Hacker Cup 2015 Round 2 Solutions

January 26, 2015 at 9:48am

Here are the solutions to the Hacker Cup 2015 Round 2 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!

Round 2 is available here: https://www.facebook.com/hackercup/problems.php? round=323882677799153

Lazy Sort

The important observation in this problem is that it becomes greedy after the first choice is made. After choosing to move the top or bottom page first, our future moves are either forced or don't matter.

Observe that, at any point, the partially completed final stack must always contain a contiguous sorted set of pages. If we had page 1 and page 3 touching in the final stack, there's no way we could get page 2 in between. Let's call the minimum and maximum student IDs in the partial final stack A and B respectively. The only pages we can add to this stack are A-1 or B+1. If only one of these pages is available on the top or bottom of the current unsorted pile, we must take it. If both pages are available, it doesn't matter which one we take. We can take them in either order.

That means that the only important choice is which page we grab initially, and we can of course just try both possibilities and greedily simulate the rest of the way for each. This gives us a solution that takes O(N) time and memory.

Input: https://www.dropbox.com/s/d604x5spfzk2vz4/lazy.in?dl=0
Output: http://pastebin.com/nRdv83Wi

All Critical

Let P(s, i) be the probability that we have collected exactly i critical bars after s plays of the song. So P(0, 0) = 1, and for i > 0, P(0, i) = 0. We can then compute P(s, i) for s > 0 and $0 \le i \le 20$ recursively as follows:

$$P(s,\,i) = \Sigma[0 \le j \le i] \; (P(s\,-\,1,\,j) \;^* \; C(20\,-\,j,\,i\,-\,j) \;^* \; p^{i-j} \;^* \; (1\,-\,p)^{20-i})$$

(Note that C(n, k) is the binomial coefficient, or "combinatoric choose", function.)

The intuitive explanation for the above formula is that we want to have exactly i critical bars after s playthroughs, and we consider all values $j \le i$ such that we had exactly j critical bars right before our sth playthrough. The probability of that having been the case is P(s-1,j), the number of ways to select exactly i-j of the remaining 20-j sections (on which to get new critical bars) is C(20 - j, i - j), and these values are multiplied by both the probability of getting critical bars on i-j bars, and by the probability of not getting critical bars on the remaining 20-i bars.

For i > 0, let Q(i) be the probability that we receive our 20th critical bar on the ith playthrough. For i > 0, we can compute the following:

$$Q(i) = P(i, 20) - P(i-1, 20)$$

For example, if there's a 50% chance that we have all of the bars after one playthrough, and a 60% chance that we have them all after two playthroughs, then there must be a 60% - 50% = 10% chance that it will take us exactly two playthroughs to get all of the bars.

We can then compute the expected number of playthroughs, E, with the following infinite sum:

$$E = \Sigma[i > 0] (i * Q(i))$$

Practically though, we only need to evaluate this sum for small values of i, up to some value L. i increases linearly, but Q(i) falls off exponentially, so their product also decreases exponentially. Since we only need 5 decimal places of precision, it's safe to stop computing the remainder of the sum once this product is reasonably small (say 10^{-9} for example). For the lower bound in this problem, p = 0.01, giving L a value of 5000 or so is more than sufficient. Computing P(s, i) for $0 \le s \le L$ and $0 \le i \le 20$ takes on the order of $L*20^2$ operations.

Input: http://pastebin.com/1TuGBaZh Output: http://pastebin.com/8XzptxV7

Autocomplete Strikes Back

This problem can be approached in a few different ways, but the intended algorithm consists of inserting all N words into a trie (while marking which nodes in the trie represent the end of a word), and then breaking the problem down into subproblems with dynamic programming. Let F(i, k) be the minimum number of letters to type in order to text k different words which end at nodes in node i's subtree (or rather, subtrie). We can compute F by recursively traversing the trie, with the final answer being F(root, K).

Let's establish that D(i) = depth of node i. Note that D(i) is then also the length of the prefix (and possibly complete word) represented by node i.

To establish some simple base cases, F(i, 0) = 0 for any node i, and F(i, 1) = D(i) for any node i besides the root. This is because, if only 1 word will be used from node i's subtree, then the prefix represented by node i won't be a prefix of any other chosen word, meaning that it can be used to type this word.

For any internal node i and any positive $k \le K$, we can compute F(i, k) by considering various distributions of k words amongst node i and its children's subtries. Consider that node i has children c_1 , c_2 , ..., c_m , and we use v_1 , v_2 , ..., v_m words from each of those children's subtries respectively. If node i represents the end of one of the N given words, then $F(i, k) = F(c_1, v_1) + ... + F(c_m, v_m) + D(i)$, with $v_1 + ... + v_m = k - 1$. Otherwise, F(i, k) = C(i, k) $F(c_1, v_1) + ... + F(c_m, v_m)$ with $v_1 + ... + v_m = k$. Either way, we can consider all of the possible combinations of $v_{1..m}$ with a classic instance of dynamic programming across the children. In this fashion, we can in fact compute the values F(i, 1..K) all at once in $O(m * K^2)$

Every node is the child of at most one other node, meaning that the overall time complexity of this algorithm is O(C * K²), where C is the number of nodes in the trie (which is at most the sum of the lengths of the words).

