Instant Insanity Project
Math 482 – TTH 12:30PM-1:45PM
Professor John Dye
Contributors:

Programming Instant Insanity

Math 482 – TTH 12:30PM-1:45PM
Professor John Dye

Contributors: Auhbon Amiri, Roee Distiller, Vaataannnaak Keo, Baplo Leepee, Jorge Martini, Sarkis Mikaelian, Pulok Rahman

Preface:

Instant Insanity is a “fun” puzzle about blocks and colors. It’s asymptotic time complexity grows exponentially. To solve the puzzle, the blocks must be arranged on top of eachother in one stack/column with each side showing each color exactly once. In other words, from each direction you face the stack, you should see a column in which no color repeats and every color is represented. With puzzles that have no solution, a minimal obstacle can be determined. This is a representative smallest set of cubes which which are not compatible with the original set. This minimal obstacle set is found by iterating down through each set size until a size is determined to not have any possible collisions- regardless of which cubes are chosen. Then, the minimal obstacle is considered to be any of the smallest sets found which indeed did result in an unstackable set (ie had any collisions and is one larger than the set size in which no collisions could be found). When searching for sets that are not compatible, there is no concern in terms of finding a subsolution- the only concerns are collisions. In other words, we aren’t concerned with whether or not each color appears *exactly* once in each column; we are just concerned with whether or not the colors appear *more* than once in each column.

Puzzle Iteration and Finding the Solution

Python is currently one of the most widely used languages in the industry, therefore we decided it is best to use in this project as we are most comfortable with it. Here follows an explanation of the components of the program.

cubes_generator(colors)
This function constructs the puzzle by building cubes out of the color set that is given and stacks the cubes on top of one another.

find_min_obs(cubes)
This function uses brute force to find the minimal obstacle. It sets up all the combinations of all the subsets of the puzzle. It begins by checking the puzzle that’s a size of 30 cubes. If it has no solution then it goes down and checks every 30 choose k combination until a

every subset of a combination all have solutions. It will have then found the minimal obstacle. This function relies on the solve_stack function to operate.

solve_stack(cubes, visited, solution_num)
Once called, this function will follow a Depth-First-Search for a set of cubes, along with having a known count of visited cubes (to avoid incorrect overlap). The first iteration “i” loops through all of the cubes and checks each of the three possible pairs of the current cube. If it finds a half solution, it will begin to check the puzzle for second compatible half solution to solve the stack of cubes. It is written arbitrarily enough to work for any size stack of cubes although the computational requirements will increase exponentially. If it does not find a half solution after the 2 pairs of colors for the first cube, it will break out of the iteration loop of the Depth-First-Search and finalize with no solution found.

Program Structure:

