Code Jam 2015 - Round 1C

Analysis: Brattleship

Once we first make a hit, it will take at least W-1 more moves to win, since we have to hit the remainder of the ship.

If there's still more than one possible position for the ship, then the little brother will get at least one more opportunity to answer "miss." To limit the number of additional misses to one, we make moves adjacent to the hits we already have. If we get a "miss," then we will have a row of "hits" with a miss at the end, so we will know the exact location of the ship.

Now, the little brother may as well try to maximize the number of misses we make until we first get a hit. He can't control what cells we name, so all he can do is answer "miss" until we make a guess which must unavoidably be a hit. So we need to find a pattern of guesses that uses as few cells as possible, and such that each possible ship position is covered by one of them. To do this, we use a pattern that chooses every W^{th} cell of each row.

The total number of guesses is then R * floor(C/W) for the pattern, plus W-1 to hit the remainder of the ship, plus 1 more guess if there is more than one possibility for the position of the ship, which occurs if C is not an exact multiple of W. Below is a sample code in C that implements this solution:

```
#include <stdio.h>

int main() {
  int T, TC, R, C, W, score;

  scanf("%d", &T);
  for (TC = 1; TC <= T; TC++) {
    scanf("%d %d %d", &R, &C, &W);

    // The R * floor(C/W) for the guess pattern.
    score = R * (C / W);

    // Plus W-1 to hit the remainder of the ship.
    score += W - 1;

    // Plus 1 more guess if there is more than one
    // possibility for the position of the ship,
    // which occurs if C is not an exact multiple of W.
    if (C % W) score++;

    printf("Case #%d: %d\n", TC, score);
  }
}
```

tos.lunar wrote a solution for this question using Haskell, which you can download from the scoreboard.

A recursive search of the game tree, simulating all choices for our moves and the little brother's moves, will also work for the small input.