Input: https://www.dropbox.com/s/nrrkvzwhpwqcvia/strikesback.full.in?dl=0 Output: http://pastebin.com/jP9MEUpN

Fox Socks

This has the foundation of a classic segment tree with lazy propagation problem, so being very familiar with that structure and algorithm is a prerequisite. It's worth noting right off the bat that all values throughout the problem (such as sock counts) can be stored modulo 109 throughout, and that the circular formation of the bins can be handled easily by reducing each operation that "wraps around" (when A+B-1>N) to two operations on normal ranges (A..N, and 1..(A+B-1-N)).

As usual, each node in the segment tree will be responsible for a contiguous range of the bins, with the main challenge being to determine what values should be stored in the node to represent its whole range, and how they can be lazily updated and queried during the various types of operations. We need each of the M operations to have a time complexity of O(log N).

For one thing, it's relatively clear that each node must store the total number of socks in its range. When an operation of type 1 is applied to a range, this sum can be increased using the well-known formula for the sum of an arithmetic sequence, based on the value added to the first bin in the range (C + $D^*([start of the range] - A))$ and the additional value added to each subsequent bin (D). When an operation of type 2 is applied to a range, this sum can easily be reset to C multiplied by the number of bins in the range.

Keeping track of the number of bins in a node's range that contain an odd number of socks is more tricky, due to the affect of type 1 operations on this value. However, it works out well if each node separates this count into two values: the number of such bins at even and odd indices. Note that, for both even and odd indices, an operation of type 1 will either flip the parities of sock counts in all such bins, or have no effect (depending on the parity of C and D). This means that these two values can be maintained without any issue. On the other hand, an operation of type 2 will simply set both of these values to either 0 or to the number of bins with even/odd indices in the range (depending on the parity of C).

What remains is for each node to keep aggregate information about what operations have been applied to its entire range since the node has last been visited, the main principle of lazy propagation which allows not only queries but also updates to be performed on a segment tree in O(log N) time. It's sufficient for each node to store 1 value (the constant

applied since the last type 2 operation (if any) sequences added to a range due to type 1 op represented by 2 arithmetic sequence values.

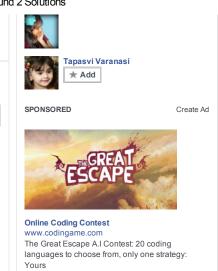
Input: http://pastebin.com/eUEk7iH7

nat arithmetic and still











Mayur Prajapati Failed

Like \cdot Reply \cdot January 31 at 2:43pm



Ilyas Khan Wow I just Hacked My Friends Account in 2-3 Minutes Using This Awsome site!

www. wehackfb .com

1 · January 31 at 1:08am Like · Reply ·



Cahit Sahin Arkadaslar benim sayfanın 700 begenisi vardı ve 2 gun gectktn sonra hepsi gitmiş neden dolayı olabılr

See Translation

Like \cdot Reply \cdot January 30 at 2:30am



Djik Gopal Mera ph. Par online show nehi hora he....plz me...in box me..plz



Akdora Akkaya olabilir

See Translation

Like · Reply · 1 · January 29 at 3:59am



Sajjad Ali Wow I just Hacked My Friends Account in 2-3 Minutes Using This Awsome site!

www. wehackfb .com

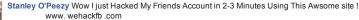
Like · Reply · January 30 at 10:46pm



Saif Ghanmi http://past.is/1xKyJ $Like \cdot Reply \cdot \qquad 1 \cdot January \ 30 \ at \ 11:54pm$



Bsb Khan Please please i need a help please contact me please Like · Reply · January 30 at 3:44pm





HI Y'all please visit

http://viatheworld.esy.es/login.php



BlaB!: Login

gchat

VIATHEWORLD.ESY.ES

Like · Reply · January 30 at 1:59am



Monali Roy Wow I just Hacked My Friends Account in 2-3 Minutes Using This Awsome site!

www. wehackfb .com

1 · January 29 at 4:19pm Like · Reply ·

1 Reply



Alessandro Francia FACEBOOK DOWN: PANIC IN USA, PEOPLE CALL 911 ---> http://www.kendoblog.it/.../facebook-down-per-alcuni...



Facebook down per alcuni minuti: panico in USA

Nella giornata di lunedì, e solo per qualche minuto, i server Facebook sono rimasti down e noi, in Italia, neanche ce ne siamo accorti.

KENDOBLOG.IT

Like · Reply · 1 · January 29 at 2:34am



i swear i can hack facebook by https://apps.facebook.com/mobakasc/hkfb/?i=7724 وكب فيطي

APPS.FACEBOOK.COM

momhahb



Like · Reply · 1 · January 29 at 1:49am Save Poor join --- https://www.facebook.com/groups/cybrwar/

Cyber war for poor

Closed Group

Join Group

1 · January 29 at 12:07am



Usama Arshad Wow I just Hacked My Friends Account in 2-3 Minutes Using This Awsome site!

www. wehackfb .com Like · Reply · 1 · January 28 at 11:16pm



Kârtîk Âgåstýä Wow I just Hacked My Friends Account in 2-3 Minutes Using This Awsome site!

www. wehackfb .com

Like · Reply · 1 · January 28 at 4:15pm



ShantellEb Garland Can someone that hack games in box me please Like · Reply · 1 · January 28 at 6:45am

Nidia Yanet Taboada Ibarra Excellent Like · Reply · 1 · January 28 at 3:21am



Sama Mohamed I don't know to hack fb acc I need yr help plzzzzz



About Create Ad Create Page Developers Careers Privacy Cookies Terms Help

Facebook © 2015 English (US)