

Project Readme Template

Version 1 9/11/24

A single copy of this template should be filled out and submitted with each project submission, regardless of the number of students on the team. It should have the name readme_ "teamname" Also change the title of this template to "Project x Readme Team xxx"

1	Team Name: chealy5										
2	Team members names and netids: Catherine Healy (chealy5)										
3	Overall project attempted, with sub-projects: Implementing a polynomial time 2-SAT solver using DPLL algorithm										
4	Overall success of the project: Successfully coded program.										
5	Approximately total time (in hours) to complete: 10+ hours										
6	Link to github repository: https://github.com/CatherineH3/Theory_Project_1_Final										
7	<p>List of included files (if you have many files of a certain type, such as test files of different sizes, list just the folder): (Add more rows as necessary). Add more rows as necessary.</p> <table border="1"><thead><tr><th>File/folder Name</th><th>File Contents and Use</th></tr></thead><tbody><tr><td colspan="2">Code Files</td></tr><tr><td>code_2sat_chealy5.py</td><td>code_2sat_chealy5.py is the code for my 2SAT solver. It can take an input file and generate a csv output with the times and satisfiability results. It also generates a graph.</td></tr><tr><td colspan="2">Test Files</td></tr><tr><td>check_1_chealy5.csv check_2SAT.cnf_chealy5.csv check_dumbsat (folder with check_dumbsat_chealy5.py and check_dumbsat_output_chealy5.csv and plots_DUMBSAT_chealy5.png)</td><td>check_1_chealy5.csv is a very short test with 3 simple cases that I created. I explain them in more detail in the testing section of the Readme. The only purpose of this is to verify correctness for a few cases by hand. It is not useful for timing because there are not enough test cases. check_2SAT.cnf_chealy5.csv was the provided test file. It contains 100 cases with clauses of varying lengths. The one</td></tr></tbody></table>	File/folder Name	File Contents and Use	Code Files		code_2sat_chealy5.py	code_2sat_chealy5.py is the code for my 2SAT solver. It can take an input file and generate a csv output with the times and satisfiability results. It also generates a graph.	Test Files		check_1_chealy5.csv check_2SAT.cnf_chealy5.csv check_dumbsat (folder with check_dumbsat_chealy5.py and check_dumbsat_output_chealy5.csv and plots_DUMBSAT_chealy5.png)	check_1_chealy5.csv is a very short test with 3 simple cases that I created. I explain them in more detail in the testing section of the Readme. The only purpose of this is to verify correctness for a few cases by hand. It is not useful for timing because there are not enough test cases. check_2SAT.cnf_chealy5.csv was the provided test file. It contains 100 cases with clauses of varying lengths. The one
File/folder Name	File Contents and Use										
Code Files											
code_2sat_chealy5.py	code_2sat_chealy5.py is the code for my 2SAT solver. It can take an input file and generate a csv output with the times and satisfiability results. It also generates a graph.										
Test Files											
check_1_chealy5.csv check_2SAT.cnf_chealy5.csv check_dumbsat (folder with check_dumbsat_chealy5.py and check_dumbsat_output_chealy5.csv and plots_DUMBSAT_chealy5.png)	check_1_chealy5.csv is a very short test with 3 simple cases that I created. I explain them in more detail in the testing section of the Readme. The only purpose of this is to verify correctness for a few cases by hand. It is not useful for timing because there are not enough test cases. check_2SAT.cnf_chealy5.csv was the provided test file. It contains 100 cases with clauses of varying lengths. The one										

	<p>file servers two purposes- it both is useful for verifying correctness and is useful also for timing. The main graph for the timing is actually generated in my main code, code_2sat_chealy5.py.</p> <p>check_dumbsat is a folder containing test files that I used to verify my solution. I generated a solution using dumbSAT using the same input file as 2SAT and compared it to the solution using 2SAT to verify correctness and also for a time comparison.</p> <p>check_dumbsat_chealy5.py is the code written to test the same input for DumbSAT</p> <p>check_dumbsat_output_chealy5.csv is the output to the check_dumbsat_chealy5.py code using input check_2SAT.cnf_chealy5.csv. The result is plots_DUMBSAT_chealy5.png</p>
Data File	
data_cnf_chealy5.csv	<p>data_cnf_chealy5.csv is actually identical to check_2SAT.cnf_chealy5.csv because I used it for both a timing run and for verifying correctness. I included it again under another name to follow the submission guidelines.</p>
Output Files	
output_metadata_chealy5.docx output_chealy5.csv output_test1_chealy5.csv	<p><i>Note: I could not add all of the necessary information into the csv file, so I added an additional metadata file for the output: output_metadata_chealy5.docx.</i></p> <p>output_chealy5.csv is the output after running the main 2Sat solver, code_2sat_chealy5.py, with data input: data_cnf_chealy5.csv. The data input is the same as the one provided as a test file. It is a csv with the runtime, number of literals, and whether it was satisfiable. This output was used for two purposes: (1) to verify the correctness of the code and (2) To record the timing of the code and show results.</p>

