com/bbs/thread-457483-1-1.html>

z 
3 
4 
6 
9 
10 
11 
INCLUDE HEADER FILES NEEDED BY YOUR PROGRAM 
SOME LIBRARY FUNCTIONALITY MAY BE RESTRICTED 
DEFINE ANY FUNCTION NEEDED 
FUNCTION SIGNATURE BEGINS, THIS FUNCTION IS REQUIRED 
int> > nearestXsteakHousesCint totatSteakhouses, 
attLocations, 
int numsteakhouses) 
// WRITE YOUR CODE HERE 
// FUNCTION SIGNATURE ENDS