

1. [1 marks] **Complete this page properly.**

Each group submits only one assignment. Print the name and ID number of each group member (at most 4) for this assignment:

Acknowledge **all** sources, including all references and all people not in your group with whom you discussed any part of any question (for each discussion, list the relevant questions) (continue on the back of this page if there is insufficient space). If none, write *webnotes and lectures only*.

Each group member must read, agree to, and sign this statement:

I am familiar with the Code of Student Behaviour. I understand that there are significant penalties for any infraction of this Code, including failure to acknowledge sources. I have not shared any written or printed version of any of my answers with any other student.

2. [7 marks] The function `foo()` from the quiz uses negamax to solve 3-pile nim.

- (a) Starting with `foo()`, and making as few changes as necessary, write a python function `calls(a,b,c)` that returns the number of recursive calls made when computing `foo(a,b,c)`. (This is the same as the number of nodes in the search space.) E.g. `calls(0,0,0)` returns 1, `calls(1,0,0)` returns 2, `calls(1,1,1)` returns 16. Using your function, complete the table below.

j	0	1	2	3	4
<code>calls(j,j,j)</code>	1	16			

- (b) Modify `foo()` by uncommenting the 3 comments. Also, insert `print(a,b,c)` as the first line of the function body. Now `(a,b,c)` is printed once every time a call is made. (If you prefer, you can modify `foo()` so that it returns both the value and the number of calls made, as in function `ab_neg()` in program `ttt_classic.py`.) Complete the following table.

j	0	1	2	3	4	5	6	7	8	9
calls made by modified <code>foo(j,j,j)</code>	1	6								

- (c) Assume that you modified the negamax version of `foo` from (a) to use alpha-beta (this would give you the alphabeta-negamax version of `foo`). Would the number of calls made be (**choose one**) (always fewer, usually fewer, always equal, usually more, always more) than the number of calls made by the modified `foo` from (b)? Explain briefly.
-
-

- (d) Modify function `solveall()` in program `nim.py` so that it prints the number of loss updates (each time execution reaches the comment "...loses, find all wins ...") and win updates (each time execution reaches the comment "`print(self.crd(pjc),'wins')`"), and complete the following table.

j	0	1	2	3	4	5	6	7	8	9
total loss/win updates made by <code>solveall nim(j,j,j)</code>	0	10								