 # Payal Chavan

 # Summer 2024

 # CS 5800 Algorithms (Seattle)

 # Assignment 8

 # Date: 07/29/2024

Question 1: Write a solver for the 3-SAT problem

This code aims to solve a 3-SAT problem using an exhaustive search algorithm. The exhaustive search function goes through all possible variable assignments. For each assignment, it verifies if all clauses are satisfied. If a satisfying assignment is found, it returns True; otherwise, it returns False.
The input file, 3sat_input.txt, contains 3-SAT clauses. In this case, the input consists of two clauses: (x1 + ^x2 + x3) and (^x1 + ^x2 + x4). Since these clauses can be satisfied by a specific assignment of variables, the output will be:

```
/usr/bin/python3
/Users/payalchavan/Documents/Algorithms/Assignment8/3SAT_Solver.py

Satisfying assignment exists.
```

Question 2: Generating Random 3-SAT Clauses

This code is designed to generate random 3-SAT problems with a specified number of variables and clauses. The ‘generate_3sat’ function constructs a list of variables using the first N letters of the alphabet (e.g., ['a', 'b', 'c'] for N=3). For each clause, the function randomly selects three distinct variables from this list. Each variable has a 50% chance of being negated, indicated by a (~) prefix. The function returns a list of clauses, where each clause consists of three literals (either variables or their negations). Due to the random selection process, each execution of the code will yield different outputs. The output for this problem will be like:

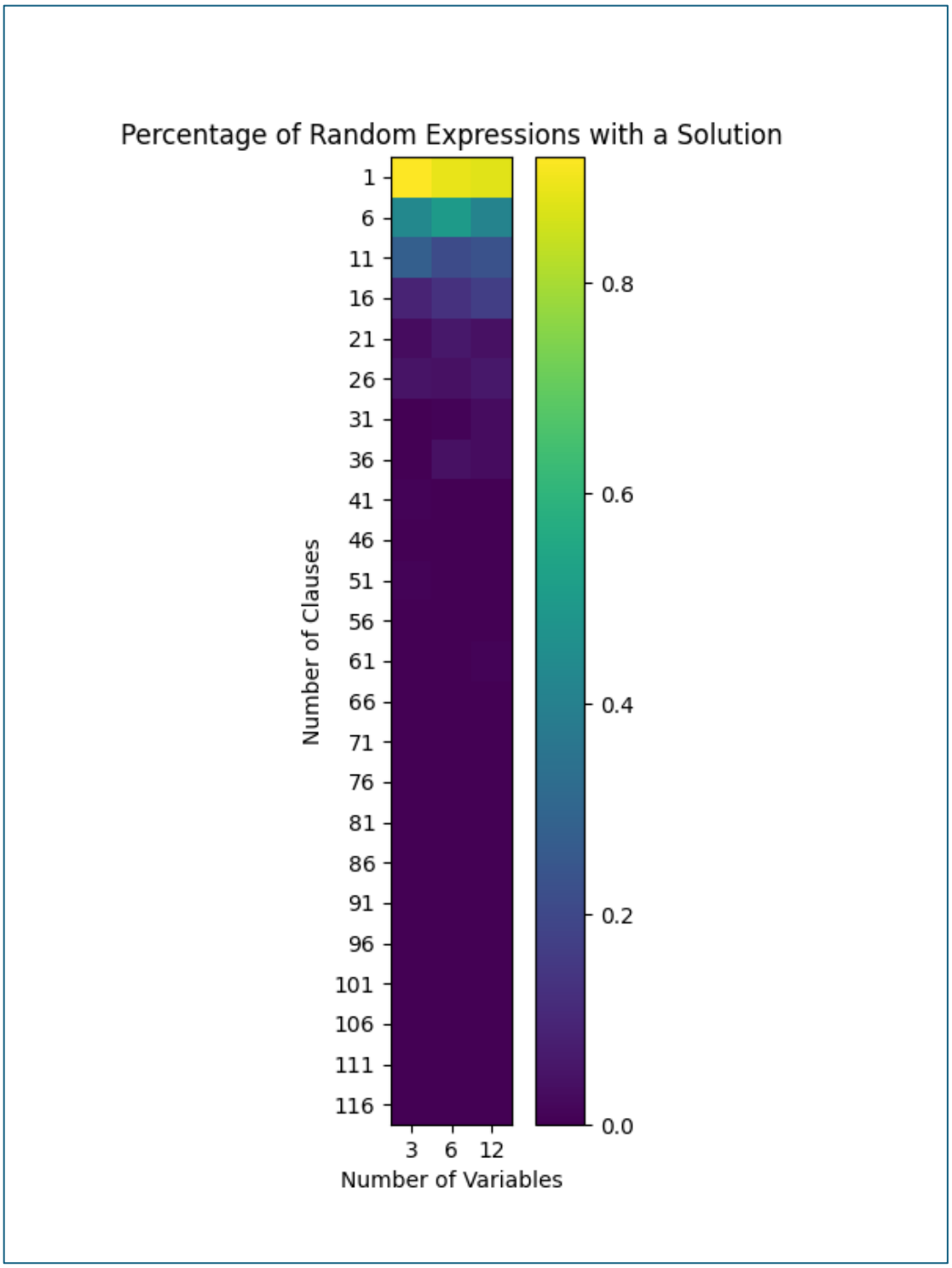
```
/usr/bin/python3 /Users/payalchavan/Documents/Algorithms/Assignment8/3SAT-
clause.py

[['~b', '~c', '~a'], ['b', 'a', '~c'], ['~c', 'a', 'b']]

[['~a', 'e', '~c'], ['~b', 'd', '~a'], ['b', '~c', 'a'], ['d', '~b', 'e'],
['~b', 'd', '~c']]
```