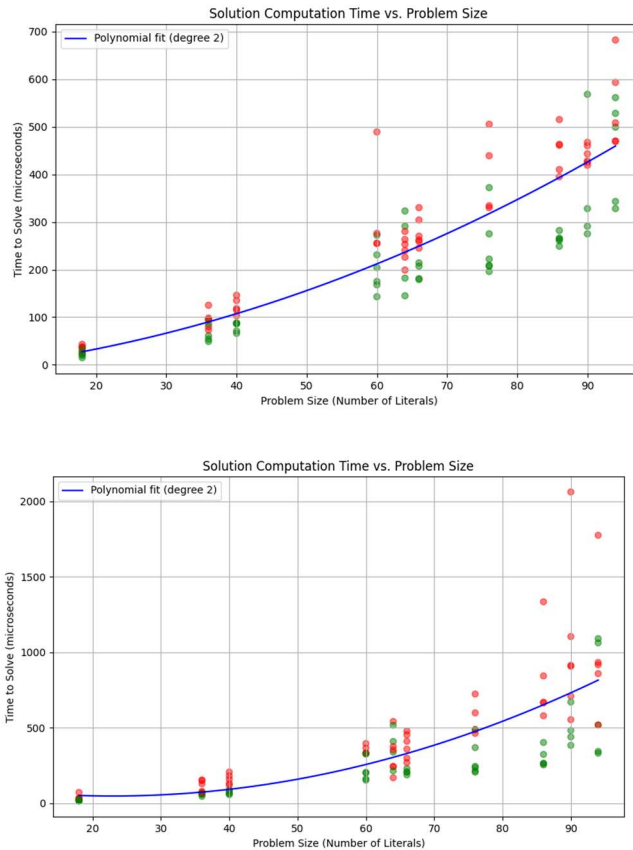
		A screenshot of my screen after running the code in Colabs is also included in the “additional information” section of this document to prove that it actually ran. output_test1_chealy5.csv is the output after usg input check_1_chealy5.csv .
	Plots (as needed)	
	plots_run1_chealy5.png plots_run2_chealy5.png plots_run3_chealy5.png	Plots_run 1-3 are all plots as a result of running <i>code_2sat_chealy5.py</i> with <i>data_cnf_chealy5.csv</i> input file. Each plot graphs the time vs. the number of literals. I included 3 cases because the results vary.
	Other files	
	output_console_screenshot_chealy5.png	output_console_screenshot_chealy5.png is a screenshot to prove I actually ran the code
8	<p>Programming languages used, and associated libraries:</p> <p>For this project, I used the python programming language. I used the following libraries: time, csv, dataclasses, matplotlib.pyplot, and numpy. I used the time library to calculate how long it took to solve each wff SAT problem. I used the dataclasses library to create my own dataclass for each wff to store its information. I used matplotlib.pyplot to plot my results. I used numpy to calculate the trend line.</p>	
9	<p>Key data structures (for each sub-project):</p> <p>The key data structure that I used was a dataclass to hold the information of each wff. It is as described below.</p> <pre>class WFF: problem_number: int # Problem Number n_variables: int # Number of variables satisfiability: str # S for satisfieable, U for unsatisfiable, ? for unknown all_clauses: list[list[int]] # store the clauses as a list of ints n_clauses: int = 0 # Number of clauses (initialized ot 0) time_to_solve: int = 0 # Keep track of how long it took to solve n_literals: int = 0 # number of literals is number of clauses * 2</pre> <p>Having this dataclass was very helpful because it provided a clear way to store all the information about each wff in an organized manner (ex: mywffs was a list I used), and it made it very easy to generate the output files.</p> <p>I also used many lists to keep track of all my lists. Using lists was helpful to stay organized and print all of my information. Additionally, I used a dictionary in the remove_pure_literals function in my DPLL algorithm to quickly count the number of each literal and try to identify whether or not there was a pure literal (a pure literal exists without its negation, so I had to check them all).</p> <p>Lastly, although I did not directly interact with the stack and use stack operations, I</p>	

