Day 3: Problem Analysis

08.05.2014

Problem A. Business

Problem A. Business

Problem statement

- You are given rooted tree forest, every vertex has profit, and profits can be negative
- You can mark vertex if and only if all its descendants are marked. You need to mark a subset of vertices so that the sum of profits of these vertices is maximized

Solution

- Define $f(v)$ as the maximal profit you can get if you try to mark only vertices from subtree rooted at vertex v
- And $s(v)$ as the sum of all profits of vertices in this subtree
- To compute $f(v)$:
  - Mark vertex v, then $f(v) = s(v)$
  - Don't mark vertex v, then $f(v) = \sum\limits_{u \in C(v)} f(u)$, where $C(v)$ are the children of vertex v
- The answer is sum of all $f(root)$ for every root

Problem B. Coding

Problem B. Coding

Problem statement

- You are given string s of letters A, B and C
- Set T consist of all strings that are built permutating letter in string s
- You need to find $C \subset T$, such that for every $u \in T$, there exist at most one $v \in C$, such that there are three positions in u containing A, B and C, that you can cyclicly shift letters in these positions to make them B, C, A or C, A, B respectively

Solution

- Generate set T. It contains not more than 600 elements
- If s consist of less than three letters, then $C = T$
- Otherwise $|C| \leqslant 6$
- One can make bruteforce to find such set

Problem C. Cyclic Suffixes

Problem C. Cyclic Suffixes

Problem statement

- You are given string s
- You are to find how many cyclic shifts of s are lexicographically less than s

Solution

- Let $n = |s|$ and $z(i)$ be Z-function for ss
- Suppose we know LCP (longest common prefix) of strings α and β, and it equals to p
- $\alpha < \beta$ if and only if $(\alpha_{p+1} < \beta_{p+1})$ or $(|\alpha| = p$ and strings have different length)
- $z(i)$ is LCP of ss and i-th suffix of ss
- Substring of length n of ss is cyclic shift of s
- Compare all cyclic shifts with s using criteria above

Problem D. MagLev

Problem D. MagLev

Problem statement

- You are given special graph and two vertices in it
- You are to find the number of different shortest paths between these vertices

Solution

- Read statement carefully
- Build the graph
- Find the vertices
- Use BFS algorithm to find shortest paths and number of these paths

Problem E. Quadrilaterals

Problem E. Quadrilaterals

Problem statement

- You are given a set of n points on the plane
- You need to find how many different convex quadrilaterals you can make, such that the vertices are in the given set

Solution

- If you choose 4 points, then you can make at most one convex quadrilateral on these points
- So the answer is $\binom{n}{4} - B$, where B is the number of subsets of four points, that no quadrilateral can be made of
- What are these subsets look like? Three of these points form a triangle and the other point is inside of this triangle
- Let's count the number of such subsets

Solution

- Let's fix the point p_0, that lies inside triangle and find how many such triangles there are
- Point lies inside triangle if and only if there is no such line, that all vertices of triangle lie in the same halfplane formed by this line
- So let's divide all vectors $p_0 p_i$ to two sets, S_1 — polar angle in $[0, \pi)$, and S_2 — $[\pi, 2\pi)$
- Get S_2 and rotate every vector by π
- Sort all other points by angle from p_0
- p_0 is inside triangle if three vectors are v_1, v_2, v_3, such that $v_1 < v_2$, $v_2 < v_3$, either $v_1, v_3 \in S_1$ and $v_2 \in S_2$ or $v_1, v_3 \in S_2$ and $v_2 \in S_1$

Solution

- Iterate over all p_0
- Rotate all vectors, polar angle of which are at least π
- Sort vectors by angle
- Solve the problem: you have array of 0-s and 1-s, how many triples of elements so that the middle element differs from two others in linear time

Problem F. Impudence Queue

Problem F. Impudence Queue

Problem statement

- There is a queue. You are given several events, that man comes to the queue and man gets served
- Every one of them has impudence I_p. Man either stands to the end of the queue or finds his friend q, such that $I_p + I_q > 2 \cdot I_r$, for all men r standing after q
- You are to simulate all these queries

Solution

- Just simulate everything, having queue, and for every suffix of the queue knowing mI_x — minimal I_r for $r > x$
- When man comes, just check criteria above for all his friends, and find the best one
- Insert man to the queue, recalculate mI
- That solution works in $O(n^2)$, it fits in time limit
- To get better performance use BST or similar data structure

