

0. On page 0, in the bubbles, write your ***** CCID *****.

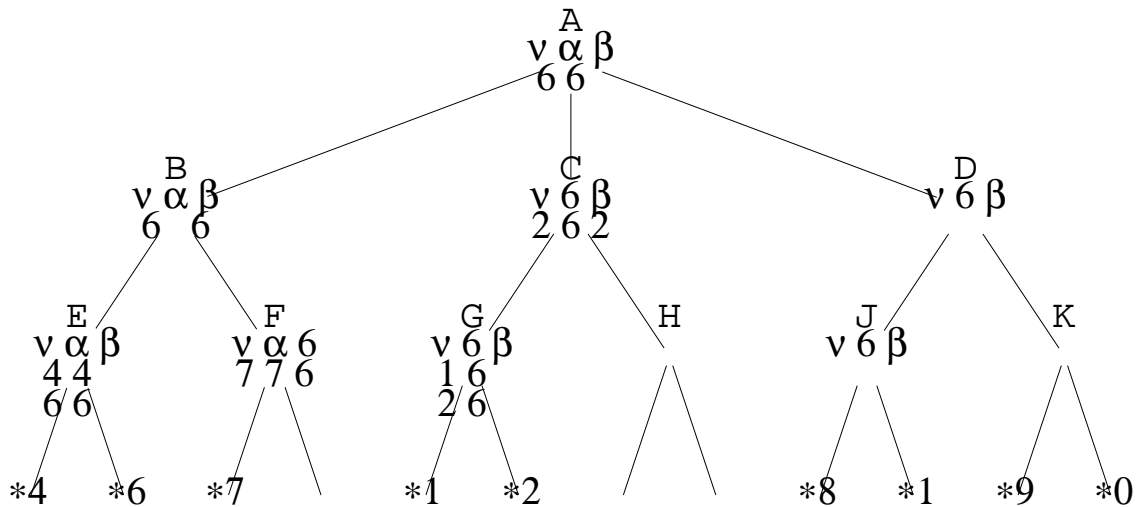
On pages 0, 1, 2, 3, write your first name, last name and student id.

1. [4 marks] At right, unscramble these lines from the start of the alphabeta function in `alphabeta.py`. Write line numbers only: indent properly. We have written the first line number for you.

```

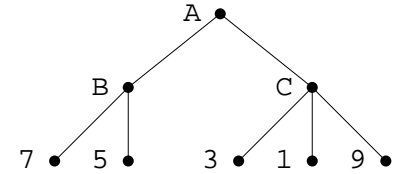
ab = alphabeta(d+1, T, V, c, alpha, beta) #1      (3)  ---  ---  ---  ---  ---
break                                         #2      ---  ---  ---  ---  ---
def alphabeta(d, T, V, v, alpha, beta):      #3      ---  ---  ---  ---  ---
for c in T[v]:                               #4      ---  ---  ---  ---  ---
if ab > val: alpha, val = ab, ab            #5      ---  ---  ---  ---  ---
if alpha >= beta:                           #6      ---  ---  ---  ---  ---
if isMaxNode(v, d):                         #7      ---  ---  ---  ---  ---
if isTerminalNode(v,V): return V[v]        #8      ---  ---  ---  ---  ---
return val                                   #9      ---  ---  ---  ---  ---
val = NEGINF                                #10     ---  ---  ---  ---  ---
    
```

2. [4 marks] This diagram shows an alphabeta search that has just reached node J. At each node, the bottom line shows current minimax, alpha, beta values. In order, in the rest of the search, show changes made at each non-leaf node. There might be fewer than 8 changes. We have shown you change 1.



change	node	minimax	alpha	beta	change	node	minimax	alpha	beta
1	J	8	8	-	5	---	---	---	---
2	---	---	---	---	6	---	---	---	---
3	---	---	---	---	7	---	---	---	---
4	---	---	---	---	8	---	---	---	---

3. [2 marks] a) For this game tree, for each node, give the minimax value.
 (MAX plays first, values are for MAX.)