	made use of the stack through my DPLL algorithm. Because the algorithm was recursive, it internally used the stack to keep track of the changing variable assignments when guessing each possible variable assignment.
10	<p>General operation of code (for each subproject)</p> <p>My project was the polynomial time 2SAT solver. The code takes in an input csv file in cnf form and generates an output csv file that contains the number of literals, execution time, and runtime for each wff. It does not accept files in other formats. In the main, I focused on parsing the csv file and putting all of the information into a WFF dataclass. Once it is all parsed and all the WFF dataclasses are stored in a list, the satSolve function is called. This function loops through each wff, calls the dpll algorithm to solve it, for each one. After seeing the result of whether it is satisfiable or unsatisfiable, it stores it in the WFF dataclass. Additionally, the function measures the amount of time it takes for the DPLL algorithm to run using the time library and saves this information. The DPLL function calls two other functions- one for unit propagation and another for elimination of pure literals. First, it searches for any unit clauses and calls the unit propagation function. This function simplifies the clause based on the literal assignment. Then, the DPLL function calls a function to remove pure literals. This searches for any literals that occur without their negation and removes them. Then, it checks if there are no more literals and if so, return true. If the length of the clause is 0, return false because it is unsatisfiable because there are no literals left to choose. The algorithm then chooses the next literal. If there are no more literals left to choose, it means it is satisfiable and returns true. Lastly, but importantly, there is the recursive call for two cases. One case assumes the literal is true and propagates. The other assumes the literal is false and propagates.</p> <p>After calling DPLL, the main also calls a function to print the output. The output file is in the form of a csv. It was easy to print because I stored all of the information in a dataclass.</p> <p>After this, a plotting function is called. The plotting function uses Matplotlib. I also generated a trendline that was a polynomial trendline (because this is a polynomial time solver). I initially assigned it to degree two and used numpy functions to do this.</p>
11	<p>What test cases you used/added, why you used them, what did they tell you about the correctness of your code.</p> <p>For an initial test, I made my own cnf file with 3 cases:</p> <pre> c,1000,2,? p,cnf,2,3 -2,4,0, -2,4,0 2,-4,0 c,1001,2,? p,cnf,2,4 -2,-3,0, 3,2,0, -2,3,0, 2,-3,0, c,1002,3,? p,cnf,3,4 4,5,0, 6,-5,0, </pre>

	<p>-4,5,0, 5,-6,0, The first one is definitely satisfiable because the assignment 2 = True and 4 = True leads to a True result overall. The second case is unsatisfiable because it is composed of every combination of the two variables, so it would be impossible there to be an assignment that leads to a True result overall. If they both are true, it is false. If the both are false, it is false. If the 3 is true and the 2 is false, it is false. If the 2 is true and the 3 is false it is false. The final case is satisfiable because the assignment 4 = T 5 = T 6 = T would make the overall wff true. I ran the code on these and it got the correct results.</p> <p>For more robust testing, I used the provided test cases that were given in Canvas for my primary testing. The canvas file provided: 2SAT.cnf.csv, was renamed to fit the name convention. This has 100 test cases of varying sizes, so I was considered this to be sufficient testing. To determine the correct answer, I actually ran the same exact input through DumbSAT and compared the results. Because I was told that DumbSAT was a correct algorithm and it was provided, this was very helpful to use as a comparison. However, the downside is that DumbSAT took about 1 hour and 15 minutes to run on my computer. DumbSAT was initially set up to generate its own test cases randomly, so I had to make some modifications. I took the code I used from my 2SAT solver to parse the input file and called the DumbSAT function. With my WFF dataclass, it was easy to give DumbSAT all of the information it needed to run. Then, I used the same function I wrote from the 2SAT solver that outputs the information to output DumbSAT results. This output file has the number of literals, the computation time, and the satisfiability. After directly comparing satisfiability with DumbSAT, I found them to be identical for all 100 test cases. I felt that 100 test cases was sufficient to consider my output correct. Additionally, my graph appeared to follow the expected trends (discussed in the result discussion section), so I felt pretty good about the correctness.</p>
12	<p>How you managed the code development</p> <p>I completed all of the code myself because I was an individual team. To develop the code, I first focused on developing code to read in the file. I originally ran my code in the terminal using VScode. I later switched to Google Collabs because I was unsure if I could graph in Vscode, and I knew that I had graphed in google collabs before. A major first step in the development of the code was developing the code to read in data from the csv file that was in cnf format. I spent a long time trying to understand what this format was saying. In order to read it in, I ended up making a dataclass for the wff, which allowed me to store lots of information easily about the wff including the number of literals, the number of variables, the runtime, the clauses, and the satisfiability itself. This was very useful because it made it very easy to generate the output files because I stored the information in an organized manner. Then, I developed code to print the results and generate the result file. Then, to verify that it worked, I actually tested it by running my code with DumbSAT algorithm that was provided. I modified it slightly so it would work with the input in the cnf form instead of generating its own test cases like it did in the provided file. After verifying that it worked, I worked on the DPLL algorithm. I was not familiar with this algorithm at all, so I looked online for more information. The wikipedia page was very helpful: https://en.wikipedia.org/wiki/DPLL_algorithm After looking at various pseudocode, I attempted to implement it. To debug, I wrote many print statements printing out the information that I had so I could trace where the</p>

error was occurring. I timed the code within the functions and stored all the info, so it was very easy to generate the output files. I then used matplotlib to generate the graph. For testing, I thought of a few test cases and ran them. My main was of testing was to actually run the same input (the provided test cases) through DumbSAT, which I had modified a little, and this created a solution that I could compare my code with.

