Solution outlines

BAPC Preliminaries 2011

October 1, 2011

F – Dividing the Loot

- Very simple greedy solution
- Simply pick the N/(P+1) most valuable items

E – Rolling Dice

- Basic simulation
- Hardest part is keeping track of orientation
- Do not roll one square at a time!
  - Orientation the same after rolling 4 times in same direction

A – Stifling the Mutiny

- Ad-hoc solution
- Every ship must have one loyal pirate
  - Place a disloyal pirate every three ships:

- For the rest:
  - Place as many disloyal pirates as possible on last ship
- Formula for n ships and k pirates:

$$F(n, k) = \begin{cases} k/2 & \text{if } n = 1 \\ k - n & \text{if } k < n + (n+4)/3 \\ (k - (n-2)/3)/3 & \text{otherwise} \end{cases}$$

Benelux Algorithm Programming Contest 2011

H – Stealth Ninja

- States have period of 16 sec
  - 8 sec after dividing by two
- Compute states (x, y, t) for which ninja is unseen (t mod 8)
- Check whether ninja succeeds using BFS

D – Polly wants a cracker

- Compute Levenshtein distance for every pair of words using DP
- Compute minimum weighted matching
  - Brute-force fast enough

- In case of brute-force: do not recompute distances!

B – RNG in Reverse

- First rewrite as: $ax^2 + bx + c = 0 \bmod 2^n$
- x is a solution for $n \rightarrow$ ($x \bmod 2^{n-1}$) is a solution for $n-1$
- Hence, if x is unique solution for $n-1$
  - x is possible solution for n
  - $x + 2^{n-1}$ is possible solution for n

- Maintain solutions for increasing n
  - If solution is unique, continue
  - Otherwise we can never get a unique solution

- Be careful with overflow!

C – Attack of the Giant n-pus

- Make complete bipartite graph for pirates & tentacles
  - Weight of edge is required time
- Perform BS over edge weights
- For a given weight w
  - Remove edges with weight > w
  - Compute maximum bipartite matching
- Find smallest w such that |Matching| = #tentacles
- Add time from captain to head of n-pus

Benelux Algorithm Programming Contest 2011

J – Shuriken Game

- Solution using Dynamic Programming
- F[stacksize][maxnum][prevmove] is too slow
- Instead, for F[stacksize][maxnum] store:
  - -1, if there are multiple winning moves
  - 0, if there is no winning move
  - x, if x is the only winning move
- Player wins if:
  - F[stacksize][maxnum] = -1 or
  - F[stacksize][maxnum] ≠ prevmove

Benelux Algorithm Programming Contest 2011

I - In and Out

- Split up nodes of sentries and connect with directed edge
- Compute shortest path using Dijkstra or Bellman-Ford
- Along shortest path:
  - Edges between split nodes: Reverse direction
  - Other edges: Negate weight in opposite direction
- Compute another shortest path using Bellman-Ford
- Dijkstra also possible after reweighting

G – Secret Island Base

- Find largest inscribed circle of polygon
- For every combination of 3 points/edges
  - Find circle(s) touching the 3 points/edges
  - Check if circle fits (and is in polygon)
- 4 different combinations
  - 3 points (easy)
  - 3 edges (easy)
  - 2 points, 1 edge (hard)
  - 2 edges, 1 point (hard)