Name: *put your name here*

This reflection is to be completed individually, although consultations with TAs and classmates are encouraged as long as they are appropriately acknowledged. This document is intended to reinforce your understanding of recursion, memoization, trees, and C++.

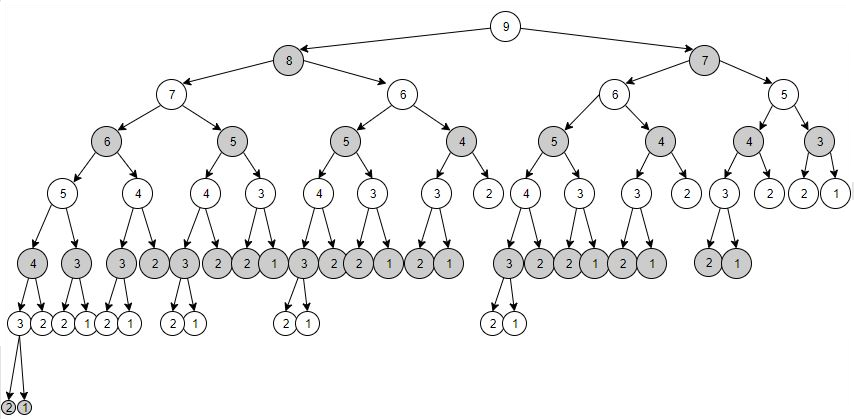
## The Pick-up Stones Game

The rules of the Pick-up Stones Game are as follows:

* There are a set number of stones in a pile: START\_STONES
* Players alternate taking at least 1 stone and at most MAX\_TAKE stones each turn.
* The player who take the last set of stones wins.

The obvious strategy is to try to leave your opponent with no stones.

**The Game Tree**  
A game tree is a tracing of all paths of a recursive search function that traces all possible moves of a strategy game, and the result of each in an attempt to find an optimal move. Game trees are used for decision-making in Artificial Intelligence, and are primarily useful in scenarios that do not require real-time decision-making or which have a relatively low number of possible choices at each decision point. Consider the following game tree which starts with START\_STONES = 9, and MAX\_TAKE = 2.



1. Considering the rules of the Pick-up Stones Game, why is the above game tree binary? Explain.

|  |
| --- |
|  |

1. The number of possible nodes doubles at each level in the above game tree. Explain why this is.

|  |
| --- |
|  |

1. What does each level in the above game tree represent in the Pick-up Stones game? Explain.

|  |
| --- |
|  |

1. In the above tree, why are all of the leaf nodes labelled either 1 or 2, yet additional nodes labelled with 1 could be drawn. Explain.

|  |
| --- |
|  |

1. If START\_STONES = 9, and MAX\_TAKE = 2 and a human plays first against a computer that plays second, use the above game tree to try to identify a winning strategy. i.e. a strategy in which you can force the computer to lose or show that this is not possible. Using only the above tree, describe the strategy and how you know it will work or how you know it is not possible.

|  |
| --- |
|  |

1. If START\_STONES = 8, and MAX\_TAKE = 2 and a human plays first against a computer that plays second, again use the above game tree to try to identify a winning strategy. (Hint: You will be looking only at a sub-tree to try to identify a strategy in which you can force the computer to lose or show that it is not possible.) Using only the above tree, describe the strategy and how you know it will work or how you know it is not possible.

|  |
| --- |
|  |

**Recursion and Memoization**

1. In the stonesmemo.cpp program, the recursive function human\_can\_win traverses the (abstract) game tree. Looking at the code, when memoize is not set, does it use in-order, pre-order, or post-order to visit the nodes of the game tree? Explain.

|  |
| --- |
|  |

**Memoization** is often used to speed up recursion by storing the results of function or method call based on the given inputs rather than simply computing the result over again. Then the function just returns the stored result rather than computing the result again. One can think of it as a cache for function or method results.

1. Try running stonesmemo.cpp with various settings with the goal of understanding the correspondence between the above game tree and the given program. The memoize setting is storing a Boolean value for a given integer in an array called can\_force\_win. In your own words, explain specifically what this Boolean represents on the above game tree.

|  |
| --- |
|  |

1. If memoization saves time by not repeatedly taking function or method calls which compute the same thing, explain this savings in terms of the game tree above. Specifically, how specifically can memoization savings be understood graphically on the game tree?

|  |
| --- |
|  |

1. Noting in the program that, the main function uses stone==0, if START\_STONES = 8, and   
   MAX\_TAKE = 2 and a human plays first against a computer that plays second, and the human makes a choice of taking 2 stones as the first move, use a section of the above game tree to explain why without memoization, the computer makes exactly 30 function calls, but with memoization makes exactly 6 calls.

|  |
| --- |
|  |

1. If START\_STONES = 21, and MAX\_TAKE = 3 and a human plays first against a computer that plays second, and the human makes a choice of taking 1 stone as the first move. With verbose off, try playing with and without memoization. How many function calls are reported in each case? Explain the huge discrepancy.

|  |
| --- |
|  |

**Big O**

If START\_STONES = n, and MAX\_TAKE = m, we can calculate the Big O of the Pick-up Stones algorithm both with memoization and without.

1. For the computer, the worse case of our Pick-up Stones algorithm without memoization occurs when the human chooses a path that might possibly lead the human winning. You can see an example of the difference which this makes if you START\_STONES = 21, and MAX\_TAKE = 3.   
   Try running without verbose and without memoization making the following choices: 1, 1, 1, 2, 2, 1.   
   Try running without verbose and without memoization making the following choices: 2, 2, 1, 1, 1, 1.  
   Write the numbers of function calls reported and explain.

|  |
| --- |
|  |

1. If START\_STONES = n, and MAX\_TAKE = m, calculate the Big O of this algorithm without memoization. Hint: Think about what paths are traced on the game tree and explain.

|  |
| --- |
|  |

1. If START\_STONES = n, and MAX\_TAKE = m, calculate the Big O of this algorithm with memoization. Hint: Think about what paths are traced on the game tree and explain.

|  |
| --- |
|  |

## Conversion of C++ to Python

Next, you need to convert the program from C++ to Python, using a kind of line-by-line conversion. Variable declarations which are not initializations can simply be converted to comments.

1. Give at least five important concepts that Python programmers should keep in mind when translating a C++ program to Python. Any explanations must be in your own words, so if you are quoting the text, you must explain why you think the concepts are important.

|  |
| --- |
|  |