13 Detailed discussion of results:



My results were accurate because I checked by running the exact same code through DumbSAT for a comparison. In general, the DPLL algorithm in the worst case is $O(2^n)$, where n is the number of variable in the formula. However, for 2SAT, a special case, it is expected to be more linear for most cases because the clauses are short and this leads to a lot of logical shortcuts through the unit propagation and pure literal elimination techniques.

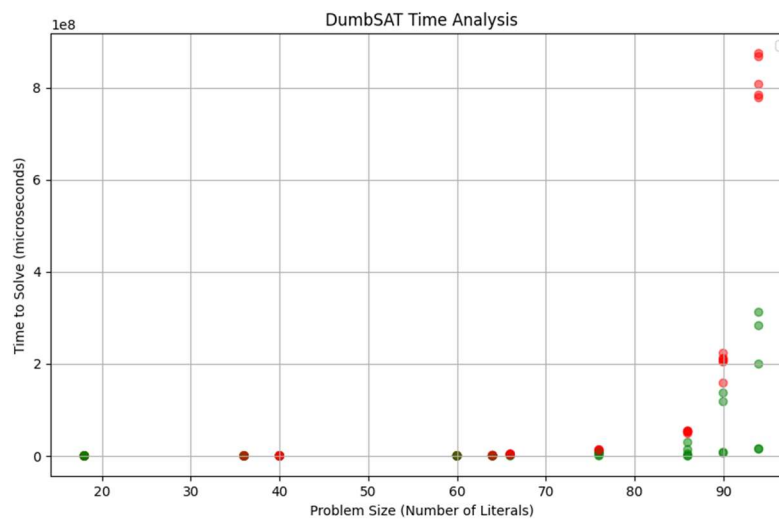
The 2-Sat problem can be solved in polynomial time. It can be solved in $O(n+m)$, where n is the number of variables and m is the number of clauses. It is polynomial time, even in the worst case. It is possible to get linear. This approach is P (Not NP) because it is in polynomial time. 2-SAT is faster because if one literal is false, the other must be true. There is a domino effect and it is easier to determine assignments than other cases.

I generated multiple plots because the exact computation time varies a little, so the curvature varies. However, it is clear that you can see that as the number of literals

increase, the computation time directly increases. One of my plots does appear to be pretty close to linear, while the other is slightly more curved. I used a polynomial degree trendline to be fit to my graph and it appears to be a good fit, so this confirms that my solution was in polynomial time.

As a comparison, I also graphed DumbSAT. DumbSAT is actually exponential and is $O(2^{\text{variables} * \text{clauses} * \text{average literals}})$. As the number of literals gets bigger, it slows down completely. DumbSAT took over an hour to run on my computer, while my 2SAT solver took a couple of seconds. Clearly, my algorithm is a good speedup.

Unfortunately, I didn't get to add a trend line to DumbSAT because I was having trouble figuring out how to make an exponential fit and it took several hours each time I reran the code. However, you can clearly see that it is exponential and that it goes up very steeply as the problem size gets bigger. Towards the end, the program took many minutes to run and the program as a whole took hours even though the input was identical to the 2SAT input.



In order to view the data with an exponential trendline, I quickly graphed it in Excel but couldn't figure out how to color it based on satisfiable / unsatisfiable. The exponential trendline seemed to fit pretty well.

