• This exam is worth 20% of your final grade (500/2500 points).
• There are 7 problems (and 1 BONUS problem).
• You have until 12:30 to finish.
• You may use the final exam cheat sheet I’ve provided, and calculators should be unnecessary.
• Be sure to show your work, as I’ll be giving partial credit.
(1) (50 points)

Prove (using induction) that

\[ \sum_{i=1}^{n} i^2 = \frac{n(n + 1)(2n + 1)}{6} \]

Answer:

Base Case: \( \frac{1(1+1)(2+1)}{6} = 1 = 1^2 \)

Inductive step: Assume that \( \sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6} \) for \( k = 1 \) to \( n - 1 \). Then

\[
\begin{align*}
\sum_{i=1}^{n} i^2 &= n^2 + \sum_{i=1}^{n-1} i^2 \\
&= n^2 + \frac{(n-1)((n-1) + 1)(2(n-1) + 1)}{6} \\
&= n^2 + \frac{(n-1)n(2n-1)}{6} \\
&= \frac{6n^2 + (n-1)n(2n-1)}{6} \\
&= \frac{n(6n + (n-1)(2n-1))}{6} \\
&= \frac{n(6n + (2n^2 - 3n + 1))}{6} \\
&= \frac{n(2n^2 + 3n + 1)}{6} \\
&= \frac{n(n + 1)(2n + 1)}{6}
\end{align*}
\]
(2) (50 points) Of the \( \frac{11!}{2!2!} \) ways to arrange the letters in the word **SERENDIPITY**, how many of them start and end with the same letter?

**Answer:** The letter either has to be E or I. In the case that it’s E, there are \( \frac{9!}{2!} \) ways to arrange the remaining letters, and likewise for the case the letter’s I. So there are \( 2 \cdot \frac{9!}{2!} = 9! \) arrangements that start and end with the same letter.
(3) (50 points) Set up a recurrence relation to find the number of ways to climb \( n \) stairs if the person climbing the stairs can take 1, 2, or 3 stairs at a time. Don’t forget to set up the base case(s).

**Answer:** The base case is that a person has 1 way of climbing 1 stair. So \( r_1 = 1 \). There are 2 ways to climb 2 stairs, and 4 ways to climb 3 stairs (1+1+1, 2+1, 1+2, and 3). For \( n \) more than 3, a person can climb 1 stair from \( n-1 \), 2 stairs from \( n-2 \), and 3 stairs from \( n-3 \). So the relation is

\[
r_n = r_{n-1} + r_{n-2} + r_{n-3}
\]
(4) (100 points total plus 40 BONUS points) Consider the game called the Onion Rings of Hanoi. In this game, there are 3 onion rings with different sizes: wee, not-so-wee, and FRIGGIN-HUGE. There are 3 poles on which the onion rings can be stacked: “Neutral”, “Home”, and “Visitor”. Initially, all 3 rings are stacked (wee atop not-so-wee atop FRIGGIN-HUGE) on the Neutral pole. The players, Home and Visitor, take turns moving the onion rings according to the following rules:

- Each turn, a player must move exactly 1 onion ring from one pole to another.
- A player can only move the topmost onion ring from any stack.
- A player can’t put the not-so-wee onion ring atop the wee onion ring, and he or she can’t put the FRIGGIN-HUGE onion ring atop either of the other 2.

The winner of the game is the one who first gets the entire stack of onion rings onto his or her pole. Home goes first.

Consider the graph where each node represents a game state, and there’s a directed edge between 2 nodes if it’s possible to immediately transition from one state to another.

(a) (50 points) How many nodes are in the game graph? (Remember that whose turn it is is part of the state.)

**Answer:** Each of the 3 onion rings can be on one of the 3 poles. This gives $3^3 = 27$ positions for the onion rings. Additionally, it can be either player’s turn, so this gives 54 nodes in the graph. (We know that each of these positions can be reached because Home can move wee onto his own peg, then Visitor can move it onto his peg, then Home can move it back onto Neutral thus winding up as if Visitor had started the game.)

(b) (50 points) Is the game graph bipartite? If so, what are the 2 partitions of nodes?
**Answer:** Yes, the graph is bipartite. There are nodes where it’s Home’s turn and nodes where it’s Visitor’s turn, and since neither can go twice in a row, these sets are disjoint.

