

BEE BREEDING

BATCH-28

T.L.Gangothri 21B01A05H8 CSE

Md.Akhila 21B01A12B2 IT

Md.Sheema 21B01A12B3 IT

N.Neha 21B01A05B3 CSE

Shri Vishnu Engineering College for Women

FEBRUARY 10,2023

Introduction

Problem Statement

- ▶ The bees have difficulty breeding in captivity and Professor Heif wants to quantify the difference in maggot placement.
- ▶ The task is to write a program that computes the distance between any pair of cells in the honeycomb, defined as the number of cells in the shortest path between them.

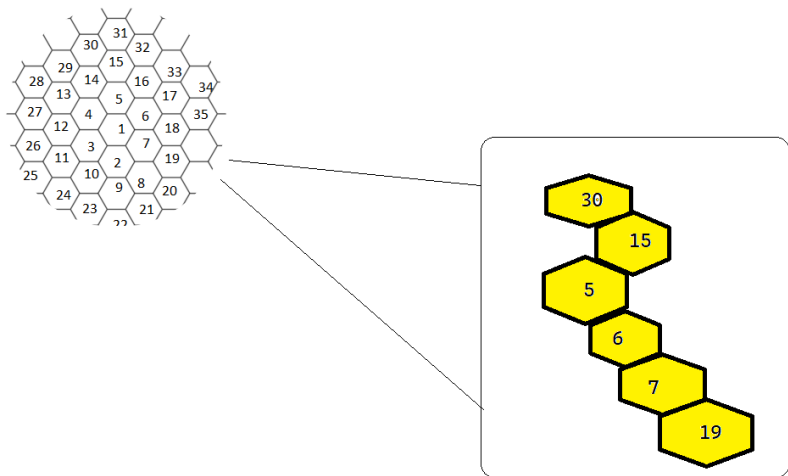


Figure 1: Honeycomb

Approach

- ▶ Create a Grid of Hexagonal cells.
- ▶ The grid is created by starting at the center cell $(0, 0, 0)$ and iterating outwards in concentric rings.
- ▶ Iterating through a list of coordinates and adding them to the current coordinates.

Learnings

- ▶ We learnt how to create a grid of hexagonal cells with a given radius.
- ▶ We came to know about Manhattan distance to calculate the shortest distance between two cells.
- ▶ We learnt how to use a dictionary to store the coordinates of each cell in the grid.

Challenges

- ▶ Understanding the hexagonal coordinate system.
- ▶ Input validation.
- ▶ Determining the correct radius to generate grid.
- ▶ Debugging and troubleshooting.

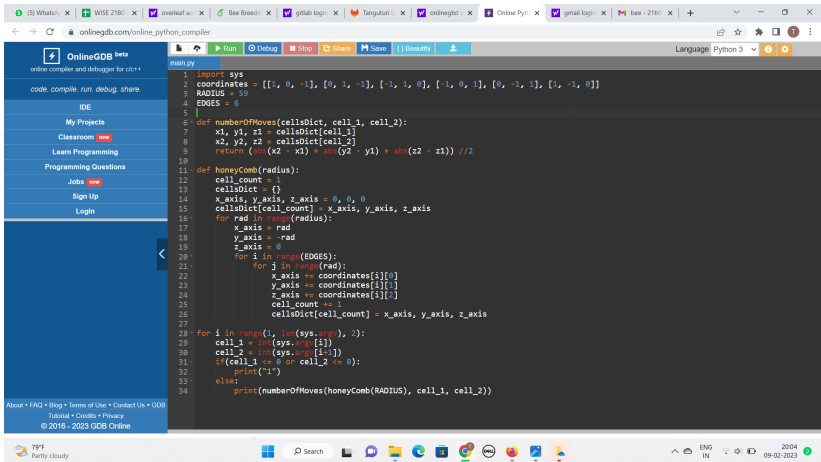
Solutions

- ▶ We have resolved this issue by utilizing a coordinate system and storing the information in a dictionary.
- ▶ We overcame the input validation problem using conditional statements.

Statistics

- ▶ The solution for this problem has 34 Lines of code
- ▶ The code has two user-defined functions.
They are:
 - 1.numberOfMoves()
 - 2.honeyComb
- ▶ The code has one built-in function.
That is:
 - 1.abs()

Demo



The screenshot displays the OnlineGDB web interface. The browser's address bar shows the URL `onlinegdb.com/online_python_compiler`. The interface includes a sidebar on the left with navigation links such as "IDE", "My Projects", "Classroom", "Learn Programming", "Programming Questions", "Jobs", "Sign Up", and "Login". The main area contains a code editor with the following Python code:

```
1 import sys
2 coordinates = [[1, 0, -1], [0, 1, -1], [-1, 1, 0], [-1, 0, 1], [0, -1, 1], [1, -1, 0]]
3 RADIUS = 50
4 EDGES = 6
5
6 def numberOfMoves(cellsDict, cell_1, cell_2):
7     x1, y1, z1 = cellsDict[cell_1]
8     x2, y2, z2 = cellsDict[cell_2]
9     return (abs(x2 - x1) + abs(y2 - y1) + abs(z2 - z1)) // 2
10
11 def honeyComb(radius):
12     cell_count = 1
13     cellsDict = {}
14     x_axis, y_axis, z_axis = 0, 0, 0
15     cellsDict[cell_count] = x_axis, y_axis, z_axis
16     for rad in range(radius):
17         x_axis = rad
18         y_axis = -rad
19         z_axis = 0
20         for i in range(EDGES):
21             for j in range(rad):
22                 x_axis += coordinates[i][0]
23                 y_axis += coordinates[i][1]
24                 z_axis += coordinates[i][2]
25                 cell_count += 1
26                 cellsDict[cell_count] = x_axis, y_axis, z_axis
27
28 for i in range(1, len(sys.argv), 2):
29     cell_1 = int(sys.argv[i])
30     cell_2 = int(sys.argv[i+1])
31     if (cell_1 <= 0 or cell_2 <= 0):
32         print("1")
33     else:
34         print(numberOfMoves(honeyComb(RADIUS), cell_1, cell_2))
```

The bottom of the interface shows a system tray with weather information (79°F, Partly cloudy), a search bar, and the date/time (2004-09-02-2023).

Figure 2: code

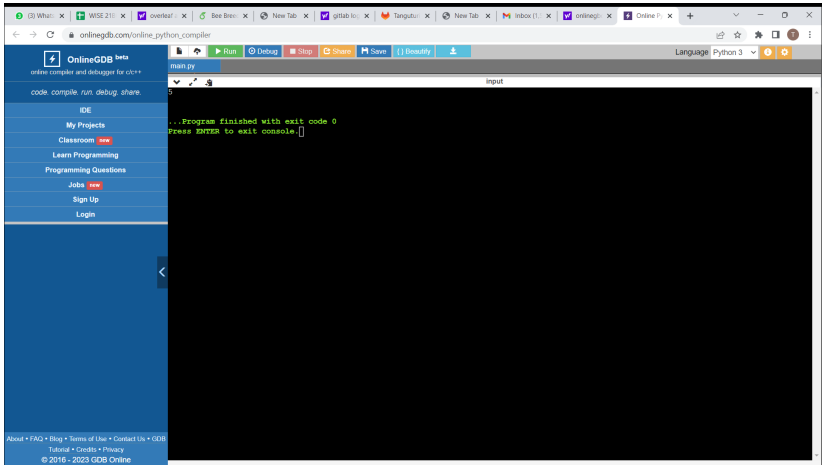


Figure 3: Output

THANK YOU