- **Pseudocode: (Not finished. Based on Cormen’s pseudo -Pablo)**
    - **(to make an arrow its “<-- “)**

```
find_min_obs(cubes)
i ← 0
Multiplicity ← [array of size 30]
Half solution flag ← False
Half solution counter ← 0
iterate through all the cubes
Do if  -1 < i < length of cubes
            backwards iteration flag ← True
            Next half solution flag ← False

iterate through the Cube’s pair of opposite sides
for each pair of the current cube:

 #this stops the dfs at the third thread (no solution)
   if we are on the first cube, half solution, and third color pair:
        Half solution flag ← False
        Backward iteration flag ← False
        Exit the current iterator

if a half solution hasn’t been found, the iterator is at the end of iteration, and the current pair is taken by the current thread, then
      if we are not at the third pair of the current cube:
           visited[i][j] = 0
           multiplicity[cubes[i][j][0]-1] -= 1
           multiplicity[cubes[i][j][1]-1] -= 1
           j += 1
```

```
                else:
                    visited[i][j] = 0
                    multiplicity[cubes[i][j][0]-1] -= 1
                    multiplicity[cubes[i][j][1]-1] -= 1
                    next_half_solution = True

            #half solution found
            if next_half_solution:
                break

            #(If this pair hasnt been visited, and the count of the colors of the pair are under 2,
            #then pick this pair and add one to the color count)
            if visited[i][j] == 0 and multiplicity[cubes[i][j][0] - 1] + 1 < 3 and
multiplicity[cubes[i][j][1] - 1] + 1 < 3:
                visited[i][j] = solution_num
                multiplicity[cubes[i][j][0] - 1] += 1
                multiplicity[cubes[i][j][1] - 1] += 1
                moving_backwards = False
                i += 1
                break

        #(moves back the thread to the previous cube to select the available pair. If on the last pair,
move back until possible)
        if moving_backwards:
            i = i - 1
            done_backtracking = False
            while not done_backtracking and i >= 0:
                for k in range(3):
                    if visited[i][k] == solution_num:
                        visited[i][k] = 0
                        multiplicity[cubes[i][k][0] - 1] -= 1
                        multiplicity[cubes[i][k][1] - 1] -= 1
                        k = k + 1
                        while k < 3:
                            if visited[i][k] == 0 and (multiplicity[cubes[i][k][0] - 1] + 1) < 3 and
(multiplicity[cubes[i][k][1] - 1] + 1) < 3:
                                visited[i][k] = solution_num
                                multiplicity[cubes[i][k][0] - 1] += 1
                                multiplicity[cubes[i][k][1] - 1] += 1
                                done_backtracking = True
                                i += 1
                                break
                            k = k + 1
                        break
                if not done_backtracking:
                    i = i - 1
        #found the end of the stack, Therefore solution.
        if i == len(cubes):
            half_sol_flag = True

        #found the end of the stack on the first solution
        if i == len(cubes) and solution_num != 2:
            half_sol_counter += 1
            #print("First Half Solution")
            #for t in range(len(visited)):
                #print(visited[t])

            #Find second half solution
            half_sol_flag = solve_stack(cubes, visited, 2)


            if not half_sol_flag:
                i -= 1

    if i == -1 :
        half_sol_flag = False
        #print("Number of half solutions: ", half_sol_counter)
        #print("No Solution")

    #if half_sol_flag and solution_num == 1:
        #print("Number of half solutions: ", half_sol_counter)
        #print("Complete Solution")
        #for t in range(len(visited)):
        #    print(visited[t])

    #if half_sol_flag == False and solution_num == 1 and i == -1:
        #print("No Solution")
    #if half_sol_flag == True and solution_num == 2:
        #print("Solution Found")
    return half_sol_flag
```

