BAPC 2009 Preliminary Rounds

The Solutions

2008-10-03

F. Box Village

F. Box Village

$$\text{print} \left\lceil \frac{n}{\lfloor d/d_1 \rfloor \cdot \lfloor w/w_1 \rfloor} \right\rceil ;$$

D. Family Politics

- The family forms a circuit.
  - ‘beach’ = true, ‘museum’ = false.
  - Children are free inputs.
  - Adults are NAND gates.
  - Gran is the output.

D. Family Politics

- The family forms a circuit.
  - ‘beach’ = true, ‘museum’ = false.
  - Children are free inputs.
  - Adults are NAND gates.
  - Gran is the output.
- Circuit satisfiability

D. Family Politics

- The family forms a circuit.
  - 'beach' = true, 'museum' = false.
  - Children are free inputs.
  - Adults are NAND gates.
  - Gran is the output.
- Circuit satisfiability, NP-Hard.

D. Family Politics

- The family forms a circuit.
  - 'beach' = true, 'museum' = false.
  - Children are free inputs.
  - Adults are NAND gates.
  - Gran is the output.
- Circuit satisfiability, NP-Hard.
- Brute force: try all possible inputs (children)

B. Hedge Maze Relay Race

- Throwing the baton doesn't help, someone still needs to walk to the finish line, he might as well carry the baton.

B. Hedge Maze Relay Race

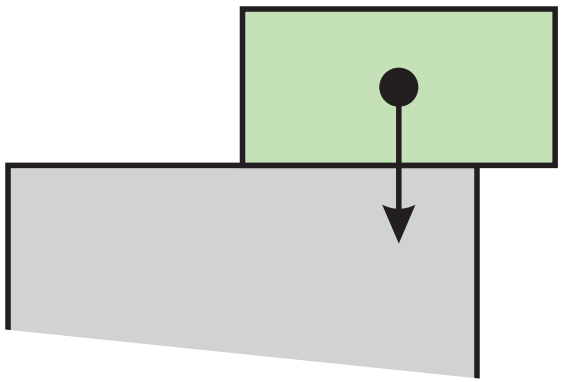
- Throwing the baton doesn't help, someone still needs to walk to the finish line, he might as well carry the baton.
- Breadth first search 4 times (depending on team)
  1. crossing only grass
  2. crossing grass and flower beds
  3. crossing grass and gates
  4. crossing grass, gates and flower beds

G. Typechecker

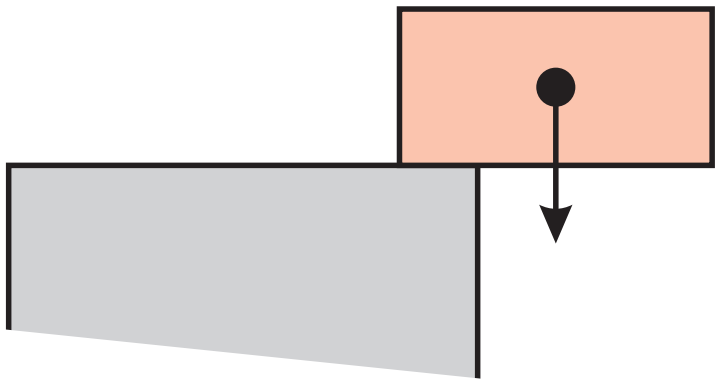
- An exercise in parsing.
- Calculate return types starting at the leafs.
- If a matching overload exists, use it.

C. Brick Stacking

- A stack is stable if its center of mass is supported.



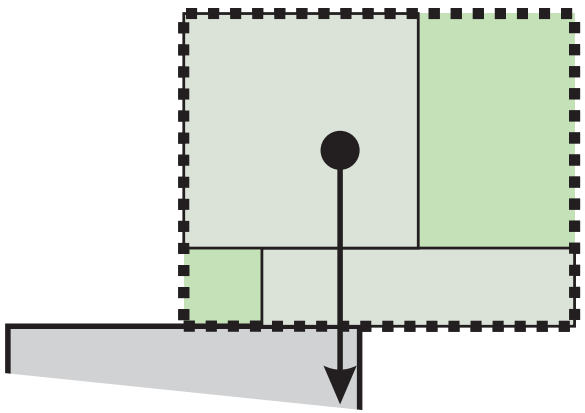
stable



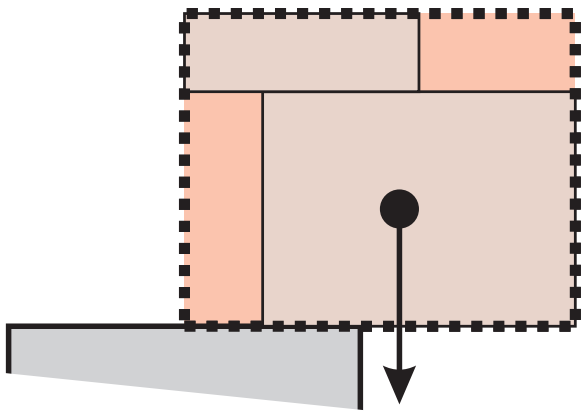
unstable

C. Brick Stacking

- A stack is stable if its center of mass is supported.
- Work from the top down, fusing stable stacks together.



stable



unstable

G. Blueprint

- First convert everything to meters.
- Calculate perimeter and area of polygon.

