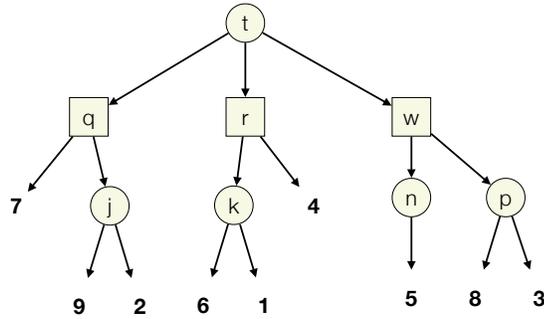


Unless otherwise indicated, assume that the root player is a MAX player and leaf nodes are labelled with the MAX player's score.

1. Consider the following state graph



(a) What are the minimax values for each nonterminal node?

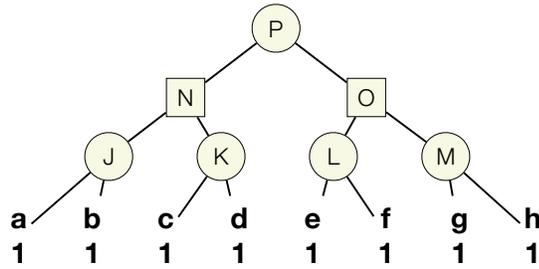
Node	j	k	n	p	q	r	w	t
minimax	9	6	5	8	7	4	5	7

(b) What are the negamax values for each nonterminal node? Assume that each terminal node is labelled with the score for the player-to-move.

Node	j	k	n	p	q	r	w	t
negamax	-2	-1	-5	-3	2	1	5	-1

(c) Why are these two values different? Because the scores on the first tree are all player 1 scores, whereas the scores on the second tree are all player 2 scores with the exception of 7 from q and 4 from r ; that is, all of the leaf nodes except for 7 and 4 have effectively been replaced with their own negations.

2. Suppose we execute alpha-beta search on the following tree.



- (a) If we use the $\alpha \geq \beta$ pruning rule, which leaf nodes will be evaluated? How many leaf nodes are pruned? **Leaf nodes $a, b, c, e, f,$ and g will be evaluated; 3 leaves ($d, g,$ and h) will be pruned.**
- (b) *After* alpha-beta completes using the $\alpha \geq \beta$ pruning rule, what is known about the minimax value for each nonterminal node? E.g., use ≥ 3 to indicate a lower bound of 3, ≤ 2 to indicate an upper bound of 2, 7 to indicate exactly 7, and ? to indicate that nothing is known.

Node	J	K	L	M	N	O	P
nmx	1	≥ 1	1	?	1	≤ 1	1

- (c) If we use the $\alpha > \beta$ pruning rule, which leaf nodes will be evaluated? How many leaf nodes are pruned? **All leaf nodes will be evaluated; 0 will be pruned.**
- (d) *After* alpha-beta completes using the $\alpha > \beta$ pruning rule, what is known about the minimax value for each nonterminal node? **Every non-terminal node will be known to have a minimax value of exactly 1.**

3. Suppose we execute alpha-beta search on the following two trees using the $\alpha \geq \beta$ pruning rule.



- (a) Which leaf nodes will be evaluated in the left tree? How many nodes are pruned? Leaf nodes *a*, *b*, *c*, *e*, *f*, and *g* will be evaluated; 2 leaves (*d* and *h*) will be pruned.
- (b) After alpha-beta completes in the left tree, what is known about the minimax value for each nonterminal node?

Node	J	K	L	M	N	O	P
nmx	2	≥ 3	6	≥ 7	2	6	6

- (c) Which leaf nodes will be evaluated in the right tree? How many nodes are pruned? Leaf nodes *a*, *b*, *c*, *e*, and *f* will be evaluated; 3 leaves (*d*, *g*, and *h*) will be pruned.
- (d) After alpha-beta completes in the right tree, what is known about the minimax value for each nonterminal node?