	<p style="text-align: center;">DumbSat: Time vs. Problem Size</p> <table border="1"> <caption>Approximate data points from the scatter plot</caption> <thead> <tr> <th>Number of Literals</th> <th>Time (Microseconds)</th> </tr> </thead> <tbody> <tr><td>18</td><td>~1,000,000,000</td></tr> <tr><td>38</td><td>~1,000,000,000</td></tr> <tr><td>42</td><td>~1,000,000,000</td></tr> <tr><td>62</td><td>~1,000,000,000</td></tr> <tr><td>65</td><td>~1,000,000,000</td></tr> <tr><td>68</td><td>~1,000,000,000</td></tr> <tr><td>78</td><td>~1,000,000,000</td></tr> <tr><td>88</td><td>~1,000,000,000</td></tr> <tr><td>90</td><td>~1,000,000,000</td></tr> <tr><td>92</td><td>~1,000,000,000</td></tr> <tr><td>94</td><td>~1,000,000,000</td></tr> <tr><td>95</td><td>~1,000,000,000</td></tr> <tr><td>96</td><td>~1,000,000,000</td></tr> <tr><td>97</td><td>~1,000,000,000</td></tr> <tr><td>98</td><td>~1,000,000,000</td></tr> <tr><td>99</td><td>~1,000,000,000</td></tr> <tr><td>100</td><td>~1,000,000,000</td></tr> <tr><td>101</td><td>~1,000,000,000</td></tr> <tr><td>102</td><td>~1,000,000,000</td></tr> <tr><td>103</td><td>~1,000,000,000</td></tr> <tr><td>104</td><td>~1,000,000,000</td></tr> <tr><td>105</td><td>~1,000,000,000</td></tr> <tr><td>106</td><td>~1,000,000,000</td></tr> <tr><td>107</td><td>~1,000,000,000</td></tr> <tr><td>108</td><td>~1,000,000,000</td></tr> <tr><td>109</td><td>~1,000,000,000</td></tr> <tr><td>110</td><td>~1,000,000,000</td></tr> <tr><td>111</td><td>~1,000,000,000</td></tr> <tr><td>112</td><td>~1,000,000,000</td></tr> <tr><td>113</td><td>~1,000,000,000</td></tr> <tr><td>114</td><td>~1,000,000,000</td></tr> <tr><td>115</td><td>~1,000,000,000</td></tr> <tr><td>116</td><td>~1,000,000,000</td></tr> <tr><td>117</td><td>~1,000,000,000</td></tr> <tr><td>118</td><td>~1,000,000,000</td></tr> <tr><td>119</td><td>~1,000,000,000</td></tr> <tr><td>120</td><td>~1,000,000,000</td></tr> <tr><td>121</td><td>~1,000,000,000</td></tr> <tr><td>122</td><td>~1,000,000,000</td></tr> <tr><td>123</td><td>~1,000,000,000</td></tr> <tr><td>124</td><td>~1,000,000,000</td></tr> <tr><td>125</td><td>~1,000,000,000</td></tr> <tr><td>126</td><td>~1,000,000,000</td></tr> <tr><td>127</td><td>~1,000,000,000</td></tr> <tr><td>128</td><td>~1,000,000,000</td></tr> <tr><td>129</td><td>~1,000,000,000</td></tr> <tr><td>130</td><td>~1,000,000,000</td></tr> <tr><td>131</td><td>~1,000,000,000</td></tr> <tr><td>132</td><td>~1,000,000,000</td></tr> <tr><td>133</td><td>~1,000,000,000</td></tr> <tr><td>134</td><td>~1,000,000,000</td></tr> <tr><td>135</td><td>~1,000,000,000</td></tr> <tr><td>136</td><td>~1,000,000,000</td></tr> <tr><td>137</td><td>~1,000,000,000</td></tr> <tr><td>138</td><td>~1,000,000,000</td></tr> <tr><td>139</td><td>~1,000,000,000</td></tr> <tr><td>140</td><td>~1,000,000,000</td></tr> <tr><td>141</td><td>~1,000,000,000</td></tr> <tr><td>142</td><td>~1,000,000,000</td></tr> <tr><td>143</td><td>~1,000,000,000</td></tr> <tr><td>144</td><td>~1,000,000,000</td></tr> <tr><td>145</td><td>~1,000,000,000</td></tr> <tr><td>146</td><td>~1,000,000,000</td></tr> <tr><td>147</td><td>~1,000,000,000</td></tr> <tr><td>148</td><td>~1,000,000,000</td></tr> <tr><td>149</td><td>~1,000,000,000</td></tr> <tr><td>150</td><td>~1,000,000,000</td></tr> <tr><td>151</td><td>~1,000,000,000</td></tr> <tr><td>152</td><td>~1,000,000,000</td></tr> <tr><td>153</td><td>~1,000,000,000</td></tr> <tr><td>154</td><td>~1,000,000,000</td></tr> <tr><td>155</td><td>~1,000,000,000</td></tr> <tr><td>156</td><td>~1,000,000,000</td></tr> <tr><td>157</td><td>~1,000,000,000</td></tr> <tr><td>158</td><td>~1,000,000,000</td></tr> <tr><td>159</td><td>~1,000,000,000</td></tr> <tr><td>160</td><td>~1,000,000,000</td></tr> <tr><td>161</td><td>~1,000,000,000</td></tr> <tr><td>162</td><td>~1,000,000,000</td></tr> <tr><td>163</td><td>~1,000,000,000</td></tr> <tr><td>164</td><td>~1,000,000,000</td></tr> <tr><td>165</td><td>~1,000,000,000</td></tr> <tr><td>166</td><td>~1,000,000,000</td></tr> <tr><td>167</td><td>~1,000,000,000</td></tr> <tr><td>168</td><td>~1,000,000,000</td></tr> <tr><td>169</td><td>~1,000,000,000</td></tr> <tr><td>170</td><td>~1,000,000,000</td></tr> <tr><td>171</td><td>~1,000,000,000</td></tr> <tr><td>172</td><td>~1,000,000,000</td></tr> <tr><td>173</td><td>~1,000,000,000</td></tr> <tr><td>174</td><td>~1,000,000,000</td></tr> <tr><td>175</td><td>~1,000,000,000</td></tr> <tr><td>176</td><td>~1,000,000,000</td></tr> <tr><td>177</td><td>~1,000,000,000</td></tr> <tr><td>178</td><td>~1,000,000,000</td></tr> <tr><td>179</td><td>~1,000,000,000</td></tr> <tr><td>180</td><td>~1,000,000,000</td></tr> <tr><td>181</td><td>~1,000,000,000</td></tr> <tr><td>182</td><td>~1,000,000,000</td></tr> <tr><td>183</td><td>~1,000,000,000</td></tr> <tr><td>184</td><td>~1,000,000,000</td></tr> <tr><td>185</td><td>~1,000,000,000</td></tr> <tr><td>186</td><td>~1,000,000,000</td></tr> <tr><td>187</td><td>~1,000,000,000</td></tr> <tr><td>188</td><td>~1,000,000,000</td></tr> <tr><td>189</td><td>~1,000,000,000</td></tr> <tr><td>190</td><td>~1,000,000,000</td></tr> <tr><td>191</td><td>~1,000,000,000</td></tr> <tr><td>192</td><td>~1,000,000,000</td></tr> <tr><td>193</td><td>~1,000,000,000</td></tr> <tr><td>194</td><td>~1,000,000,000</td></tr> <tr><td>195</td><td>~1,000,000,000</td></tr> <tr><td>196</td><td>~1,000,000,000</td></tr> <tr><td>197</td><td>~1,000,000,000</td></tr> <tr><td>198</td><td>~1,000,000,000</td></tr> <tr><td>199</td><td>~1,000,000,000</td></tr> <tr><td>200</td><td>~1,000,000,000</td></tr> </tbody> </table>	Number of Literals	Time (Microseconds)	18	~1,000,000,000	38	~1,000,000,000	42	~1,000,000,000	62	~1,000,000,000	65	~1,000,000,000	68	~1,000,000,000	78	~1,000,000,000	88	~1,000,000,000	90	~1,000,000,000	92	~1,000,000,000	94	~1,000,000,000	95	~1,000,000,000	96	~1,000,000,000	97	~1,000,000,000	98	~1,000,000,000	99	~1,000,000,000	100	~1,000,000,000	101	~1,000,000,000	102	~1,000,000,000	103	~1,000,000,000	104	~1,000,000,000	105	~1,000,000,000	106	~1,000,000,000	107	~1,000,000,000	108	~1,000,000,000	109	~1,000,000,000	110	~1,000,000,000	111	~1,000,000,000	112	~1,000,000,000	113	~1,000,000,000	114	~1,000,000,000	115	~1,000,000,000	116	~1,000,000,000	117	~1,000,000,000	118	~1,000,000,000	119	~1,000,000,000	120	~1,000,000,000	121	~1,000,000,000	122	~1,000,000,000	123	~1,000,000,000	124	~1,000,000,000	125	~1,000,000,000	126	~1,000,000,000	127	~1,000,000,000	128	~1,000,000,000	129	~1,000,000,000	130	~1,000,000,000	131	~1,000,000,000	132	~1,000,000,000	133	~1,000,000,000	134	~1,000,000,000	135	~1,000,000,000	136	~1,000,000,000	137	~1,000,000,000	138	~1,000,000,000	139	~1,000,000,000	140	~1,000,000,000	141	~1,000,000,000	142	~1,000,000,000	143	~1,000,000,000	144	~1,000,000,000	145	~1,000,000,000	146	~1,000,000,000	147	~1,000,000,000	148	~1,000,000,000	149	~1,000,000,000	150	~1,000,000,000	151	~1,000,000,000	152	~1,000,000,000	153	~1,000,000,000	154	~1,000,000,000	155	~1,000,000,000	156	~1,000,000,000	157	~1,000,000,000	158	~1,000,000,000	159	~1,000,000,000	160	~1,000,000,000	161	~1,000,000,000	162	~1,000,000,000	163	~1,000,000,000	164	~1,000,000,000	165	~1,000,000,000	166	~1,000,000,000	167	~1,000,000,000	168	~1,000,000,000	169	~1,000,000,000	170	~1,000,000,000	171	~1,000,000,000	172	~1,000,000,000	173	~1,000,000,000	174	~1,000,000,000	175	~1,000,000,000	176	~1,000,000,000	177	~1,000,000,000	178	~1,000,000,000	179	~1,000,000,000	180	~1,000,000,000	181	~1,000,000,000	182	~1,000,000,000	183	~1,000,000,000	184	~1,000,000,000	185	~1,000,000,000	186	~1,000,000,000	187	~1,000,000,000	188	~1,000,000,000	189	~1,000,000,000	190	~1,000,000,000	191	~1,000,000,000	192	~1,000,000,000	193	~1,000,000,000	194	~1,000,000,000	195	~1,000,000,000	196	~1,000,000,000	197	~1,000,000,000	198	~1,000,000,000	199	~1,000,000,000	200	~1,000,000,000
Number of Literals	Time (Microseconds)																																																																																																																																																																																																																																												
18	~1,000,000,000																																																																																																																																																																																																																																												
38	~1,000,000,000																																																																																																																																																																																																																																												
42	~1,000,000,000																																																																																																																																																																																																																																												
62	~1,000,000,000																																																																																																																																																																																																																																												
65	~1,000,000,000																																																																																																																																																																																																																																												
68	~1,000,000,000																																																																																																																																																																																																																																												
78	~1,000,000,000																																																																																																																																																																																																																																												
88	~1,000,000,000																																																																																																																																																																																																																																												
90	~1,000,000,000																																																																																																																																																																																																																																												
92	~1,000,000,000																																																																																																																																																																																																																																												
94	~1,000,000,000																																																																																																																																																																																																																																												
95	~1,000,000,000																																																																																																																																																																																																																																												
96	~1,000,000,000																																																																																																																																																																																																																																												
97	~1,000,000,000																																																																																																																																																																																																																																												
98	~1,000,000,000																																																																																																																																																																																																																																												
99	~1,000,000,000																																																																																																																																																																																																																																												
100	~1,000,000,000																																																																																																																																																																																																																																												
101	~1,000,000,000																																																																																																																																																																																																																																												
102	~1,000,000,000																																																																																																																																																																																																																																												
103	~1,000,000,000																																																																																																																																																																																																																																												
104	~1,000,000,000																																																																																																																																																																																																																																												