Problem G. Raccoons

Problem G. Raccoons

Problem statement

- There is a set of n elements
- One can process nonempty subset of this set only if all its nonempty subset were already processed
- A subset of k elements is being processed in $f(k)$ seconds given in input
- You have to choose the order of subsets to be processed so that the $\sum_{i=0}^{S-1} \frac{R(i)}{n}$, where S is the time needed to process subsets, and $R(i)$ is the number of subsets of size $n-1$ already processed

Solution

- If you rewrite the formula like this: $\sum_{i=1}^{n-1} \frac{1}{n}(S - T(i))$, where $T(i)$ is the time when i-th set of size $n - 1$ has been processed
- You see that sum is maximal when the sum of $T(i)$ is minimized, so you always have to do it greadily
- Try to get new subset of size $n - 1$ in minimal time
- Constructive solution: print all numbers from 1 to $2^n - 2$ in binary, you can see, that this always minimizes the sum of $T(i)$

Problem H. Robots

Problem H. Robots

Problem statement

- You are given two strings s and t consisting of four different letters: A, T, G and C
- You are to find such cyclic shift, so that the number of indices containing the same letter in these strings is maximal possible

Solution

- Let $f_c(h)$ be the number of indices, that both strings have letter c in this position, when s is cyclicly shifted h times
- Mathematically $f_c(h) = |\{i \mid t_i = c \wedge s_{(i+h) \bmod |t|} = c\}|$
- And $f(h)$ be the number that you need to maximize $f(h) = f_A(h) + f_T(h) + f_G(h) + f_C(h)$

Solution

- So let's solve number for a particular letter c
- Replace all entries of c in s and t to 1, and all other letters to 0
- Let's look at ss (s concatenated with itself), any substring of s of length $|t|$ is cyclic shift of s
- Make two polynomials $P(x) = \sum\limits_{i=0}^{|ss|-1} s_i x^i$ and $Q(x) = \sum\limits_{i=0}^{|t|-1} t_i^r x^i$, where t^r is reversed t

Solution

- Look at $(P \times Q)(x) = \sum_{i=0}^{|ss|+|t|-2} a_i x^i$
- And see that $a_{h+|t|-1} = f_c(h)$
- Use Fast Fourier Transform to multiply the polynomials
- Deeper analysis of FFT algorithm can help to get the algorithm that makes only 3 FFT calls

Problem I. Sums

Problem I. Sums

Problem statement

- You are given n positive numbers
- You have to choose k of these numbers, so that the minimal number, that you can't represent as the sum of some of these k chosen numbers is the largest possible

Solution

- Suppose you have chosen several numbers, and only numbers you can represent are such y, so that $1 \leqslant y < p$
- Let's try to choose one more number, say x, and see how the set of representable numbers change
  - if $x > p$, then you still can't represent p as the sum of chosen numbers
  - otherwise, you can represent numbers from 1 to $p + x - 1$, and only them

Solution

- If you choose the numbers in nondecreasing order, then while $x \leqslant p$, p increases
- And when x becomes larger than p, then p doesn't change anymore
- Gready approach: while there is a number x, that's not chosen and $x \leqslant p$, just choose the maximal one among them
- Otherwise choose any number

Problem J. Psychic Test

Problem J. Psychic Test

Problem statement

- You are given n
- You are to find such A and B so that the sequences

$$a_i = (1 + Ai) \mod n$$

and

$$b_i = B^i \mod n$$

have longest common prefix of different elements

Solution

- There is always solution of length 2
- $a_1 = b_1 \Rightarrow (1 + A) \equiv B \pmod n$
- $a_2 = b_2 \Rightarrow (1 + 2A) \equiv (1 + A)^2 \Rightarrow 0 \equiv A^2 \pmod n$
- $a_k = b_k \Rightarrow (1 + kA) \equiv (1 + A)^k \Rightarrow 0 \equiv A^2 Q(A) \pmod n$, where $Q(A)$ is polynomial
- So if n divides A^2, then we get two equal sequences
- What is the length of period? It's $\frac{n}{GCD(n,A)}$
- So we need to find A, so that $A \not\equiv 0 \pmod n$, $A^2 \equiv 0 \pmod n$ and $GCD(n, A)$ is minimized
- $A = \prod_i p_i^{\lceil \frac{\alpha_i}{2} \rceil}$, when $n = \prod_i p_i^{\alpha_i}$