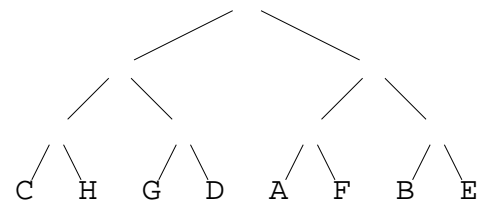
- b) Repeat a) if MIN plays first.
 (Values are still for MAX: do not change any leaf values).

a) A ___ B ___ C ___ D ___ E ___ F ___ G ___ H ___ I ___ J ___ K ___

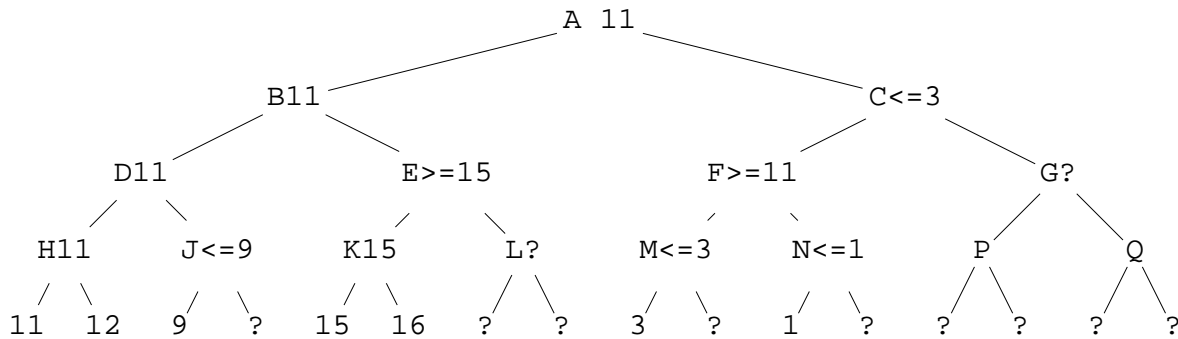
b) A ___ B ___ C ___ D ___ E ___ F ___ G ___ H ___ I ___ J ___ K ___

4. [2 marks] All leaf nodes have value 1. Which leaf nodes are reached in an alphabeta search with cutoff test $\alpha \geq \beta$?

answer: _____



5. [4 marks] In this alphabeta search, some subtrees are cut off. Explain the reason for each cut by filling in the blanks. We have given the answer for node J.



- J: on path-to-root J-D-B-A, MAX has move option D-H so alpha 11,
 MIN has move option J-9 so beta 9, so alpha \geq beta
- E: on path-to-root _____, MAX has move option _____ so alpha ____
 MIN has move option _____ so beta _____ so alpha \geq beta
- N: on path-to-root _____, MAX has move option _____ so alpha ____
 MIN has move option _____ so beta _____ so alpha \geq beta
- C: on path-to-root _____, MAX has move option _____ so alpha ____
 MIN has move option _____ so beta _____ so alpha \geq beta

6. [4 marks] For this tic-tac-toe position with **x** to play, draw a proof tree that shows that **o** can win: draw all **x**-moves, then for each a winning **o**-reply, then all **x**-moves, then for each a winning **o**-reply, and so on. Use the position as the root of your tree.

```

      . o .
      x o x
      . . .

```

7. [4 marks] Modify the win condition for tic-tac-toe: 1st-player **x** wins with 3-in-a-row, otherwise 2nd player **o** wins. (It doesn't matter whether **o** gets 3-in-a-row.) For this position, give the minimax value: **x-win or o-win**). Below the position, draw the next two levels of a proof tree that justifies your answer. Label every position in the bottom level of your diagram with its minimax value.

```

o . .   minimax
. x .   value  _____
. . .

```

0. On page 0, in the bubbles, write your ***** CCID *****.

On pages 0, 1, 2, 3, write your first name, last name and student id.