Code:

```
import math as math
from itertools import combinations

def cubes_generator(colors):
    cubes = []
    cube = []
    pair = []
    for i in range(180):
        pair.append(colors[i])
        if (i + 1) % 2 == 0:
          cube.append(pair)
          pair = []
        if (i + 1) % 6 == 0:
          cubes.append(cube)
          cube = []
    return cubes

def solve_stack(cubes, visited, solution_num):
    i = 0
    multiplicity = [0 for i in range(30)]
    half_sol_flag = False
    half_sol_counter = 0
    while -1 < i < len(cubes):
        moving_forward = True
        next_half_solution = False
        for j in range(3):

          # Stops the dfs at the third thread
          if i == 0 and solution_num == 1 and j == 2:
            half_sol_flag = False
            moving_forward = False
            break

          if half_sol_flag == False and i == (len(cubes) - 1) and visited[i][j] == solution_num:
            if j < 2:
              visited[i][j] = 0
              multiplicity[cubes[i][j][0]-1] -= 1
              multiplicity[cubes[i][j][1]-1] -= 1
              j += 1
            else:
              visited[i][j] = 0
              multiplicity[cubes[i][j][0]-1] -= 1
              multiplicity[cubes[i][j][1]-1] -= 1
              next_half_solution = True

          if next_half_solution:
            break

          if visited[i][j] == 0 and multiplicity[cubes[i][j][0] - 1] + 1 < 3 and
multiplicity[cubes[i][j][1] - 1] + 1 < 3:
            visited[i][j] = solution_num
            multiplicity[cubes[i][j][0] - 1] += 1
            multiplicity[cubes[i][j][1] - 1] += 1
            moving_forward = False
            i += 1
```

```
                break

        if moving_forward:
            i = i - 1
            done_backtracking = False
            while not done_backtracking and i >= 0:
                for k in range(3):
                    if visited[i][k] == solution_num:
                        visited[i][k] = 0
                        multiplicity[cubes[i][k][0] - 1] -= 1
                        multiplicity[cubes[i][k][1] - 1] -= 1
                        k = k + 1
                        while k < 3:
                            if visited[i][k] == 0 and (multiplicity[cubes[i][k][0] - 1] + 1) < 3 and
(multiplicity[cubes[i][k][1] - 1] + 1) < 3:
                                visited[i][k] = solution_num
                                multiplicity[cubes[i][k][0] - 1] += 1
                                multiplicity[cubes[i][k][1] - 1] += 1
                                done_backtracking = True
                                i += 1
                                break
                            k = k + 1
                        break
                if not done_backtracking:
                    i = i - 1
        if i == len(cubes):
            half_sol_flag = True

        if i == len(cubes) and solution_num != 2:
            half_sol_counter += 1
            print("First Half Solution")
            for t in range(len(visited)):
                print(visited[t])
            half_sol_flag = solve_stack(cubes, visited, 2)

            if not half_sol_flag:
                i -= 1

    if i == -1 :
        half_sol_flag = False
        print("Number of half solutions: ", half_sol_counter)
        print("No Solution")

    if half_sol_flag and solution_num == 1:
        print("NUmber of half solutions: ", half_sol_counter)
        print("Complete Solution")
    for t in range(len(visited)):
      print(visited[t])

  if half_sol_flag == False and solution_num == 1 and i == -1:
    print("No Solution")
  if half_sol_flag == True and solution_num == 2:
    print("Solution Found")

  return half_sol_flag

def find_min_obs(cubes):
  min_obs = len(cubes) + 1
  i = len(cubes)
  while i > 0:
    solution = False
    subsets = combinations(cubes, i)
    for subset in subsets:
      visited = [[0, 0, 0] for i in range(len(subset))]
      solution = solve_stack(subset, visited, 1)
      if not solution:
        print("A 'No Solution' was found in a stack of ", i ," cubes!")
        print("Minimum obstacle is now: ", min_obs -1)
        min_obs -= 1
        for element in subset:
          print(element)
        break
      else:
        print("A Solution Was Found")
    if solution:
      print("Every subset of ", i , " cubes had a solution")
      print("Minimum Obstacle: ", min_obs)
      if min_obs > len(cubes):
        print("No Minimal Obstacle")
      break
    i -= 1


# Puzzle 1
color1 = [1 + math.floor(n * math.pi) % 30 for n in range(1, 181)]
# Puzzle 2
color2 = [(1 + math.floor( i * math.e % 30)) for i in range(1,181)]
# Puzzle 3
color3 = [(1 + math.floor( i * math.sqrt(3) ) % 30) for i in range(1,181)]
# Puzzle 4
color4 = [(1 + math.floor( i * math.sqrt(5)) % 30) for i in range(1,181)]
```

```
cubes = cubes_generator(color1)

find_min_obs(cubes)
```

Output:

Solution for **Puzzle 1**: 1 + ((floor nπ) mod 30), 1 ≤ n ≤ 180

```
A 'No Solution' was found in a stack of  30  cubes!
Minimum obstacle is now:  30
[[4, 7], [10, 13], [16, 19]]
[[22, 26], [29, 2], [5, 8]]
[[11, 14], [18, 21], [24, 27]]
[[30, 3], [6, 10], [13, 16]]
[[19, 22], [25, 28], [2, 5]]
[[8, 11], [14, 17], [20, 24]]
[[27, 30], [3, 6], [9, 12]]
[[16, 19], [22, 25], [28, 1]]
[[4, 8], [11, 14], [17, 20]]
[[23, 26], [30, 3], [6, 9]]
[[12, 15], [18, 22], [25, 28]]
[[1, 4], [7, 10], [14, 17]]
[[20, 23], [26, 29], [2, 6]]
[[9, 12], [15, 18], [21, 24]]
[[28, 1], [4, 7], [10, 13]]
[[16, 20], [23, 26], [29, 2]]
[[5, 8], [12, 15], [18, 21]]
[[24, 27], [30, 4], [7, 10]]
[[13, 16], [19, 22], [25, 29]]
[[2, 5], [8, 11], [14, 17]]
[[21, 24], [27, 30], [3, 6]]
[[9, 13], [16, 19], [22, 25]]
[[28, 1], [5, 8], [11, 14]]
[[17, 20], [23, 27], [30, 3]]
[[6, 9], [12, 15], [19, 22]]
[[25, 28], [1, 4], [7, 11]]
[[14, 17], [20, 23], [26, 29]]
[[3, 6], [9, 12], [15, 18]]
[[21, 25], [28, 1], [4, 7]]
[[10, 13], [17, 20], [23, 26]]
Every subset of  29  cubes had a solution
Minimum Obstacle:  30
-----------------------------------------------------------------------------------------------------------------
------------------------
```