G. Blueprint

- First convert everything to meters.
- Calculate perimeter and area of polygon.

$$p = \sum \|p_{i+1} - p_i\| \qquad a = \frac{1}{2} \sum p_{i+1} \times p_i$$

G. Blueprint

- First convert everything to meters.
- Calculate perimeter and area of polygon.

$$p = \sum \|p_{i+1} - p_i\| \qquad a = \frac{1}{2} \sum p_{i+1} \times p_i$$

- Solve $2 \cdot a \cdot \text{scale}^2 + 3 \cdot p \cdot \text{scale} = \text{material}$.

A. All Out Arctic Warfare

- a.k.a. Rock-Paper-Scissors

A. All Out Arctic Warfare

- a.k.a. Rock-Paper-Scissors
- Suppose you start with P. rockus.

A. All Out Arctic Warfare

- a.k.a. Rock-Paper-Scissors
- Suppose you start with P. rockus.
  1. if the opponent's line starts with S, kill it with special attack and continue.

A. All Out Arctic Warfare

- a.k.a. Rock-Paper-Scissors
- Suppose you start with P. rockus.
  1. if the opponent's line starts with S, kill it with special attack and continue.
  2. otherwise kill the opponent's penguin with the gun.

A. All Out Arctic Warfare

- a.k.a. Rock-Paper-Scissors
- Suppose you start with P. rockus.
  1. if the opponent's line starts with S, kill it with special attack and continue.
  2. otherwise kill the opponent's penguin with the gun.
  3. then it is the opponent's turn.

A. All Out Arctic Warfare

- a.k.a. Rock-Paper-Scissors
- Suppose you start with P. rockus.
  1. if the opponent's line starts with S, kill it with special attack and continue.
  2. otherwise kill the opponent's penguin with the gun.
  3. then it is the opponent's turn.
  4. if the opponent has a P, your R will be killed, as well as your next penguin.

A. All Out Arctic Warfare

- a.k.a. Rock-Paper-Scissors
- Suppose you start with P. rockus.
  1. if the opponent's line starts with S, kill it with special attack and continue.
  2. otherwise kill the opponent's penguin with the gun.
  3. then it is the opponent's turn.
  4. if the opponent has a P, your R will be killed, as well as your next penguin.
  5. if the opponent has something else, only your R will be killed.

A. All Out Arctic Warfare

$N(i, p)$ is number of penguins needed when starting with p.

$$N(i) = \min_p N(i, p)$$

$$N(i, p) = \begin{cases} N(i+1, p) & \text{if } p \text{ beats opp}(i) \\ N(i+1)+2 & \text{if opp}(i+1) \text{ beats } p \\ N(i+1)+1 & \text{otherwise} \end{cases}$$

$$N(n, p) = 1$$

A. All Out Arctic Warfare

$N(i, p)$ is number of penguins needed when starting with p.

$$N(i) = \min_p N(i, p)$$

$$N(i, p) = \begin{cases} N(i+1, p) & \text{if } p \text{ beats opp}(i) \\ N(i+1)+2 & \text{if opp}(i+1) \text{ beats } p \\ N(i+1)+1 & \text{otherwise} \end{cases}$$

$$N(n, p) = 1$$

- Dynamic Programming makes this fast (start from the back of the line)

G. Word Search Puzzle

- Built trie for dictionary.
- For each cell, search in all directions.
- Is there a match in the trie?
- Mark all found words.

E. Box City

- For each item, place it in the box.
- Try at bottom/right coordinates of previous items.

E. Box City

- For each item, place it in the box.
- Try at bottom/right coordinates of previous items.

code

for each y in 0, $\langle$bottom of previous items$\rangle$
  for each x in 0, $\langle$right of previous items$\rangle$
    try to fit item at (x, y)

E. Box City

- For each item, place it in the box.
- Try at bottom/right coordinates of previous items.

code

for each y in 0, ⟨bottom of previous items⟩
  for each x in 0, ⟨right of previous items⟩
    try to fit item at (x, y)

- Too slow! $O(\#\text{items-in-box}^3)$ per item.

E. Box City

faster code

for each y in 0, $\langle$bottom of previous items$\rangle$

1. find items vertically intersecting $[y; y + h)$ (only these matter for this y coordinate.)
2. find a horizontal interval of size w that doesn't intersect these items.
3. if found, then place item there.

See you at the BAPC

BAPC 2009
October 17
Groningen