SevenSeas.java

```java
import java.io.*;
import java.util.*;

public class SevenSeas implements Comparable<SevenSeas>{

    static int[] xAxis = { 1, 0, -1, 0, 1, 1, -1, -1};
    static int[] yAxis = { 0, 1, 0, -1, -1, 1, -1, 1};
    static ArrayList<Integer> rocks = new ArrayList<>();
    HashSet<Integer> ships;
    ArrayList<Integer> crashes;
    int ship, heuristic;

    SevenSeas (SevenSeas s){
        ships = new HashSet<>(s.ships);
        ship = s.ship;
        crashes = new ArrayList<>( s.crashes );
    }

    public SevenSeas(HashSet<Integer> ships, int ship) {
        this.ships = ships;
        this.ship = ship;
        crashes = new ArrayList<>();
    }

    @Override
    public int compareTo(SevenSeas o) {
        return heuristic - o.heuristic;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;

        SevenSeas sevenSeas = (SevenSeas) o;

        if (ship != sevenSeas.ship) return false;
        if (ships != null ? !ships.equals(sevenSeas.ships) : sevenSeas.ships != null) return false;
        return crashes != null ? crashes.equals(sevenSeas.crashes) : sevenSeas.crashes == null;
    }

    @Override
    public int hashCode() {
        int result = ships != null ? ships.hashCode() : 0;
        result = 31 * result + (crashes != null ? crashes.hashCode() : 0);
        result = 31 * result + ship;
        return result;
    }

    int distance(int a){
        int x = a >>> 4;
        int y = a & 15;
        int r = Integer.MAX_VALUE, x1, y1, aux;
        for (int tmp : rocks){
            x1 = tmp >>> 4;
            y1 = tmp & 15;
            aux = Math.abs(x1 - x) + Math.abs(y1 - y);
            if ( aux < r )
                r = aux;
        }
        for (int tmp : crashes){
            x1 = tmp >>> 4;
            y1 = tmp & 15;
            aux = Math.abs(x1 - x) + Math.abs(y1 - y);
            if ( aux < r )
                r = aux;
        }
        return r;
    }

    void moveEnemies(){
        int x = ship >>> 4;
```

```
        int y = ship & 15;
        int r2 = 0, aux2;
        HashSet<Integer> ships2 = new HashSet<>();
        for (int tmp : ships){
            int x1 = tmp >>> 4;
            int y1 = tmp & 15;
            int r = Integer.MAX_VALUE, aux, X = x1, Y = y1;
            for (int i = 0; i < 8; i++){
                int a = x1 + xAxis[i];
                int b = y1 + yAxis[i];
                if ( a < 9 && a >= 0 && b < 8 && b >= 0){
                    aux = Math.abs( a - x ) + Math.abs( b - y );
                    if ( aux == 0 ){
                        ship = -1;
                        return;
                    }
                    if ( aux < r ) {
                        r = aux;
                        X = a;
                        Y = b;
                    }
                }
            }
            int tmp2 = X * 16 + Y;

            aux2 = distance(tmp2);
            if (aux2 != 0) {
                if ( !ships2.add(tmp2) ) {
                    crashes.add(tmp2);
                    ships2.remove(tmp2);
                    r2 -= aux2;
                }
                else {
                    r2 += aux2;
                }
            }
        }
        ships = ships2;
        heuristic = r2;
    }

    static void printMap(SevenSeas s){
        char[][] map = new char[9][8];
        for (int i = 0; i < 9; i++) {
            for (int j = 0; j < 8; j++) {
                map[i][j] = '.';
            }
        }
        for (int tmp : rocks){
            int a = tmp >>> 4;
            int b = tmp & 15;
            map[a][b] = '#';
        }
        for (int tmp : s.ships){
            int a = tmp >>> 4;
            int b = tmp & 15;
            map[a][b] = 'E';
        }
        for (int tmp : s.crashes){
            int a = tmp >>> 4;
            int b = tmp & 15;
            map[a][b] = 'C';
        }
        int a = s.ship >>> 4;
        int b = s.ship & 15;
        map[a][b] = 'S';

        for (int i = 0; i < 9; i++) {
            for (int j = 0; j < 8; j++) {
                System.out.print(map[i][j]);
            }
            System.out.println();
        }
        System.out.println();
    }
```

```
    static int search(SevenSeas sevenSeas){
        HashSet<SevenSeas> hs = new HashSet<>();
        PriorityQueue<SevenSeas> pq = new PriorityQueue<>();
        pq.add(sevenSeas);
        SevenSeas u;
        while (!pq.isEmpty()){
            u = pq.poll();
            //printMap(u);
            int x = u.ship >>> 4;
            int y = u.ship & 15;
            for (int i = 0; i < 8; i++){
                int a = x + xAxis[i];
                int b = y + yAxis[i];
                if ( a < 9 && a >= 0 && b < 8 && b >= 0 ) {
                    int aux = a * 16 + b;
                    if ( !rocks.contains( aux ) && !u.ships.contains( aux ) && !u.crashes.contains( aux ) ){
                        SevenSeas s = new SevenSeas(u);
                        s.ship = aux;
                        s.moveEnemies();
                        if (s.ship != -1) {
                            if (s.ships.size() == 0)
                                return 1;
                            if (!hs.contains(s)) {
                                pq.add(s);
                                hs.add(s);
                            }
                        }
                    }
                }
            }
        }
        return 0;
    }

        public static void main(String[] args) throws IOException{
        BufferedReader br = new BufferedReader( new InputStreamReader( System.in ) );
        //BufferedReader br = new BufferedReader( new FileReader(
          //      new File( "/home/juan/Documents/Maratones/Rogue/src/input.txt" ) ) );
        int T;
        char[][] map = new char[9][8];
        T = Integer.parseInt( br.readLine() );
        for ( int c = 0; c < T; c++ ) {
            for (int i = 0; i < 9; i++)
                    map[i] = br.readLine().toCharArray();
            br.readLine();
            int ship = 0;
            HashSet<Integer> ships = new HashSet<>();
            rocks.clear();
            for (int i = 0; i < 9; i++){
                for (int j = 0; j < 8; j++){
                    if (map[i][j] == '#')
                        SevenSeas.rocks.add(i * 16 + j);
                    else if (map[i][j] == 'E')
                        ships.add(i * 16 + j);
                    else if (map[i][j] == 'S')
                        ship = i*16+j;
                }
            }
            SevenSeas s = new SevenSeas(ships, ship);
            if (SevenSeas.search(s) == 1)
                System.out.println("I'm the king of the Seven Seas!");
            else
                System.out.println("Oh no! I'm a dead man!");
        }
    }
}
```