Solution for **Puzzle 2**: 1 + ((floor ne) mod 30), 1 ≤ n ≤ 180

```
A 'No Solution' was found in a stack of  30  cubes!
Minimum obstacle is now:  30
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
[[29, 2], [5, 8], [10, 13]]
[[16, 18], [21, 24], [27, 29]]
[[2, 5], [7, 10], [13, 16]]
[[18, 21], [24, 26], [29, 2]]
[[5, 7], [10, 13], [16, 18]]
[[21, 24], [26, 29], [2, 5]]
[[7, 10], [13, 15], [18, 21]]
[[24, 26], [29, 2], [4, 7]]
[[10, 13], [15, 18], [21, 23]]
[[26, 29], [2, 4], [7, 10]]
A 'No Solution' was found in a stack of  29  cubes!
Minimum obstacle is now:  29
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
```

```
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
[[29, 2], [5, 8], [10, 13]]
[[16, 18], [21, 24], [27, 29]]
[[2, 5], [7, 10], [13, 16]]
[[18, 21], [24, 26], [29, 2]]
[[5, 7], [10, 13], [16, 18]]
[[21, 24], [26, 29], [2, 5]]
[[7, 10], [13, 15], [18, 21]]
[[24, 26], [29, 2], [4, 7]]
[[10, 13], [15, 18], [21, 23]]
A 'No Solution' was found in a stack of  28  cubes!
Minimum obstacle is now:  28
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
[[29, 2], [5, 8], [10, 13]]
[[16, 18], [21, 24], [27, 29]]
[[2, 5], [7, 10], [13, 16]]
[[18, 21], [24, 26], [29, 2]]
[[5, 7], [10, 13], [16, 18]]
```

```
[[21, 24], [26, 29], [2, 5]]
[[7, 10], [13, 15], [18, 21]]
[[24, 26], [29, 2], [4, 7]]
A 'No Solution' was found in a stack of  27  cubes!
Minimum obstacle is now:  27
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
[[29, 2], [5, 8], [10, 13]]
[[16, 18], [21, 24], [27, 29]]
[[2, 5], [7, 10], [13, 16]]
[[18, 21], [24, 26], [29, 2]]
[[5, 7], [10, 13], [16, 18]]
[[21, 24], [26, 29], [2, 5]]
[[7, 10], [13, 15], [18, 21]]
A 'No Solution' was found in a stack of  26  cubes!
Minimum obstacle is now:  26
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
```

```
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
[[29, 2], [5, 8], [10, 13]]
[[16, 18], [21, 24], [27, 29]]
[[2, 5], [7, 10], [13, 16]]
[[18, 21], [24, 26], [29, 2]]
[[5, 7], [10, 13], [16, 18]]
[[21, 24], [26, 29], [2, 5]]
A 'No Solution' was found in a stack of  25  cubes!
Minimum obstacle is now:  25
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
[[29, 2], [5, 8], [10, 13]]
[[16, 18], [21, 24], [27, 29]]
[[2, 5], [7, 10], [13, 16]]
[[18, 21], [24, 26], [29, 2]]
[[5, 7], [10, 13], [16, 18]]
A 'No Solution' was found in a stack of  24  cubes!
Minimum obstacle is now:  24
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
```

```
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
[[29, 2], [5, 8], [10, 13]]
[[16, 18], [21, 24], [27, 29]]
[[2, 5], [7, 10], [13, 16]]
[[18, 21], [24, 26], [29, 2]]
A 'No Solution' was found in a stack of  23  cubes!
Minimum obstacle is now:  23
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
[[29, 2], [5, 8], [10, 13]]
[[16, 18], [21, 24], [27, 29]]
[[2, 5], [7, 10], [13, 16]]
```

```
A 'No Solution' was found in a stack of  22  cubes!
Minimum obstacle is now:  22
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
[[29, 2], [5, 8], [10, 13]]
[[16, 18], [21, 24], [27, 29]]
A 'No Solution' was found in a stack of  21  cubes!
Minimum obstacle is now:  21
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
```

```
[[29, 2], [5, 8], [10, 13]]
A 'No Solution' was found in a stack of  20  cubes!
Minimum obstacle is now:  20
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
A 'No Solution' was found in a stack of  19  cubes!
Minimum obstacle is now:  19
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
[[27, 30], [2, 5], [8, 10]]
A 'No Solution' was found in a stack of  18  cubes!
Minimum obstacle is now:  18
```

```
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[10, 13], [16, 19], [21, 24]]
A 'No Solution' was found in a stack of  17  cubes!
Minimum obstacle is now:  17
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
A 'No Solution' was found in a stack of  16  cubes!
Minimum obstacle is now:  16
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[6, 9], [11, 14], [17, 19]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
```

```
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[22, 24], [27, 30], [2, 5]]
[[8, 11], [13, 16], [19, 21]]
A 'No Solution' was found in a stack of  15  cubes!
Minimum obstacle is now:  15
[[3, 6], [9, 11], [14, 17]]
[[20, 22], [25, 28], [30, 3]]
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[11, 14], [17, 19], [22, 25]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[22, 24], [27, 30], [2, 5]]
[[24, 27], [30, 2], [5, 8]]
[[27, 30], [2, 5], [8, 10]]
A 'No Solution' was found in a stack of  14  cubes!
Minimum obstacle is now:  14
[[22, 25], [28, 30], [3, 6]]
[[8, 11], [14, 17], [19, 22]]
[[25, 27], [30, 3], [6, 8]]
[[27, 30], [3, 6], [8, 11]]
[[14, 16], [19, 22], [25, 27]]
[[30, 3], [5, 8], [11, 14]]
[[16, 19], [22, 24], [27, 30]]
[[3, 5], [8, 11], [13, 16]]
[[19, 22], [24, 27], [30, 3]]
[[5, 8], [11, 13], [16, 19]]
[[8, 11], [13, 16], [19, 21]]
[[24, 27], [30, 2], [5, 8]]
[[27, 30], [2, 5], [8, 10]]
[[13, 16], [19, 21], [24, 27]]
Every subset of  13  cubes had a solution
Minimum Obstacle:  14
--------------------------------------------------------------------------------------------------------------------------
--------------------------
```

