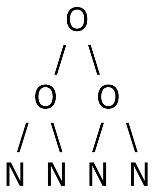
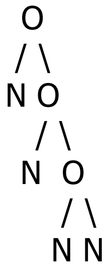
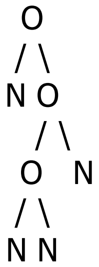
Stuart Smith

5/16/2013

24 - The Card Game

Twenty-four is an algebra game that is played with a standard deck of cards. A player four cards into their hand and is tasked to find a way to make the values of the cards, from 1 to 9, equal 24 using simple mathematics by adding, subtracting, multiplying, and dividing the face value of the cards, with or without parenthesis. Ace cards are valued at 1. A group of players can play together as well, in this case four cards are drawn into a pile, the first player to formulate an expression to make the cards equal 24 picks up the pile. The player with the most cards at the end of the game, wins. There are other versions of that game that include face cards J, Q, and K, that are valued at 11, 12, and 13 respectively; some versions of the game do not include the face cards. Also more mathematical operations can be utilized to get 24, like exponential and logarithmic operations.

Implementing this game on the computer requires knowledge of certain algorithms. Humans can sometimes easily figure out a solution to the algebra problem, but computers do not work like human brains. I will use a brute force algorithm to implement a solution to make sure a problem is solvable. When four cards are chosen the computer will check if they are able to be solved into a solution of 24 before they are dealt to the user. If there exists at least one solution to a set of four cards the cards will proceed to be dealt. The solution is checked by iterating through all possible solutions possible for a game. Given a set of four cards in a standard sized deck of 52 cards there are $4!$ (four factorial) possible combinations of numbers that can be dealt. Between the four cards, there are 4 possible operations (adding, subtract, multiply, and divide) that can be performed on them to make a solution of 24. That’s $4 * 4 * 4$ possible operations, since only 3 operations can be used between four cards. The last thing to check is the number of possible arrangements of the numbers. To get a picture of the solutions we can use trees.

	
 (A mirrored version of this tree is also a solution)	 (A mirrored version of this tree is also a solution)

In the trees above, O stands for the operations possible, N stands for the possible numbers that can be distributed. Since multiplication and addition are commutative in an equation, they can create disjoint solutions, so the first and second trees have two possible arrangements depending on if an O is addition or multiplication. Given the trees above, there are 5 possible ways to arrange the four numbers and operations to get a solution of 24. With the information provided, it is known that there are $4! * 4^3 * 5$ possible solutions to iterate through using brute force; that’s 7,680 solutions in total. In short, the computer will randomly choose 4 numbers to be dealt, it will iterate through the 7,680 possible solutions for at least one solution before displaying those 4 numbers to the user. At that point it is up to the user to arrange those numbers to come up with 24.

Here is the algorithm I will use to write the program:

```
procedure brute_force($n_1, n_2, n_3 \ldots n_9$: possible numbers)
x1, x2, x3, x4 := $n_{x_1}, n_{x_2}, n_{x_3}, n_{x_4}$
parentheses := “(”,”)”
for i := 1 to 4
     for j := 1 to 4
          for k := 1 to 4
               for p := (length of parenthesis) * 3
                    if p = 0
                         x = p1 +x1 + i +x2 + p2 + j + x3 + k
                               + x4
                    if p = 1
                         x = p1 +x1 + i +x2 + j + x3 + p2 + k
                              + x4
                    if p = 2
                         x = x1 + i + p1 + x2 + j + x3 + k
                              + x4 + p2
                    if p = 3
                         x = x1 + i + p1 + x2 + j + x3 + p2 +
                              k + x4
                    if p = 4
                         x = x1 + i +x2 + p2 + j + p1 + x3 +
                              k + x4 + p2
                    if p = 5
```

```
                         x = p1 +x1 + i +x2 + p2 + j + p1 +
                          x3 + k + x4 + p2
          if x = 24
               return x
return 0
```

Four randomly generated numbers will be passed into the procedure and the algorithm cycles through each of the 4 operations and applies all possible combinations to the four numbers passed into the algorithm. This procedure essentially creates partitions between the operations, including placement of parentheses around certain parts of the equation. If `x` is returned, it means it is possible to create 24 with the numbers generated. Once `x` is returned, it is displayed to the user and it is up to that person to come up with an equation.