(c) (40 BONUS points) If we make it so there are only 2 rings, and we introduce a new rule that says that a player loses if he or she puts the onion rings into a position in which they already were (regardless of whose turn it was) what’s the deepest that a game tree will be?

**Answer:** The onion configuration graph looks like this:

```
N
/ \
*___* \\
/ \\
* * \\
/ \\
*___*___*___*
```

where **N** is the start state, and **H** and **V** are the cases where all the rings are on Home’s and Visitor’s poles, respectively. From this, we see that the longest path that ends when it gets to **H** or **V** is 8 nodes. So, the deepest the search tree will be is 8 nodes (or 7 edges).
(5) (50 points) Using only the 6 following symbols $a$, $b$, $(, )$, $F$, and $\implies$ (where $F$ is always false), construct an expression that’s equivalent to $a \land b$ (thus proving that \textbf{implies} and \textbf{False} make a universal boolean operator set).

**Answer:** Here’s one way to do it.

$$a \land b = b \implies (a \implies F)$$
(6) (100 points total) Consider the grammar $G$ with starting nonterminal $<S>$:

$$
G : \begin{align*}
<S> & ::= A<S> \mid B<S> \mid <M> \\
<M> & ::= AAA \mid B 
\end{align*}
$$

(a) (10 points) Is the string $AAABABBABAA$ in $G$’s language?

**Answer:** No, $G$’s language is all the strings over $A$ and $B$ that end in $AAA$ or $B$.

(b) (10 points) Is the string $ABBBABBBAAA$ in $G$’s language?

**Answer:** Yes.

(c) (40 points) Draw the parse tree for the string $AAAA$.

**Answer:**

```
    <S>
     / \ 
    A  <S> 
     /  
    A  <M> 
     /  
    AAA
```

(d) (40 points) Is $G$ regular? That is, is there a regular expression that has the same language as $G$? Either prove that $G$ is not regular, or give a regular expression that has the same language as $G$.

**Answer:** Yes, the regular expression is $(A|B)^*(B|AAA)$. 
(7) (100 points)

How many different programs are there for a Turing machine with \( n \) states (and that have 3 characters \( \text{A, B, and # (the blank character)} \))? You can assume that the start state is always state 1, and 2 programs that are functionally equivalent are still considered different programs. (Hint, such a Turing machine will have 3\( n \) lines.)

For simplicity, follow the syntax below, and count these 2 lines as different:

\[
\text{In state and reading write move and goto state} \\
2 \# \text{A Right \texttt{halt}}
\]

and

\[
\text{In state and reading write move and goto state} \\
2 \# \text{A Left \texttt{halt}}
\]

I.e., the reader head must always have a \texttt{left} or \texttt{right} instruction and \texttt{halt} is a special state transition that causes the machine to halt. (If you have problems with a general equation for \( n \) state Turing machines, you may give an answer for a 1 state Turing machine for partial credit.) Also don’t worry about state reachability (that would be incomputable).

\textbf{Answer:} There are 3 lines for every state, which gives 3\( n \) lines. In each of these lines, what to \texttt{write}, \texttt{move}, and which \texttt{state} to goto can all vary. There are 3 possible symbols to write, 2 possible directions to move and \( n + 1 \) possible states to goto (including \texttt{halt}). This gives \( 3 \cdot 2 \cdot (n + 1) = 6(n + 1) \) combinations for each line. So there are \( (6(n + 1))^{3n} \) possible Turing machines with \( n \) states. So, there are \( (6(1 + 1))^3 = 1,738 \) single state Turing machines.
(8) (10 BONUS points total)

(a) (5 BONUS points) Briefly, what’s the relationship between Aunt Hillary and Johant Sebastiant Fermant. (These are the 2 ant hills that the Anteater discusses in Ant Fugue. Recall that Aunt Hillary is the younger ant hill, and Johant Sebastiant Ant is “deceased”)?

**Answer:** Aunt Hillary was originally made of the same ants that comprised JSF at the time of his demise.

(b) (5 BONUS points) How is an ant hill (like Aunt Hillary) like a traffic jam or a “glider” in Conway’s Game of Life?

**Answer:** Ant hills are different from a collection of ants (since ants may come and go) just as traffic jams are different from the cars that comprise them (since cars enter and leave the traffic jam), and gliders are separate from the automata that make them up (since no single automaton is part of the glider for more than a few clock cycles).