Dynamic Programming Problem Set

Nick Haliday

2012-10-12

1 Intro

Once you know the basics, the best way to improve at DP is practice, so that's what we're doing today. Some of the problems might have non-DP solutions, but all have at least one DP solution. The problems are ordered roughly by my estimation of their difficulty. Problem difficulty is subjective, though, so please look at all of them even if you get stuck on some of the earlier ones. Some of the harder problems came from last week's lecture.

2 Warmup

1. Solitaire [Codeforces Round 132]

   A boy named Vasya wants to play an old Russian solitaire called "Accordion". In this solitaire, the player must observe the following rules:

   - A deck of n ($n \leq 52$) cards is carefully shuffled, then all n cards are put on the table in a line from left to right;
   - Before each move the table has several piles of cards lying in a line (initially there are n piles, each pile has one card). Let's number the piles from left to right, from 1 to x. During one move, a player can take the whole pile with the maximum number x (that is the rightmost of remaining) and put it on the top of pile $x - 1$ (if it exists) or on the top of pile $x - 3$ (if it exists). The player can put one pile on top of another one only if the piles' top cards have the same suits or values. Please note that if pile x goes on top of pile y, then the top card of pile x becomes the top card of the resulting pile. Also note that each move decreases the total number of piles by 1;
   - The solitaire is considered completed if all cards are in the same pile.

   Vasya has already shuffled the cards and put them on the table, help him understand whether completing this solitaire is possible or not.

2. Eating Together [Ionescu Victor, 2007]

   The cows are so very silly about their dinner partners. They have organized themselves into three groups (conveniently numbered 1, 2, and 3) that insist upon dining together. The trouble starts when they line up at the barn to enter the feeding area.

   Each cow i carries with her a small card upon which is engraved D_i ($1 \leq D_i \leq 3$) indicating her dining group membership. The entire set of N ($1 \leq N \leq 30,000$) cows has lined up for dinner but it's easy for anyone to see that they are not grouped by their dinner-partner cards.

   FJ's job is not so difficult. He just walks down the line of cows changing their dinner partner assignment by marking out the old number and writing in a new one. By doing so, he creates groups of cows like 111222333 or 333222111 where the cows' dining groups are sorted in either ascending or descending order by their dinner cards.

   FJ is just as lazy as the next fellow. He's curious: what is the absolute minimum number of cards he must change to create a proper grouping of dining partners? He must only change card numbers and must not rearrange the cows standing in line.

3 Tougher

1. Barcode [Codeforces Round 139]

   You've got an $n \times m$ pixel picture. Each pixel can be white or black. Your task is to change the colors of as few pixels as possible to obtain a barcode picture.

   A picture is a barcode if the following conditions are fulfilled:

   - All pixels in each column are of the same color.
   - The width of each monochrome vertical line is at least x and at most y pixels. In other words, if we group all neighbouring columns of the pixels with equal color, the size of each group can not be less than x or greater than y.

   Given $1 \leq n, m, x, y \leq 1,000$ with $x \leq y$, and the original picture, output the minimum number of pixels you need to change to create a barcode picture.

2. Billboards [Interviewstreet]

   ADZEN is a very popular advertising firm in your city. In every road you can see their advertising billboards. Recently they are facing a serious challenge, MG Road the most used and beautiful road in your city has been almost filled by the billboards and this is having a negative effect on the natural view.

   On people's demand ADZEN has decided to remove some of the billboards in such a way that there are no more than K billboards standing together in any part of the road.

   You may assume the MG Road to be a straight line with N billboards. Initially there is no gap between any two adjacent billboards.

   ADZEN's primary income comes from these billboards so the billboard removing process has to be done in such a way that the billboards remaining at end should give maximum possible profit among all possible final configurations. The total profit of a configuration is the sum of the profit values of all billboards present in that configuration.

   Given N, K and the profit value of each of the N billboards, output the maximum profit that can be obtained from the remaining billboards under the conditions given.

3. Table [Codeforces Round 144]

   John Doe has an $n \times m$ table. John Doe can paint points in some table cells, not more than one point in one table cell. John Doe wants to use such operations to make each square subtable of size $n \times n$ have exactly k points.

   John Doe wondered, how many distinct ways to fill the table with points are there, provided that the condition must hold. As this number can be rather large, John Doe asks to find its remainder after dividing by 1000000007 ($10^9 + 7$).

   You should assume that John always paints a point exactly in the center of some cell. Two ways to fill a table are considered distinct if there exists a table cell that has a point in one way and doesn't have it in the other.

   Given integers n, m, k ($1 \leq n \leq 100$; $n \leq m \leq 10^{18}$; $0 \leq k \leq n^2$) – the number of rows of the table, the number of columns of the table and the number of points each square must contain – output the number of valid configurations.

4 Even Tougher

1. Cow Hopscotch [John Pardon, 2010]

   The cows have reverted to their childhood and are playing a game similar to human hopscotch. Their hopscotch game features a line of N ($3 \leq N \leq 250,000$) squares conveniently labeled $1..N$ that are chalked onto the grass.

   Like any good game, this version of hopscotch has prizes! Square i is labeled with some integer monetary value V_i ($-2,000,000,000 \leq V_i \leq 2,000,000,000$). The cows play the game to see who can earn the most money.

   The rules are fairly simple:

   - A cow starts at square "0" (located just before square 1; it has no monetary value).
   - She then executes a potentially empty sequence of jumps toward square N. Each square she lands on can be a maximum of K ($2 \leq K \leq N$) squares from its predecessor square (i.e., from square 1, she can jump outbound to squares 2 or 3 if $K = 2$).

- Whenever she wishes, the cow turns around and jumps back towards square 0, stopping when she arrives there. In addition to the restrictions above (including the K limit), two additional restrictions apply:
- She is not allowed to land on any square she touched on her outbound trip (except square 0, of course).
- Except for square 0, the squares she lands on during the return trip must directly precede squares she landed on during the outbound trip (though she might make some larger leaps that skip potential return squares altogether).

She earns an amount of money equal to the sum of the monetary values of all the squares she jumped on. Find the largest amount of cash a cow can earn.

2. Poklon [COCI 2008]

   Mirko got a set of intervals (size of set bounded by 10^5) for his birthday. There are many games he can play with them. In one of them, Mirko must find the longest sequence of distinct intervals such that each interval in the sequence is in the set and that each interval contains the one that follows in the sequence. Write a program which finds one such longest sequence.