



**CALIFORNIA STATE SCIENCE FAIR  
2008 PROJECT SUMMARY**

<b>Name(s)</b> <b>Nathan J. Manohar</b>	<b>Project Number</b> <b>J1310</b>
<b>Project Title</b> <b>Writing a Rubik's Cube Solver using Mathematica</b>	
<b>Objectives/Goals</b> I enjoy solving the Rubik's cube, and I have developed my own method for solving it. I am interested in mathematics and have written programs to solve Sudoku puzzles and magic squares. I decided to try to write a computer program that solves the Rubik's cube. I wrote the program using the Mathematica software package because Mathematica has built in graphics commands that allowed me to draw the Rubik's cube on a computer screen, and rotate the image using the mouse.	
<b>Abstract</b> <b>Methods/Materials</b> The program I wrote uses 56 simple steps that can be used to solve the cube. I used a recursive algorithm, which applies the 56 basic steps multiple times. This made the program powerful without having to write a large amount of computer code. I then generated a random cube by performing n number of random moves on a completely solved Rubik's cube. I then applied my computer program to solve this scrambled cube. I wrote a small program to repeat this process multiple times and save the number of moves needed to solve the cube. I then took the average and standard deviation of these values.	
<b>Results</b> I noticed that after the cube had been mixed up by twenty or more moves, the average number of moves needed to solve the cube remained constant at 140 moves. This result shows that after performing twenty random moves, the cube is as scrambled as it can be.	
<b>Conclusions/Discussion</b> After running the program 25 million times, the highest number of moves needed to solve the Rubik's cube was 222. This suggests that my program can solve the Rubik's cube in or under 222 moves every time. My program can usually solve a Rubik's cube in about 140 moves. On the computer, this takes less than 0.04 seconds! After testing my Rubik's cube solver over 25 million times, I programmed the animations for my Rubik's cube solver so that I could visualize the solutions.	
<b>Summary Statement</b> I wrote a computer program to solve a Rubik's cube, evaluated the results, then wrote a program to display animations of the solutions.	
<b>Help Received</b> Father taught me the basics of programming when I was ten; Uncle wrote a program with me that solves Sudoku puzzles.	