 # Question 3: Experiment

The code generates a heatmap to illustrate the percentage of random solutions that satisfy 3-SAT problems. The x-axis shows the number of variables (N), and the y-axis shows the number of clauses (C). The color intensity represents the percentage of satisfying solutions, with yellow indicating a higher percentage (close to 1) and dark purple indicating a lower percentage (close to 0). This visualization helps to understand how the complexity of 3-SAT problems varies with the number of variables and clauses. Generally, as the number of clauses increases, the percentage of solvable expressions decreases. For fewer variables, the percentage of solvable expressions is higher, but it decreases as the number of variables increases. This suggests that as the complexity (number of clauses and variables) of an expression increases, the likelihood of finding a solution decreases. There are distinct transition zones where the color shifts from yellow to dark purple. These zones indicate critical points where the solvability of the 3-SAT problems drops significantly. Identifying these zones can help understand the thresholds of complexity. For a fixed number of clauses, increasing the number of variables initially maintains a higher percentage of solvable expressions, but beyond a certain point, the percentage drops sharply. This suggests that there is a balance between the number of variables and clauses that maximizes solvability. The heatmap shows that for a small number of variables, even a moderate increase in the number of clauses can drastically reduce the percentage of solvable expressions. This highlights the sensitivity of 3-SAT problems to clause density. Hence, this illustration makes sense, as it helps us to understand the relation between variables and clauses with the percentage of random solutions.


Percentage of Random Expressions with a Solution
Number of Clauses
1
6
11
16
21
26
31
36
41
46
51
56
61
66
71
76
81
86
91
96
101
106
111
116
3
6
12
Number of Variables
0.8
0.6
0.4
0.2
0.0

 # Reflection:

In this coding exercise, I developed a 3SAT solver to determine whether a given assignment satisfies the problem. Additionally, I learned to generate random 3-SAT expressions based on a specified number of variables (N) and clauses (C). Finally, I conducted an experiment to calculate the percentage of random solutions that satisfy the 3-SAT problem.

 # Acknowledgements:

I would like to express my sincere gratitude to the following individuals and resources for their invaluable contributions to this assignment:

1. Prof. Bruce Maxwell: Thank you, Professor Bruce Maxwell for your guidance in this assignment. Your expertise in algorithms greatly influenced my work.
2. Classmates and TA's: I appreciate the discussions of my classmates, and TA's who clarified my doubts.
3. DPV Algorithms Textbook: This textbook was a useful resource for me to understand the basics of algorithms.
4. https://www.baeldung.com/cs/cook-levin-theorem-3sat: This website was a useful resource to understand the 3SAT problem and its satisfiability.