

NCPC 2009

Presentation of solutions

Head of Jury: Jon Marius Venstad
co-Heads of Jury: Fredrik Svensson and Jimmy Mårdell

2010-10-02

Problem authors

- Andreas Björklund (ARM)
- Jimmy Mårdell (Silobreaker)
- Jon Marius Venstad (NTNU)
- Pehr Söderman (KTH)
- Fredrik Svensson (LiU)
- Christian Jonassen (NTNU)

Solution

- Create a map of family relations, where children map to their parents.
- Use another map where each person (in the end) maps to their blood contents. Add only the founder, with blood content $2^{50}L$.
- Do a DFS on each of the claimants to find their blood contents. If a parent is not in the family relations map, the blood content of that parent is 0. Otherwise, calculate the parent's blood content (recursively) and use the result to calculate the child's content.

Solution

- For both pairs of points to be connected, try to connect these by two different shortest paths: one preferring horizontal and one preferring vertical moves. For each of these two do a BFS to connect the other pair of points. Take the minimum over all these combinations.
- Special case when all four points are on the same row or column.

Solution

- Initial probability of the sentry being in a certain node is proportional to that node's degree.
- When the captain walks into a room, his chances of success are reduced by the probability of meeting the sentry in or into that room, and (assuming success) the probability of the sentry being in that room is reset to 0. (The probability of the sentry being in the captain's previous room is also reduced by the probability that they meet on the way.)
- Then propagate the probability distribution for the borg, and repeat until the captain is back on his Galaxy class cruiser.

Solution

- Sort the distances to those cars, to easily find the number of cars between you and some other car (by just looking at its index in the sorted array).
- Let S be the sorted array of length n , and find $M = \max\{p(i+1) - S_i : i \in [0, n-1]\}$.
- This is what you should modify your distance to the queue with, so after this modification your (correct) distance to the next car is $M + S_0$

Solution

- Repeatedly mirroring the grid on its lower row and rightmost column, reveals that the number of squares you fill out before you fill out a new corner is the LCM of $h - 1$ and $w - 1$ (let this be L), plus 1 for the new corner. The pattern will run between top and bottom $\frac{L}{h-1}$ times, and between left and right $\frac{L}{w-1}$ times.
- Some squares will be counted twice in this way. Each sequence from the left to the right edge intersects each sequence from the upper to the lower edge exactly once, except when these sequences are overlapping. This is the case for each of the edge points, of which there are $\frac{L}{h-1} + \frac{L}{w-1} - 1$. Also, everything is counted twice in this way, as the two kinds of sequences are made up of the same squares.
- The final result is $L + 1 - \frac{1}{2} \left(\frac{L}{h-1} - 1 \right) \left(\frac{L}{w-1} - 1 \right)$.

F - Help a PhD candidate out!

Solution

- Read input, check if lines are “P=NP”, and if not parse numbers and add them together. Hard?

Solution

- Everyone wins if there are no cycles of length greater than K among the N boxes. Let P_n^k denote this probability. Obviously $P_n^k = 1$ when $k \geq n$. To find the probabilities when $k < n$ do the following reasoning:
- Open one of the remaining boxes and check the length of the cycle this box is in. With n remaining boxes the probability of this cycle having a length of $m \in [1, n]$ telescopes to $\frac{1}{n}$ by simply opening boxes until the cycle is complete.
- Thus we get $P_n^k = \frac{1}{n} \sum_{i=1}^k P_{n-i}^k$. By letting S_n^k be this sum (without the $\frac{1}{n}$ constant), we can calculate each P_n^k and S_n^k in constant time:
 - $P_n^k = \frac{S_n^k}{n}$
 - $S_{n+1}^k = S_n^k + P_n^k - P_{n-k+1}^k$

Solution

- Try all possible positionings of 2 to 5 pieces relative to the first piece.
- For each relative positioning, try building the pieces in parallel using a DFS. This DFS does not need to backtrack! Doing so will cause a timeout.
- Either of:
 - Keep first piece fixed at first possible location.
 - Prune away positionings which prove to be impossible already before all pieces are placed.

Solution

- Create an integral image ii , i.e.
$$ii[Y, X] = \sum_{x=0}^{X-1} \sum_{y=0}^{y-1} s[y, x].$$
 - Cumulative sum of each row.
 - Cumulative sum of each resulting column.
- Find all areas.
 - for x, y, w, h
 - $area(x, y, w, h) =$
 $ii[y + h, x + w] - ii[y + h, x] - ii[y, x + w] + ii[y, x]$
- Bucket sort to find median.
 - $0 \leq \text{Statisticians}/\text{area} \leq 10$.
 - Precision 10^{-3} .
 - $10000 * 10^3 + 1$ buckets.

Solution

- Represent a skyline (zero or more buildings) as a list of line segments. This list is sorted in the horizontal direction. Its area can easily be calculated in linear time by summing the area under each line segment.
- Merge two such skylines (the total skyline and each new building) in linear time, inserting new vertices where line segments intersect.
- Add the buildings in the given order, and find the visible area of each new building by comparing the area difference of the skylines with and without the building to the area of the building itself. (If the skyline's area did not increase, no part of the new building was visible.)

Solution

- For each node with four edges there are three connection alternatives, two of which you can always choose between. Sort these nodes in descending order based on the difference between the two best alternatives.
- Use a DisjointSet over the edges to keep track of components of the circuit, by taking the union whenever two edges are connected. Connecting two edges with the same representative will cause a cycle, and can only be done in the last node.
- Pick first the nodes with only two edges, and then the nodes with four edges, in the order found by the sorting. Choose the best available alternative for each of these.
- If you still have disjoint circuits after all nodes have been processed, look at the nodes where these circuits intersect (again in order) and join the circuits by changing connection alternatives.