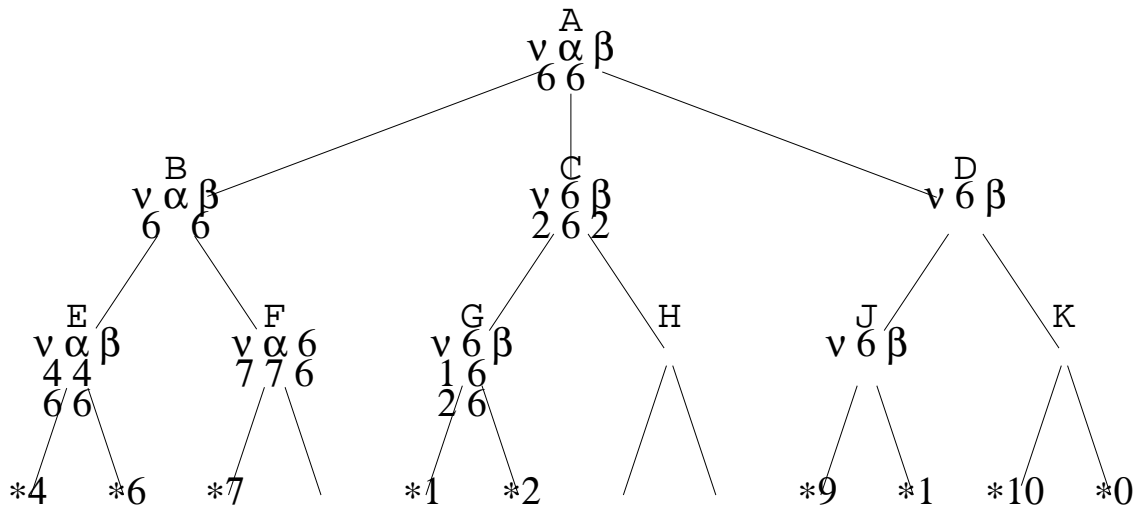
1. [4 marks] At right, unscramble these lines from the start of the alphabeta function in `alphabeta.py`. Write line numbers only: indent properly. We have written the first line number for you.

```

if isMaxNode(v, d):                #1    (9)  --- --- --- ---
if isTerminalNode(v,V): return V[v] #2    --- --- --- ---
return val                          #3    --- --- --- ---
val = NEGINF                        #4    --- --- --- ---
if ab > val: alpha, val = ab, ab    #5    --- --- --- ---
if alpha >= beta:                   #6    --- --- --- ---
ab = alphabeta(d+1, T, V, c, alpha, beta) #7    --- --- --- ---
break                                #8    --- --- --- ---
def alphabeta(d, T, V, v, alpha, beta): #9    --- --- --- ---
for c in T[v]:                       #10   --- --- --- ---

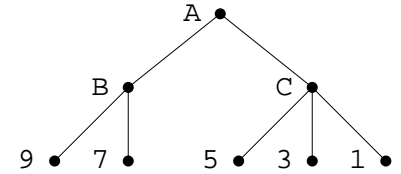
```

2. [4 marks] This diagram shows an alphabeta search that has just reached node J. At each node, the bottom line shows current minimax, alpha, beta values. In order, in the rest of the search, show changes made at each non-leaf node. There might be fewer than 8 changes. We have shown you change 1.



change	node	minimax	alpha	beta	change	node	minimax	alpha	beta
1	J	9	9	-	5	---	---	---	---
2	---	---	---	---	6	---	---	---	---
3	---	---	---	---	7	---	---	---	---
4	---	---	---	---	8	---	---	---	---

3. [2 marks] a) For this game tree, for each node, give the minimax value.
 (MAX plays first, values are for MAX.)



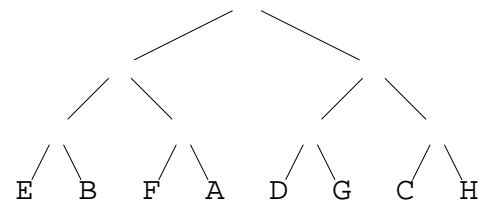
b) Repeat a) if MIN plays first.
 (Values are still for MAX: do not change any leaf values).

a) A ___ B ___ C ___ D ___ E ___ F ___ G ___ H ___ I ___ J ___ K ___

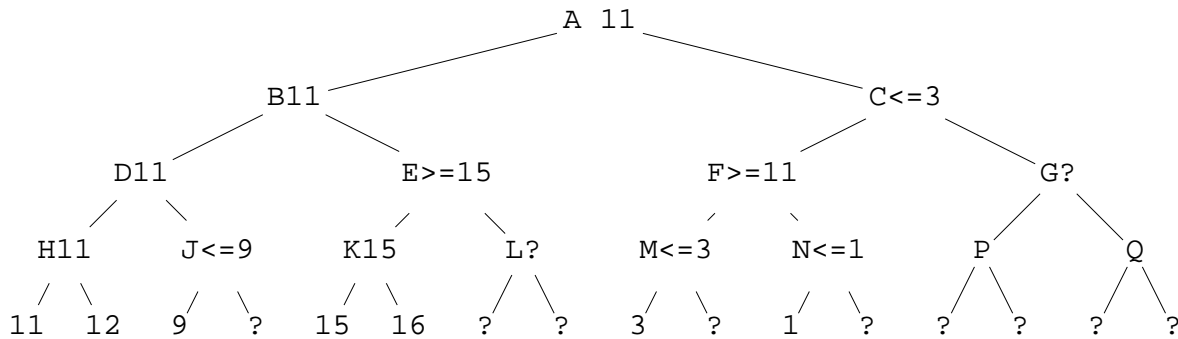
b) A ___ B ___ C ___ D ___ E ___ F ___ G ___ H ___ I ___ J ___ K ___

4. [2 marks] All leaf nodes have value 1. Which leaf nodes are reached in an alphabeta search with cutoff test $\alpha \geq \beta$?

answer: _____



5. [4 marks] In this alphabeta search, some subtrees are cut off. Explain the reason for each cut by filling in the blanks. We have given the answer for node J.