Node	J	K	L	M	N	O	P
nmx	6	≥ 8	2	?	6	≤ 2	6

4. Consider the follow code from abeta/negamax.py, with three missing expressions:

```
def negamax(d, T, V, v): # leaf scores for player-to-move
    if isTerminalNode(v,V):
        val = V[v]
        return val
    val = _____expression_(A)_____
    for c in T[v]: # for each child c of v
        nmx = negamax(d+1, T, V, c)
        val = max(_____expression_(B)_____, _____expression_(C)_____)
    return val
```

- (a) What should expression (A) be?
-inf or -float('inf') or NEGINF or anything that indicates negative infinity.
- (b) What should expression (B) and expression (C) be?
-nmx and val (in either order)

5. Consider the following code in `tt.py`:

```
Isos = ( (0,1,2,3,4,5,6,7,8),  
        (0,3,6,1,4,7,2,5,8),  
        (2,1,0,5,4,3,8,7,6),  
        (2,5,8,1,4,7,0,3,6),  
        (8,7,6,5,4,3,2,1,0),  
        (8,5,2,7,4,1,6,3,0),  
        (6,7,8,3,4,5,0,1,2),  
        (6,3,0,7,4,1,8,5,2) )
```

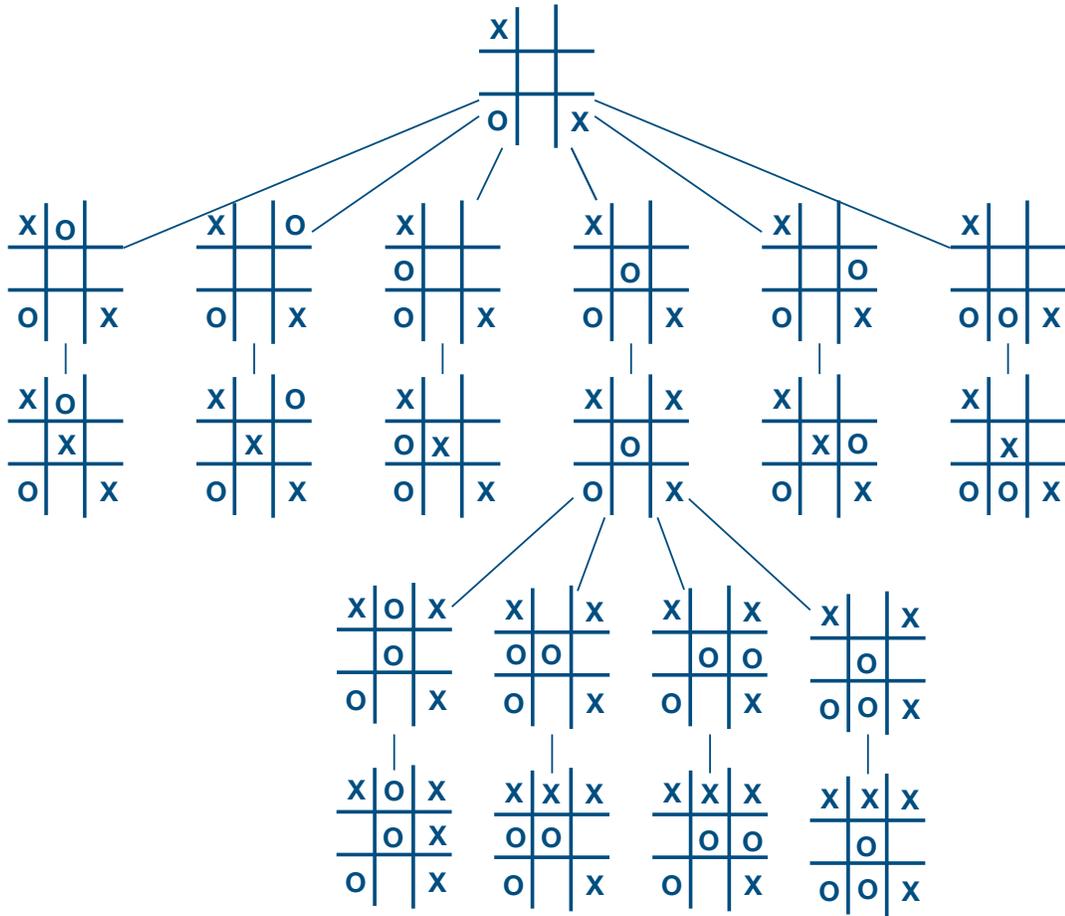
(a) What is the purpose of this code?