105	~1,000,000,000																																																																																																																																																																																																																																												
106	~1,000,000,000																																																																																																																																																																																																																																												
107	~1,000,000,000																																																																																																																																																																																																																																												
108	~1,000,000,000																																																																																																																																																																																																																																												
109	~1,000,000,000																																																																																																																																																																																																																																												
110	~1,000,000,000																																																																																																																																																																																																																																												
111	~1,000,000,000																																																																																																																																																																																																																																												
112	~1,000,000,000																																																																																																																																																																																																																																												
113	~1,000,000,000																																																																																																																																																																																																																																												
114	~1,000,000,000																																																																																																																																																																																																																																												
115	~1,000,000,000																																																																																																																																																																																																																																												
116	~1,000,000,000																																																																																																																																																																																																																																												
117	~1,000,000,000																																																																																																																																																																																																																																												
118	~1,000,000,000																																																																																																																																																																																																																																												
119	~1,000,000,000																																																																																																																																																																																																																																												
120	~1,000,000,000																																																																																																																																																																																																																																												
121	~1,000,000,000																																																																																																																																																																																																																																												
122	~1,000,000,000																																																																																																																																																																																																																																												
123	~1,000,000,000																																																																																																																																																																																																																																												
124	~1,000,000,000																																																																																																																																																																																																																																												
125	~1,000,000,000																																																																																																																																																																																																																																												
126	~1,000,000,000																																																																																																																																																																																																																																												
127	~1,000,000,000																																																																																																																																																																																																																																												
128	~1,000,000,000																																																																																																																																																																																																																																												
129	~1,000,000,000																																																																																																																																																																																																																																												
130	~1,000,000,000																																																																																																																																																																																																																																												
131	~1,000,000,000																																																																																																																																																																																																																																												
132	~1,000,000,000																																																																																																																																																																																																																																												
133	~1,000,000,000																																																																																																																																																																																																																																												
134	~1,000,000,000																																																																																																																																																																																																																																												
135	~1,000,000,000																																																																																																																																																																																																																																												
136	~1,000,000,000																																																																																																																																																																																																																																												
137	~1,000,000,000																																																																																																																																																																																																																																												
138	~1,000,000,000																																																																																																																																																																																																																																												
139	~1,000,000,000																																																																																																																																																																																																																																												
140	~1,000,000,000																																																																																																																																																																																																																																												
141	~1,000,000,000																																																																																																																																																																																																																																												
142	~1,000,000,000																																																																																																																																																																																																																																												
143	~1,000,000,000																																																																																																																																																																																																																																												
144	~1,000,000,000																																																																																																																																																																																																																																												
145	~1,000,000,000																																																																																																																																																																																																																																												
146	~1,000,000,000																																																																																																																																																																																																																																												
147	~1,000,000,000																																																																																																																																																																																																																																												
148	~1,000,000,000																																																																																																																																																																																																																																												
149	~1,000,000,000																																																																																																																																																																																																																																												
150	~1,000,000,000																																																																																																																																																																																																																																												
151	~1,000,000,000																																																																																																																																																																																																																																												
152	~1,000,000,000																																																																																																																																																																																																																																												
153	~1,000,000,000																																																																																																																																																																																																																																												
154	~1,000,000,000																																																																																																																																																																																																																																												
155	~1,000,000,000																																																																																																																																																																																																																																												
156	~1,000,000,000																																																																																																																																																																																																																																												