Solution for **Puzzle 3**: 1 + ((floor n√3) mod 30), 1 ≤ n ≤ 180

```
A 'No Solution' was found in a stack of  30  cubes!
Minimum obstacle is now:  30
[[2, 4], [6, 7], [9, 11]]
[[13, 14], [16, 18], [20, 21]]
[[23, 25], [26, 28], [30, 2]]
[[3, 5], [7, 9], [10, 12]]
[[14, 16], [17, 19], [21, 22]]
[[24, 26], [28, 29], [1, 3]]
[[5, 6], [8, 10], [12, 13]]
[[15, 17], [18, 20], [22, 24]]
[[25, 27], [29, 1], [2, 4]]
[[6, 7], [9, 11], [13, 14]]
[[16, 18], [20, 21], [23, 25]]
[[27, 28], [30, 2], [3, 5]]
[[7, 9], [10, 12], [14, 16]]
[[17, 19], [21, 23], [24, 26]]
[[28, 29], [1, 3], [5, 6]]
[[8, 10], [12, 13], [15, 17]]
[[19, 20], [22, 24], [25, 27]]
[[29, 1], [2, 4], [6, 8]]
[[9, 11], [13, 14], [16, 18]]
[[20, 21], [23, 25], [27, 28]]
[[30, 2], [4, 5], [7, 9]]
[[10, 12], [14, 16], [17, 19]]
[[21, 23], [24, 26], [28, 30]]
[[1, 3], [5, 6], [8, 10]]
[[12, 13], [15, 17], [19, 20]]
[[22, 24], [26, 27], [29, 1]]
[[2, 4], [6, 8], [9, 11]]
[[13, 15], [16, 18], [20, 21]]
[[23, 25], [27, 28], [30, 2]]
[[4, 5], [7, 9], [11, 12]]
A 'No Solution' was found in a stack of  29  cubes!
Minimum obstacle is now:  29
[[2, 4], [6, 7], [9, 11]]
[[13, 14], [16, 18], [20, 21]]
[[23, 25], [26, 28], [30, 2]]
[[3, 5], [7, 9], [10, 12]]
[[14, 16], [17, 19], [21, 22]]
[[24, 26], [28, 29], [1, 3]]
[[5, 6], [8, 10], [12, 13]]
[[15, 17], [18, 20], [22, 24]]
[[25, 27], [29, 1], [2, 4]]
[[6, 7], [9, 11], [13, 14]]
```

```
[[16, 18], [20, 21], [23, 25]]
[[27, 28], [30, 2], [3, 5]]
[[7, 9], [10, 12], [14, 16]]
[[17, 19], [21, 23], [24, 26]]
[[28, 29], [1, 3], [5, 6]]
[[8, 10], [12, 13], [15, 17]]
[[19, 20], [22, 24], [25, 27]]
[[29, 1], [2, 4], [6, 8]]
[[9, 11], [13, 14], [16, 18]]
[[20, 21], [23, 25], [27, 28]]
[[30, 2], [4, 5], [7, 9]]
[[10, 12], [14, 16], [17, 19]]
[[21, 23], [24, 26], [28, 30]]
[[1, 3], [5, 6], [8, 10]]
[[12, 13], [15, 17], [19, 20]]
[[22, 24], [26, 27], [29, 1]]
[[2, 4], [6, 8], [9, 11]]
[[13, 15], [16, 18], [20, 21]]
[[23, 25], [27, 28], [30, 2]]
A 'No Solution' was found in a stack of  28  cubes!
Minimum obstacle is now:  28
[[2, 4], [6, 7], [9, 11]]
[[13, 14], [16, 18], [20, 21]]
[[23, 25], [26, 28], [30, 2]]
[[3, 5], [7, 9], [10, 12]]
[[14, 16], [17, 19], [21, 22]]
[[24, 26], [28, 29], [1, 3]]
[[5, 6], [8, 10], [12, 13]]
[[15, 17], [18, 20], [22, 24]]
[[25, 27], [29, 1], [2, 4]]
[[6, 7], [9, 11], [13, 14]]
[[16, 18], [20, 21], [23, 25]]
[[27, 28], [30, 2], [3, 5]]
[[7, 9], [10, 12], [14, 16]]
[[17, 19], [21, 23], [24, 26]]
[[28, 29], [1, 3], [5, 6]]
[[8, 10], [12, 13], [15, 17]]
[[19, 20], [22, 24], [25, 27]]
[[29, 1], [2, 4], [6, 8]]
[[9, 11], [13, 14], [16, 18]]
[[20, 21], [23, 25], [27, 28]]
[[30, 2], [4, 5], [7, 9]]
[[10, 12], [14, 16], [17, 19]]
[[21, 23], [24, 26], [28, 30]]
[[1, 3], [5, 6], [8, 10]]
[[12, 13], [15, 17], [19, 20]]
```

```
[[22, 24], [26, 27], [29, 1]]