Each element of `Isos` is a permutation of the representation of a tic-tac-toe position that represents an isomorphic transformation of that position; each element contains an index of the original position that indicates the element of the original position that should be placed there. So, for example, the first permutation is the identity transformation; the second permutation flips the position horizontally; etc. By applying each transformation to a position, we get a position that is isomorphic to the original. These are all of the isomorphic transformations, so if we apply them all, we will have the set of all positions that are isomorphic to the original position (including the original position itself).

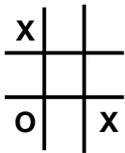
(b) What is the result of applying the last transformation in `Isos` to the position $(1,0,0,2,1,0,0,0,0)$? $(0,2,1,0,1,0,0,0,0)$

6. Give a proof tree that proves that the following position is a win for X.

x		
o		x



7. What are the non-isomorphic children of the following position? Use the canonical representation from lecture that chooses the smallest numerical representation of a position, where empty cells map to 0, X cells map to 1, and O cells map to 2.



The canonical forms of each non-isomorphic child are:



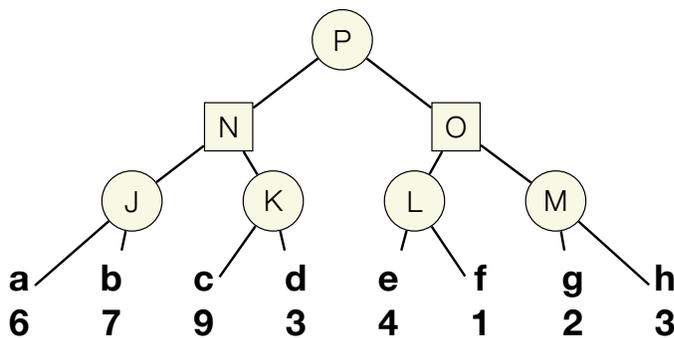
For part marks, these are the non-isomorphic actual children of the position, where I have chosen the children that are earliest in the canonical representation:



8. Consider the Nim position $(7,7,7,7,7,7,7)$:

- (a) How many children does this position have? *There is no need to list the children.* $7 \times 7 = 49$: We can choose from 7 different piles, and within each pile we can take 7 different numbers of stones (i.e., 1–7).
- (b) How many *non-isomorphic* children does this position have? List them all in canonical form. Canonical form sorts the piles by size; I chose to sort in ascending order, but descending order is also fine for full marks. A move means removing some stones from one of the piles, but the pile that is chosen will always get sorted to the beginning of the order, so there's only 7 non-isomorphic children: $(6,7,7,7,7,7,7)$, $(5,7,7,7,7,7,7)$, $(4,7,7,7,7,7,7)$, $(3,7,7,7,7,7,7)$, $(2,7,7,7,7,7,7)$, $(1,7,7,7,7,7,7)$, $(0,7,7,7,7,7,7)$.

9. This alpha-beta search has just reached K . In the table below, list the new values of **so_far**, **alpha**, and **beta** in order, every time at least one of them changes. We've listed the changes up to but not including node K .



chg	node	so_far	alpha	beta
1	P	$-\infty$	$-\infty$	∞
2	N	∞	$-\infty$	∞
3	J	$-\infty$	$-\infty$	∞
4	J	6	6	∞
5	J	7	7	∞
6	N	7	$-\infty$	7
7	K	$-\infty$	$-\infty$	7
8	K	9	9	7
9	P	7	7	∞
10	O	∞	7	∞
11	L	$-\infty$	7	∞
12	L	4	7	∞
13	O	4	7	4