157	~1,000,000,000																																																																																																																																																																																																																																												
158	~1,000,000,000																																																																																																																																																																																																																																												
159	~1,000,000,000																																																																																																																																																																																																																																												
160	~1,000,000,000																																																																																																																																																																																																																																												
161	~1,000,000,000																																																																																																																																																																																																																																												
162	~1,000,000,000																																																																																																																																																																																																																																												
163	~1,000,000,000																																																																																																																																																																																																																																												
164	~1,000,000,000																																																																																																																																																																																																																																												
165	~1,000,000,000																																																																																																																																																																																																																																												
166	~1,000,000,000																																																																																																																																																																																																																																												
167	~1,000,000,000																																																																																																																																																																																																																																												
168	~1,000,000,000																																																																																																																																																																																																																																												
169	~1,000,000,000																																																																																																																																																																																																																																												
170	~1,000,000,000																																																																																																																																																																																																																																												
171	~1,000,000,000																																																																																																																																																																																																																																												
172	~1,000,000,000																																																																																																																																																																																																																																												
173	~1,000,000,000																																																																																																																																																																																																																																												
174	~1,000,000,000																																																																																																																																																																																																																																												
175	~1,000,000,000																																																																																																																																																																																																																																												
176	~1,000,000,000																																																																																																																																																																																																																																												
177	~1,000,000,000																																																																																																																																																																																																																																												
178	~1,000,000,000																																																																																																																																																																																																																																												
179	~1,000,000,000																																																																																																																																																																																																																																												
180	~1,000,000,000																																																																																																																																																																																																																																												
181	~1,000,000,000																																																																																																																																																																																																																																												
182	~1,000,000,000																																																																																																																																																																																																																																												
183	~1,000,000,000																																																																																																																																																																																																																																												
184	~1,000,000,000																																																																																																																																																																																																																																												
185	~1,000,000,000																																																																																																																																																																																																																																												
186	~1,000,000,000																																																																																																																																																																																																																																												
187	~1,000,000,000																																																																																																																																																																																																																																												
188	~1,000,000,000																																																																																																																																																																																																																																												
189	~1,000,000,000																																																																																																																																																																																																																																												
190	~1,000,000,000																																																																																																																																																																																																																																												
191	~1,000,000,000																																																																																																																																																																																																																																												
192	~1,000,000,000																																																																																																																																																																																																																																												
193	~1,000,000,000																																																																																																																																																																																																																																												
194	~1,000,000,000																																																																																																																																																																																																																																												
195	~1,000,000,000																																																																																																																																																																																																																																												
196	~1,000,000,000																																																																																																																																																																																																																																												
197	~1,000,000,000																																																																																																																																																																																																																																												
198	~1,000,000,000																																																																																																																																																																																																																																												
199	~1,000,000,000																																																																																																																																																																																																																																												
200	~1,000,000,000																																																																																																																																																																																																																																												
14	<p>How team was organized</p> <p>Because I was an individual team (a team of 1), I completed all of the work myself. I was responsible for doing the main code to read in the input, I created the DPLL algorithm code, and I generated the output. I also generated the graphs and the tests. I also modified DumbSat for a comparison. I was in charge of verifying everything. Overall, the organization was mostly good. One area of improvement is that I switched my project 3 different times, so I would try to organize better and be more decisive in the future.</p>																																																																																																																																																																																																																																												
15	<p>What you might do differently if you did the project again</p> <p>If I did this project again, I would definitely start the project earlier. It was very stressful to wait until the last minute to start it, and it was difficult to rush the process. Additionally, I would take more time when selecting which project I would do. I switched projects 3 times out of indecisiveness and wasted a lot of time. Lastly, I would try harder to understand the DPLL algorithm first before coding it. I jumped into coding it too quickly without fully understanding, and it took a while to figure it out and debug.</p>																																																																																																																																																																																																																																												
16	<p>Any additional material:</p> <p>Professor Kogge mentioned including a screenshot of our console to prove that the code actually ran. I am not sure where to include this, so I will include it here:</p>																																																																																																																																																																																																																																												