[[2, 4], [6, 8], [9, 11]]
[[13, 15], [16, 18], [20, 21]]
A 'No Solution' was found in a stack of  27  cubes!
Minimum obstacle is now:  27
[[2, 4], [6, 7], [9, 11]]
[[13, 14], [16, 18], [20, 21]]
[[23, 25], [26, 28], [30, 2]]
[[3, 5], [7, 9], [10, 12]]
[[14, 16], [17, 19], [21, 22]]
[[24, 26], [28, 29], [1, 3]]
[[5, 6], [8, 10], [12, 13]]
[[15, 17], [18, 20], [22, 24]]
[[25, 27], [29, 1], [2, 4]]
[[6, 7], [9, 11], [13, 14]]
[[16, 18], [20, 21], [23, 25]]
[[27, 28], [30, 2], [3, 5]]
[[7, 9], [10, 12], [14, 16]]
[[17, 19], [21, 23], [24, 26]]
[[28, 29], [1, 3], [5, 6]]
[[8, 10], [12, 13], [15, 17]]
[[19, 20], [22, 24], [25, 27]]
[[29, 1], [2, 4], [6, 8]]
[[9, 11], [13, 14], [16, 18]]
[[20, 21], [23, 25], [27, 28]]
[[30, 2], [4, 5], [7, 9]]
[[10, 12], [14, 16], [17, 19]]
[[21, 23], [24, 26], [28, 30]]
[[1, 3], [5, 6], [8, 10]]
[[2, 4], [6, 8], [9, 11]]
[[13, 15], [16, 18], [20, 21]]
[[23, 25], [27, 28], [30, 2]]
A 'No Solution' was found in a stack of  26  cubes!
Minimum obstacle is now:  26
[[2, 4], [6, 7], [9, 11]]
[[13, 14], [16, 18], [20, 21]]
[[23, 25], [26, 28], [30, 2]]
[[3, 5], [7, 9], [10, 12]]
[[14, 16], [17, 19], [21, 22]]
[[24, 26], [28, 29], [1, 3]]
[[5, 6], [8, 10], [12, 13]]
[[15, 17], [18, 20], [22, 24]]
[[25, 27], [29, 1], [2, 4]]
[[6, 7], [9, 11], [13, 14]]
[[16, 18], [20, 21], [23, 25]]
[[27, 28], [30, 2], [3, 5]]
```

```
[[7, 9], [10, 12], [14, 16]]
[[28, 29], [1, 3], [5, 6]]
[[8, 10], [12, 13], [15, 17]]
[[19, 20], [22, 24], [25, 27]]
[[29, 1], [2, 4], [6, 8]]
[[9, 11], [13, 14], [16, 18]]
[[20, 21], [23, 25], [27, 28]]
[[30, 2], [4, 5], [7, 9]]
[[10, 12], [14, 16], [17, 19]]
[[1, 3], [5, 6], [8, 10]]
[[12, 13], [15, 17], [19, 20]]
[[2, 4], [6, 8], [9, 11]]
[[13, 15], [16, 18], [20, 21]]
[[23, 25], [27, 28], [30, 2]]
A 'No Solution' was found in a stack of  25  cubes!
Minimum obstacle is now:  25
[[2, 4], [6, 7], [9, 11]]
[[13, 14], [16, 18], [20, 21]]
[[23, 25], [26, 28], [30, 2]]
[[3, 5], [7, 9], [10, 12]]
[[14, 16], [17, 19], [21, 22]]
[[24, 26], [28, 29], [1, 3]]
[[5, 6], [8, 10], [12, 13]]
[[25, 27], [29, 1], [2, 4]]
[[6, 7], [9, 11], [13, 14]]
[[16, 18], [20, 21], [23, 25]]
[[27, 28], [30, 2], [3, 5]]
[[7, 9], [10, 12], [14, 16]]
[[28, 29], [1, 3], [5, 6]]
[[8, 10], [12, 13], [15, 17]]
[[29, 1], [2, 4], [6, 8]]
[[9, 11], [13, 14], [16, 18]]
[[20, 21], [23, 25], [27, 28]]
[[30, 2], [4, 5], [7, 9]]
[[10, 12], [14, 16], [17, 19]]
[[21, 23], [24, 26], [28, 30]]
[[1, 3], [5, 6], [8, 10]]
[[2, 4], [6, 8], [9, 11]]
[[13, 15], [16, 18], [20, 21]]
[[23, 25], [27, 28], [30, 2]]
[[4, 5], [7, 9], [11, 12]]
Every subset of  24  cubes had a solution
Minimum Obstacle:  25
```

```
--------------------------------------------------------------------------------------------------------------------------
--------------------------
```