J: on path-to-root J-D-B-A, MAX has move option D-H so alpha 11,
 MIN has move option J-9 so beta 9, so alpha >= beta

C: on path-to-root _____, MAX has move option _____ so alpha ____
 MIN has move option _____ so beta _____ so alpha >= beta

E: on path-to-root _____, MAX has move option _____ so alpha ____
 MIN has move option _____ so beta _____ so alpha >= beta

N: on path-to-root _____, MAX has move option _____ so alpha ____
 MIN has move option _____ so beta _____ so alpha >= beta

6. [4 marks] For this tic-tac-toe position with **x** to play, draw a proof tree that shows that **o** can win: draw all **x**-moves, then for each a winning **o**-reply, then all **x**-moves, then for each a winning **o**-reply, and so on. Use the position as the root of your tree.

```

. x .
o o .
. x .

```

7. [4 marks] Modify the win condition for tic-tac-toe: 1st-player **x** wins with 3-in-a-row, otherwise 2nd player **o** wins. (It doesn't matter whether **o** gets 3-in-a-row.) For this position, give the minimax value: **x-win or o-win**). Below the position, draw the next two levels of a proof tree that justifies your answer. Label every position in the bottom level of your diagram with its minimax value.

```

. x .   minimax
. o .   value _____
. . .

```

0. On page 0, in the bubbles, write your ***** CCID ***** .

On pages 0, 1, 2, 3, write your first name, last name and student id.

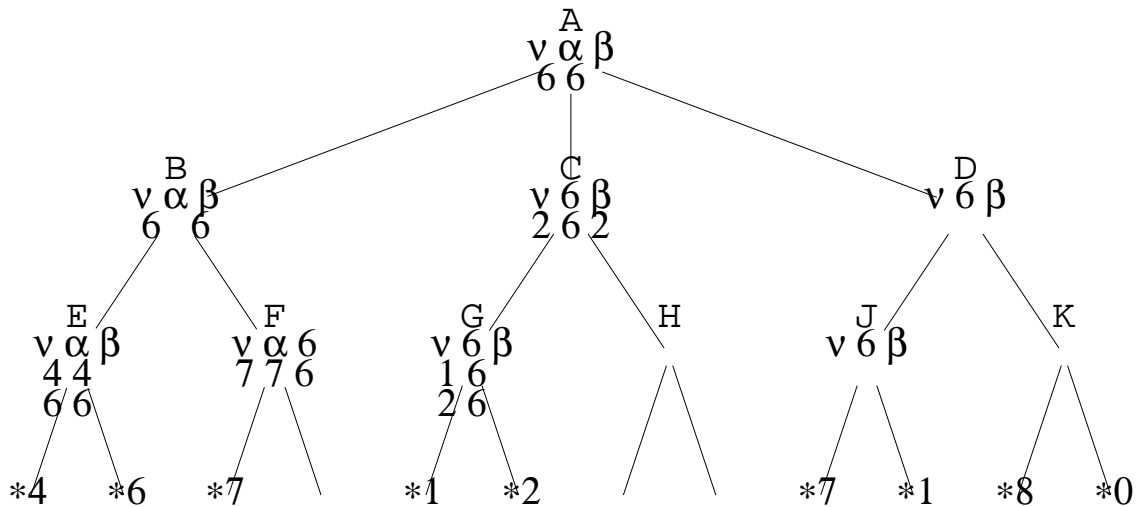
1. [4 marks] At right, unscramble these lines from the start of the alphabeta function in `alphabeta.py`. Write line numbers only: indent properly. We have written the first line number for you.

```

if ab > val: alpha, val = ab, ab      #1      (5)  --- --- --- ---
for c in T[v]:                        #2      --- --- --- ---
break                                  #3      --- --- --- ---
ab = alphabeta(d+1, T, V, c, alpha, beta) #4      --- --- --- ---
def alphabeta(d, T, V, v, alpha, beta): #5      --- --- --- ---
val = NEGINF                           #6      --- --- --- ---
if alpha >= beta:                       #7      --- --- --- ---
return val                              #8      --- --- --- ---
if isTerminalNode(v,V): return V[v]     #9      --- --- --- ---
if isMaxNode(v, d):                    #10     --- --- --- ---

```

2. [4 marks] This diagram shows an alphabeta search that has just reached node J. At each node, the bottom line shows current minimax, alpha, beta values. In order, in the rest of the search, show changes made at each non-leaf node. There might be fewer than 8 changes. We have shown you change 1.



