Solutions

Benelux Algorithm Programming Contest 2015

Universiteit Leiden

October 24, 2015

A: Freight Train

- Do a binary search for the minimum length of the longest train to Luxembourg.
- For maximum length L solve the problem using a greedy algorithm.
- At any point look for the next wagon heading to Luxembourg.
- If it is within the next block of N wagons, make a train segment of exactly N wagons and send it to Luxembourg.
- If not, then make a train segment with all the wagons before the Luxembourg wagon and send it to The Netherlands.

B: Physical Music

- For $i = 1$ to N
  check if single i in Single Top N has overtaken
  any other single j, as compared to Download Top N
- if so, i is available as CD-single
- $O(N^2)$ *should be* too slow
- for $O(N)$,
  - for single $i = 1, \ldots, N$ in Single Top N, keep track of highest position in Download Top N we we have not seen yet
  - for single $i = N, \ldots, 1$ in Single Top N, keep track of highest position in Download Top N we have seen
  - or use efficient set datastructure

C: Godzilla

Kill Godzilla with your moving Mechs, minimizing the number of destroyed sectors.

Answer: Flood fill (BFS) with multiple sources.

- Each mech is a source for a BFS
- All mechs share the same 'flood'
- Godzilla dies when it is in range of the 'flood'

D: Hotels

- Too many floors to make a graph of all floors.
- We can compress the graph:
  - Make a graph with one vertex for every floor that has an elevator.
  - Distance between two vertices is the minimum number of stairs one has to take.
  - At most 10^5 vertices and $2 \cdot 10^5$ edges.
  - Dijkstra $\mathcal{O}(m \log(n))$ from bottom floor is fast enough.
- Alternative solution:
  - Make a graph with one vertex for every elevator.
  - Distance between two elevators is the minimum number of stairs one has to take.
  - At most 100 vertices and 100^2 edges.
  - Dijkstra, Bellman–Ford, Floyd–Warshall.
- Second solution is easier to implement.

E The Kings Walk — Trinomial Triangle

12	18	16	10	4
5	7	6	3	1
2	3	2	1	3
1	1	1	2	5
1	♔	1	2	4

			0		1	0			
		0		1	1	1			
			1	2	3	2	1		
	1		3	6	7	~~6~~ 5	3	1	
1	4		10	16	~~19~~ 18	~~16~~ 12	10	4	1

F: Map Colouring

Deciding graph colourability with $k \in \{1, 2, 3, 4, \mathrm{many}\}$ colours and $n \leq 16$ vertices.

Many possible correct answers:

- Brute force with pruning.
- Worst case $O(2^n m)$.
  - One and two colouring can be done greedily.
  - Try all possible splits of the graph into two sets.
  - Set 1 for colours 1 and 2, set 2 for colours 3 and 4.
- Multi-restart randomized greedy colouring.
- $O(3^n m)$ dynamic programming over subsets.

G Mario

- For all testcases in the input, Mario arrives within $O(W^2)$ steps or never.
- Simulation: maintain position of all boats at each step.
- It is optimal to move to the right if possible.
- Be careful with transition on $t = 0.5 + int$.
- Provably correct solution in $O(n^2 \cdot W)$ time

H: Museum

Count number of triangles that can be formed in the room, given that the lines that form the triangle can be blocked.

Answer: For each pillar, find the pillars with which it can form a triangle; $O(n^3)$

- Take a pillar
- Find two other pillars, with at least one pillar on a different wall then the other two
- These pillars are a potential triangle:
  - Determine whether a line of this triangle intersects the pedestal
  - If there is no intersection: $++$triangles
- Continue with the next pillar

If interested, ask jury for $O(n^2)$ solution (or perhaps $O(n)$)

I Six Degrees

- Unweighted undirected graph with $n \leq 3000$ nodes and $m \leq 30,000$ edges
- Note: "worst-case" version of six degrees of separation
- So, determine eccentricity (longest shortest path length) for each node using a BFS
- Decide based on percentage of nodes with eccentricity ≥ 7
- Total of n BFSes in $O(m)$ time
- Given that the graph is either small or sparse, do not use an adjacency matrix (that will give TLE), but use adjacency lists

J: Tour de France

Directed TSP with max in- and out-degree 2. Jury solution:

- Adding a route to a tour forbids the other incoming route of the target city
- Forbidding a route forces the other outgoing route of the source city to be part of the tour
- A 'chain' of such take-forbid-take actions must consist of at least 4 routes and has 2 'states': take or forbid first route
- Try all possible states of all chains: $O(2^{|E|/4}) = O(2^{|V|/2})$
- Can implement with just trying all outgoing edges combined with this pruning

J: Tour de France

Participant solution:

- Try all forward paths of length $n/2$: $2^{n/2}$ such paths because of the degree restriction
- Remember vertices visited (with bitmask); store best mask-distance pairs
- Also traverse backwards paths, match paths, add best pairs

K: Wipe Your Whiteboards

Find minimal positive A and B with $A \cdot R + B \cdot S = T$:

- Compute extended Euclidean algorithm on R and $-S$
- Results in $g = \gcd(R, -S)$ and A', $-B'$
- By Bezout's identity, result is for some k:
- $A = A' + k \cdot \frac{R}{g}$ and $B = B' + k \cdot \frac{S}{g}$
- $k = \max\left(\left\lceil \frac{1-A'}{-S/g} \right\rceil, \left\lceil \frac{1-B'}{R/g} \right\rceil\right)$
- Can also do while-loop to find k, is fast enough
- Watch out for overflows (use longs). Don't do anything linear.

L Zanzibar

- Straight-forward
- Sum difference between population and (population $\times$ 2) for all positions i and $i + 1$
- Read the book by John Brunner