Solution for **Puzzle 4**: 1 + ((floor n√5) mod 30), 1 ≤ n ≤ 180

```
A 'No Solution' was found in a stack of  30  cubes!
Minimum obstacle is now:  30
[[3, 5], [7, 9], [12, 14]]
[[16, 18], [21, 23], [25, 27]]
[[30, 2], [4, 6], [9, 11]]
[[13, 15], [17, 20], [22, 24]]
[[26, 29], [1, 3], [5, 8]]
[[10, 12], [14, 17], [19, 21]]
[[23, 25], [28, 30], [2, 4]]
[[7, 9], [11, 13], [16, 18]]
[[20, 22], [25, 27], [29, 1]]
[[3, 6], [8, 10], [12, 15]]
[[17, 19], [21, 24], [26, 28]]
[[30, 3], [5, 7], [9, 11]]
[[14, 16], [18, 20], [23, 25]]
[[27, 29], [2, 4], [6, 8]]
[[11, 13], [15, 17], [20, 22]]
[[24, 26], [28, 1], [3, 5]]
[[7, 10], [12, 14], [16, 19]]
[[21, 23], [25, 28], [30, 2]]
[[4, 6], [9, 11], [13, 15]]
[[18, 20], [22, 24], [27, 29]]
[[1, 3], [6, 8], [10, 12]]
[[14, 17], [19, 21], [23, 26]]
[[28, 30], [2, 5], [7, 9]]
[[11, 14], [16, 18], [20, 22]]
[[25, 27], [29, 1], [4, 6]]
[[8, 10], [13, 15], [17, 19]]
[[22, 24], [26, 28], [1, 3]]
[[5, 7], [9, 12], [14, 16]]
[[18, 21], [23, 25], [27, 30]]
[[2, 4], [6, 9], [11, 13]]
Every subset of  29  cubes had a solution
Minimum Obstacle:  30
----------------------------------------------------------------------------------------------------------------------
-------------------------
```

Extra Solution for **Puzzle 5**: 1 + ((floor n√7) mod 30), 1 ≤ n ≤ 180