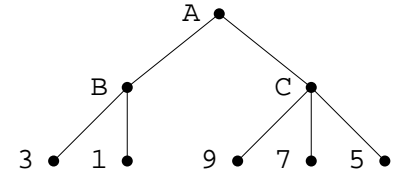
change	node	minimax	alpha	beta	change	node	minimax	alpha	beta
1	J	7	7	-	5	---	---	---	---
2	---	---	---	---	6	---	---	---	---
3	---	---	---	---	7	---	---	---	---
4	---	---	---	---	8	---	---	---	---

3. [2 marks] a) For this game tree, for each node, give the minimax value.

(MAX plays first, values are for MAX.)

b) Repeat a) if MIN plays first.

(Values are still for MAX: do not change any leaf values).



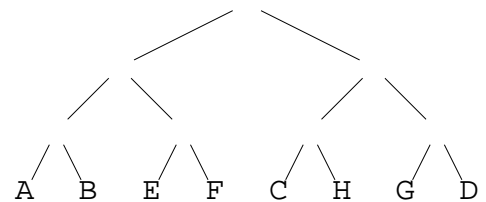
a) A ___ B ___ C ___ D ___ E ___ F ___ G ___ H ___ I ___ J ___ K ___

b) A ___ B ___ C ___ D ___ E ___ F ___ G ___ H ___ I ___ J ___ K ___

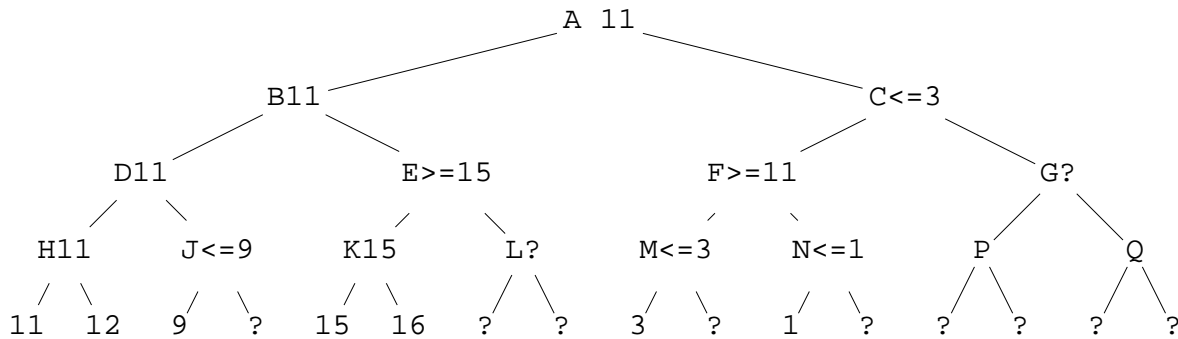
4. [2 marks] All leaf nodes have value 1. Which leaf nodes are reached

in an alphabeta search with cutoff test $\alpha \geq \beta$?

answer: _____



5. [4 marks] In this alphabeta search, some subtrees are cut off. Explain the reason for each cut by filling in the blanks. We have given the answer for node J.



J: on path-to-root J-D-B-A, MAX has move option D-H so alpha 11,

MIN has move option J-9 so beta 9, so alpha >= beta

N: on path-to-root _____, MAX has move option _____ so alpha ____

MIN has move option _____ so beta _____ so alpha >= beta

C: on path-to-root _____, MAX has move option _____ so alpha ____

MIN has move option _____ so beta _____ so alpha >= beta

E: on path-to-root _____, MAX has move option _____ so alpha ____

MIN has move option _____ so beta _____ so alpha >= beta

6. [4 marks] For this tic-tac-toe position with **x** to play, draw a proof tree that shows that **o** can win: draw all **x**-moves, then for each a winning **o**-reply, then all **x**-moves, then for each a winning **o**-reply, and so on. Use the position as the root of your tree.

```

      . x .
      . o o
      . x .
  
```

7. [4 marks] Modify the win condition for tic-tac-toe: 1st-player **o** wins with 3-in-a-row, otherwise 2nd player **x** wins. (It doesn't matter whether **x** gets 3-in-a-row.) For this position, give the minimax value: **x-win or o-win**). Below the position, draw the next two levels of a proof tree that justifies your answer. Label every position in the bottom level of your diagram with its minimax value.

```

o . .   minimax
. x .   value _____
. . .
  
```