ICPC 2014 Preliminary problem presentation

F - Runway Planning

Solution

Just do it.

I - Trainstation Tunnel of Awesomeness

Problem description

Simulate lots of stuff.

Solution

Read and program carefully!
Make use of grids, but don't exaggerate (e.g. loop 6 times per tick over all gridpoints and subsequently loop over each person per tick).

J - Word Search

Solution

Brute force.

For every word:

- Consider every grid letter as a potential start.
- Check all directions.

Watch out for:

- Palindromes (see sample #4)
- Words of length 1

B - Failing Components

Solution

The time at which a component fails is the *shortest path* to it from the initial component that fails. Use Dijkstra's shortest path algorithm.

D - Lift Problems

Solution

Dynamic programming

Let $A(i)$ be the least possible amount of anger if and when the lift stops at floor i. Then:

$A(i) = \min_{j<i} A(j) + \sum_{k=j}^{i-1}(i-k) * s_k + \sum_{k=i}^{n} s_k$ (j is the previous stop)

Compute $A(1), A(2), \ldots, A(n)$.

Don't recompute each sum for every j.

A - Choosing Ice Cream (1/2)

Problem description

Find the smallest t such that k^t is divisible by n.

Solution

Method 1:

- Compute $k, k^2, k^3, \ldots$ modulo n by repeated multiplication.
- Stop when you get zero.
- The answer is at most $\log_2 n$, which is at most 29, so stop when you get there.

A - Choosing Ice Cream (2/2)

Solution

Method 2:

- Determine the prime factorization of n: $n = \prod_i p_i^{a_i}$.
- For every prime factor p_i, determine its multiplicity b_i in k.
- In k^t, the multiplicity is tb_i.
- k^t is divisible by n iff $tb_i \geq a_i$ for all i.
- $t = \max_i \lceil a_i / b_i \rceil$
- Unbounded if $b_i = 0$.

H - Talent Selection

Solution

If you want to get x favourites through:

- Give the most points to the top x favourites *in reverse order*.
- Give the next most points to the other favourites and the top $s - x$ non-favourites (any order).
- Give the remaining points to the others *in reverse order*.
- Check if the x top favourites have at least as many points as the bottom $n - f - (s - x)$ non-favourites.

Do a binary search over x.

G - Spy Network (1/2)

Solution

Method 1:

- For every spy, "send" its report to all its contacts.
- Whenever a spy's report is changed, have him resend its report.
- Either "schedule" a spy for resending (breadth-search), or send it immediately (depth-search).
- A report is changed at most $\log r$ times, since a change involves a division.
- Use Euclid's algorithm to compute the gcd in logarithmic time.
- Complexity: $O(m(\log r)^2)$

G - Spy Network (2/2)

Solution

Method 2:

- Identify the strongly connected components (Tarjan's algorithm).
- In each component, determine the report of all the spies based on their internal communication only (it is the gcd of all their reports).
- For each component, determine the final report by recursively asking for the final report of all the components that link to it, and computing the gcd of them.
- Complexity: $O(m \log r)$

E - Pawns of Death (1/3)

Solution

Figure it out logically.
For convenience, let's call a pawn that can (still) move two squares a "joker". Note:

- If there are no jokers, it is a simple matter of parity.
- If the player to move has a joker left, while his opponent doesn't, he can win by using his joker to "set the parity" in his favour.

E - Pawns of Death (2/3)

Solution

Strategy:

- Neutralize your opponent's jokers by moving your pawns forward in those columns.
- If one player can manage to neutralize all jokers before his opponent, he wins.
- If both players neutralize each other jokers at the same time, it is down to parity.

E - Pawns of Death (3/3)

Solution

What if both sides have a joker in the same column?

- If $n > 4$, these columns can be ignored: the player who would win in the absence of these columns, can wait for his opponent to move in one of these columns and respond appropriately in the same column.
  Special case $n = 4$:
- If the number of columns with double jokers is even, the columns can be ignored: the winning player can copy his opponent's move in another column.
- If the number of columns is odd, White wins, since he can set the parity in his favour.

C - Floor Painting of Terribleness

Solution

- Determine grid of lines through vertices;
- Build data structure with for each cell:
  - Pointer to first cell to the right that is outside polygon;
  - Pointer to first cell above that is outside polygon;
- Search for squares of given size and lower left corner;
  Find largest by exponential search on square sizes.

C - Floor Painting of Terribleness