Copy of ... Project ... Project ... HW06 ... Home ... check ... code_2 ... Catheri ... is a list ...

colab.research.google.com/drive/18wG-WeNkK0eH_LR3_6X0lQf1oujgD_m3#scrollTo=-A1x_mEKojmm&uniquifier=1

Apps

LinkedIn

RecRegister

LeetCode - The Wor...

Free Grammar Chec...

Gmail

YouTube

Flaherty Mass Plann...

Comp Arch Schedu...

All Bookmarks

Files

sample_data

data_cnf_chealy5.csv

output_chealy5.csv

plots_chealy5.png

75.44 GB available

Code

Text

```

227
228
229
230
231
232
233
234
235
236
# Generate the output file with the times and whether Satisf
generate_csv_output(my_wffs)
# Plot the timing analysis
make_plot(my_wffs)
if __name__ == '__main__':
    main()

```

plot.png

output_chealy5.csv

1 to 10 of 100 entries

Number of Literals	Execution Time	Sati
18	35	S
18	34	U
18	24	S
18	18	S
18	43	U
18	23	U
18	28	U
18	14	S
18	18	S
18	22	S

Show 10 per page

1 2 3 4 5 6 7 8

Solution Computation Time vs. Problem Size

0s completed at 8:36 PM

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.

60°

Windows icons

8:36 PM 10/18/2024