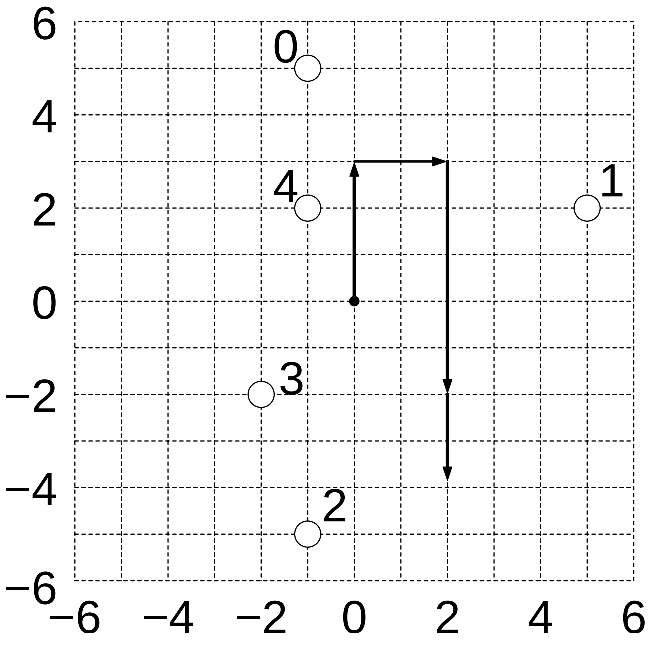
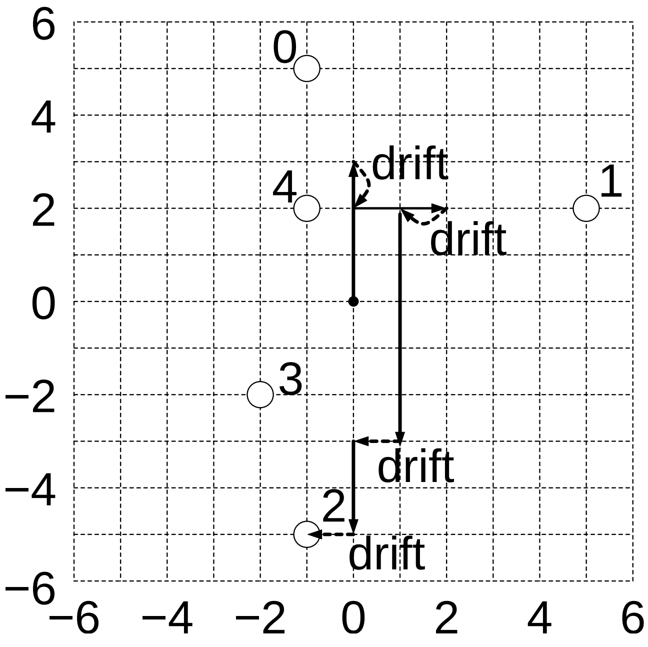
Navigate: Solution

Problem: You are given a set of reef points and a list of navigation segments, each of the form {N,S,E,W} + distance. After each segment the ship may drift in any direction or not at all.

Example: [N 3], [E 2], [S 5], [S 2]





Approach

Hardest: Simulate the motion of the ship, and use backtracking to generate **all possible drifts**. (This will take way too long!)

Easier: Construct a 2-dimensional array representing the ocean. Place a **marker** in each place at each **possible** location of the ship. Move all the markers to simulate the ship motion.

To simulate drift, at the end of each segment, copy each marker location to its **4 surrounding neighbors**.

Easiest: Keep track of the nominal position of the ship (without drifting). When processing the s-th segment, check whether there are **any reefs within distance s-1**, where "distance" is defined as follows:

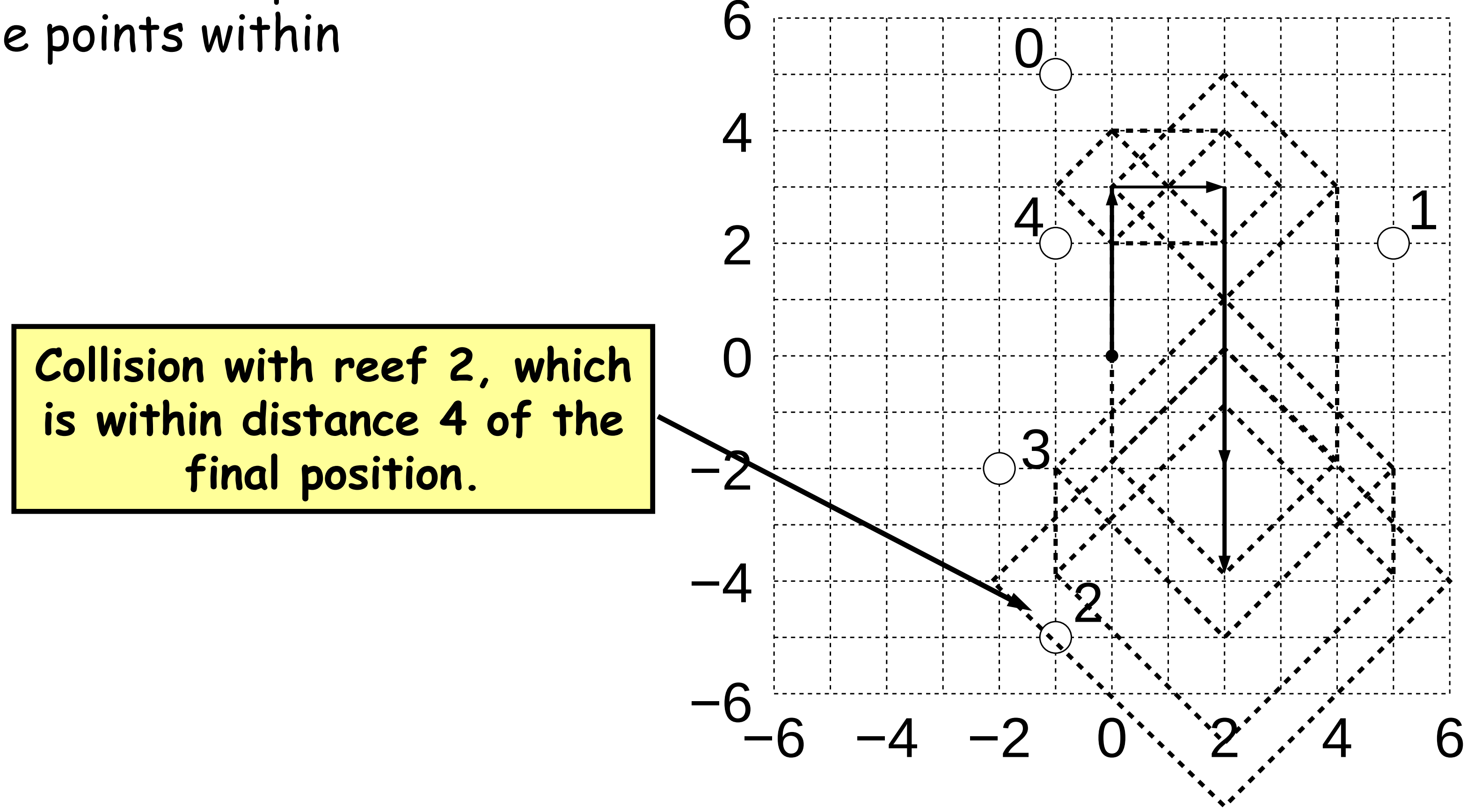
$$\mathrm{dist}(p, q) = |p_x - q_x| + |p_y - q_y|$$

This is called the **L_1 distance**.

Example

Example: [N 3], [E 2], [S 5], [S 2]

The diamond shape indicates the points within



Pseudo-code

```
x ← y ← 0
for (s ← 1 to nSegs) {
    for (j ← 0 to segLength[s]) {                 // process a segment
        for (i ← 1 to nReefs) {                   // check for collision
            distToReef ← | x - reefX[i] |  +  | y - reefY[i] |;
            if (distToReef ≤ s)                   // within drift distance?
                return i;                         // ...yes, return collision
                                                  // update location
            if (j < segLength[s]) {               // stop at segment end
                x += (value based on segmentDirection[s]);
                y += (value based on segmentDirection[s]);
            }
        }
    }
}
return -1;                                        // no collision detected
```