```
1: BOARD_SIZE = 8
2: def under_attack(col, queens):
3:     left = right = col
4:     for r, c in reversed(queens):
5:         left, right = left - 1, right + 1
6:         if c in (left, col, right):
7:             return True
8:     return False
9: def solve(n):
10:     if n == 0:
11:         return [[]]
12:     smaller_solutions = solve(n - 1)
13:     return [solution+[(n,i+1)]
14:         for i in xrange(BOARD_SIZE)
15:             for solution in smaller_solutions
16:                 if not under_attack(i+1, solution)]
17: for answer in solve(BOARD_SIZE):
18:     print answer
```

```
  1           0 LOAD_CONST               0 (8)
              3 STORE_NAME               0 (BOARD_SIZE)
  3           6 LOAD_CONST               1 ()
              9 MAKE_FUNCTION            0
             12 STORE_NAME               1 (under_attack)
 13          15 LOAD_CONST               2 ()
             18 MAKE_FUNCTION            0
             21 STORE_NAME               2 (solve)
 23          24 SETUP_LOOP              25 (to 52)
             27 LOAD_NAME                2 (solve)
             30 LOAD_NAME                0 (BOARD_SIZE)
             33 CALL_FUNCTION            1
             36 GET_ITER
        >>   37 FOR_ITER                11 (to 51)
             40 STORE_NAME               3 (answer)
 24          43 LOAD_NAME                3 (answer)
             46 PRINT_ITEM
             47 PRINT_NEWLINE
             48 JUMP_ABSOLUTE           37
        >>   51 POP_BLOCK
        >>   52 LOAD_CONST               3 (None)
             55 RETURN_VALUE
```