

### Case1

Input:

case = "((4 (7 9 8) 8) (((3 6 4) 2 6) ((9 2 9) 4 7 (6 4 5))))"

Output:

MINIMAX:

maximum value : 6

path : [2, 1, 3]

ALPHA-BETA:

max cut after 7 in subtree ( 7 9 8 )

min cut after 3 in subtree ( 3 6 4 )

min cut after 2 in subtree ( 9 2 9 )

max cut after 7 in subtree ( ( 9 2 9 ) 4 7 ( 6 4 5 ) )

maximum value : 6

### Case2

Input:

case = "(((1 4) (3 (5 2 8 0) 7 (5 7 1)) (8 3)) (((3 6 4) 2 (9 3 0)) ((8 1 9) 8 (3 4 ))))"

Output:

MINIMAX:

maximum value : 4

path : [1, 1, 2]

ALPHA-BETA:

min cut after 2 in subtree ( 5 2 8 0 )

max cut after 7 in subtree ( 3 ( 5 2 8 0 ) 7 ( 5 7 1 ) )

max cut after 8 in subtree ( 8 3 )

min cut after 3 in subtree ( 3 6 4 )

min cut after 3 in subtree ( 9 3 0 )

min cut after ( ( 3 6 4 ) 2 ( 9 3 0 ) ) in subtree ( ( ( 3 6 4 ) 2 ( 9 3 0 ) ) ( ( 8 1 9 ) 8 ( 3 4 ) ) )

maximum value : 4

### Case3

Input:

case = "(5 (((4 7 -2) 7) 6))"

Output:

MINIMAX:

maximum value : 6

path : [2, 2]

ALPHA-BETA:

min cut after 4 in subtree ( 4 7 -2 )

maximum value : 6

### Case4

Input:

case = "((8 (7 9 8) 4) (((3 6 4) 2 1) ((6 2 9) 4 7 (6 4 5))))"

Output:

MINIMAX:

maximum value : 4

path : [1, 3]

ALPHA-BETA:

max cut after 9 in subtree ( 7 9 8 )

min cut after 3 in subtree ( 3 6 4 )

min cut after ( ( 3 6 4 ) 2 1 ) in subtree ( ( ( 3 6 4 ) 2 1 ) ( ( 6 2 9 ) 4 7 ( 6 4 5 ) ) )

maximum value : 4

#### Case5

Input:

case = "(((1(4 7)) (3 ((5 2) (2 8 9) 0 -2) 7 (5 7 1)) (8 3)) (((8 (9 3 2) 5) 2 (9 (3 2) 0)) ((3 1 9) 8 (3 4))))"

Output:

MINIMAX:

maximum value : 5

path : [2, 1, 1, 3]

ALPHA-BETA:

max cut after 5 in subtree ( 5 2 )

max cut after 8 in subtree ( 2 8 9 )

min cut after 0 in subtree ( ( 5 2 ) ( 2 8 9 ) 0 -2 )

max cut after 7 in subtree ( 3 ( ( 5 2 ) ( 2 8 9 ) 0 -2 ) 7 ( 5 7 1 ) )

max cut after 8 in subtree ( 8 3 )

max cut after 9 in subtree ( 9 3 2 )

min cut after ( 3 2 ) in subtree ( 9 ( 3 2 ) 0 )

min cut after 3 in subtree ( 3 1 9 )

max cut after 8 in subtree ( ( 3 1 9 ) 8 ( 3 4 ) )

maximum value : 5