Amazon | OA 2020 | Five Star Sellers

LEETCODE : <https://leetcode.com/discuss/interview-question/854110/>

https://leetcode.com/discuss/interview-question/854110/

QUES) Given the number of five-star and total reviews for each product a company sells, as well as the threshold percentage, what is the minimum number of additional five-star reviews the company needs to become five star seller.  
For ex, there are 3 products (n=3) with productRatings =[[4,4],[1,2],[3,6]], percentage rating threshold = 77.  
[1,2] indicates => [1 (five star reviews) ,2 (total reviews)].  
We need to get the seller reach the threshold with minimum number of additional five star reviews.

Before we add more five star reviews, the percentage for this seller is ((4/4) + (1/2) + (3/6))/3 = 66.66%  
If we add a five star review to 2nd product, ((4/4) + (2/3) + (3/6))/3 = 72.22%  
If we add another five star review to 2nd product, ((4/4) + (3/4) + (3/6))/3 = 75%  
If we add a five star review to 3rd product, ((4/4) + (3/4) + (4/7))/3 = 77.38%  
At this point, 77% (threshold) is met. Therefore, answer is 3 (because that is the minimum five star reviews we need to add, to get the seller reach the threshold).

public static int fiveStarReviews(List<List<Integer>> productRatings, int ratingsThreshold){

}

Constraints:  
1<= productRatings.size() <=200  
In product ratings, [fivestar, total], fivestar <=100, total<=100  
1<=ratingsThreshold< 100  
productRatings contains only non negative integers.

Ans :

class Solution:

def fiveStartReviews(self, productRatings, ratingsThreshold):

n = len(productRatings)

ans = 0

cur\_rating = sum(p[0]/p[1] for p in productRatings) / n \* 100

def diff(p): return (p[0]+1)/(p[1]+1) - p[0]/p[1]

heap = [[-diff(p)]+p for p in productRatings] # create diff and store in heap & heapify

heapq.heapify(heap)

while cur\_rating < ratingsThreshold:

p\_diff, p0, p1 = heapq.heappop(heap)

cur\_rating = (cur\_rating \* n + (-p\_diff) \* 100) / n # update current rating by adding the difference

heapq.heappush(heap, [-diff([p0+1, p1+1]), p0+1, p1+1]) # maintain diff order in heap

ans += 1

return ans