

Hacker Cup 2017 Qualification Round Solutions

 FACEBOOK HACKER CUP · MONDAY, JANUARY 9, 2017

Here are the solutions to the Hacker Cup 2017 Qualification Round problems. If you had a rejected solution and want to find out where you went wrong, read on and download the official input and output!

Input / Output / Solutions: <https://www.dropbox.com/sh/rzyhrzol...>

The problems for this round were written by Wesley May.

Progress Pie

For each point, we need to check two things:

1) Is the point close enough to the center that it could be within the circle at all?

This is as simple as computing the distance between the point and the center of the circle:
 $\text{sqrt}((X - 50)^2 + (Y - 50)^2)$. This distance must be no greater than 50, the radius of the circle.

2) Is the point between the two lines that define the bounds of the current circle sector?

We'd like to know the angle of the given point in the reference frame where 0 degrees is up and 90 degrees is to the right, since that's the way the boundary of the circle sector moves as the progress increases.

Most languages have an `atan2()` function which can give you the angle from (0, 0) to a given point. However, this function puts 0 degrees to the right and 90 degrees upwards, so it's necessary to translate the output to the desired reference frame.

And of course, if $P = 0\%$, all points are white.

Lazy Loading

Consider the heaviest item that hasn't yet been moved. When this item is moved, it should certainly be on the top of the stack to make the current bag appear as heavy as possible. To move this item we'll need to add as many other items as necessary to make the bag appear to weigh at least 50 pounds.

This leads us to the following greedy solution: At every step, put the heaviest available item (with weight w) in the bag, along with the K lightest available items, where K is the lowest integer that satisfies $(K + 1) * w \geq 50$. If there aren't enough remaining items to fake a 50-pound bag, then you can't complete another trip. Pretend that you put those items in the last bag moved.

To efficiently find the heaviest and lightest items, we should first sort the input. This takes $O(N \log N)$ time, and the rest of the algorithm takes $O(N)$ time. (Given the small bound on the weights of the items, a more efficient sorting method is possible, but unnecessary.)

Fighting the Zombie

For each attack, we need to compute the probability that it rolls at least H damage. We can compute this using dynamic programming.

Let $f(D, K)$ be the probability of dealing at least K damage with D dice. For a given input X, Y, Z we want to compute $f(X, H - Z)$. We can use the following recursive definition:

- $f(D, K) = 1$ for $K \leq 0$ (We can always do at least 0 damage)
- $f(0, K) = 0$ for $K > 0$ (We can't do a positive amount of damage with 0 dice)
- $f(D, K) = (1 / Y) * (f(D - 1, K - 1) + f(D - 1, K - 2) + \dots + f(D - 1, K - Y))$

This last formula combines the outcomes of all possible die rolls for a single die, and weights them evenly by $1 / Y$.

In this way, we can compute the probability of success for each attack in $O(X * Y * (H - Z))$ time.

Since the most damage we can do is $X * Y$, we can trivially reject any case where $H - Z > X * Y$. That means we can also consider the time complexity to be $O(X * Y * \min(X * Y, H - Z)) = O(X^2 * Y^2)$.

Like

Comment

Share

205

[Top Comments](#)



Write a comment...



Davide Russo Zachary Burt good idea to use a closed form, recursive relations are more charming tho 😊 not considering the fact that you can easily use memoization there!

Like · Reply · about an hour ago



Pratik Anand Facebook Hacker Cup I faced a potential bug during final submissions which invalidated all of my submissions, where to reach out for this ?

Like · Reply · about an hour ago

3 Replies · about an hour ago



Adam Benali I had a completely different approach for fighting the zombie : I am computing the number of positive integer solutions (x_i) to the equation $\sum_{i=1}^X (x_i) = n$, then I'm summing them up. Divided by Y^X , this is what defines the probability of getting a final damage equal to n for some spell. Does it make sense ? (Well it should since it got accepted but I'm sharing the idea anyway ..)

Like · Reply · 33 minutes ago · Edited



Shiva Soni number of ways for given x, y, z, h are coefficient of q^i in $(q^1 + q^2 + \dots + q^y)^x$ for all $i \geq (h-z)$..brute multiplication or FFT does the work easily ..and probability is $(\text{number of ways}) / \text{pow}(y, x) \dots$ 😊

Like · Reply · 58 minutes ago · Edited



Palash Das For Progress Pie :

atan2 will give wrong answer when point lies at center (50,50) i.e. atan2(0,0) gives error value 0 instead of $\pi/4$ (expected) <http://en.cppreference.com/w/cpp/numeric/math/atan2...> See More

Like · Reply · 1 · about an hour ago

7 Replies · 26 minutes ago



Gustavo Luis Veloso I dont know the format of sourc code and how can i do the output

Like · Reply · about an hour ago



Daly Kjeldsen Score break down:

Solved #1: 5752 people

Solved #2: 7422 people... See More

Like · Reply · 7 · about an hour ago



Sotirios Nikoloutsopoulos Facebook Hacker Cup can you recheck my solution for Lazy Loading? My output is the same as the one given in your dropbox folder. Thank you.

Like · Reply · 2 · about an hour ago

1 Reply



Galang Kangin problem A do not contain the tricky case: 0 50 50

Like · Reply · 3 · about an hour ago

1 Reply



Sayed Ahmad can anyone clarify more what atan2() function does.i didn't understand how to check a point inside the sector or not?

Like · Reply · 56 minutes ago

[View 13 more comments](#)