Final Exam Question 1
netID: joshcoop

Josh Cooper

Question 1)

C(0,0) = max([-1 + 0.9* C(1,0) + 0.1*C(0,1)], //action = Go Down

[-1 + 0.9*C(0,1) + 0.1*C(1,0)]) //action = Go Right

Question 2)

C(1,0) = max([-1 + 0.9* C(2,0) + 0.1*C(1,1)], //action = Go Down

[-1 + 0.8* C(1,1) + 0.1*C(0,1) + 0.1*C(2,0)], //action = Go Right

[-1 + 0.9* C(0,0) + 0.1*C(1,1)]) //action = Go Up

Since 2,0 is a ravine, C(2,0) = -1000, so rewrite the above equations as:

C(1,0) = max([-1 + 0.9* (-1000) + 0.1*C(1,1)], //action = Go Down

[-1 + 0.8* C(1,1) + 0.1*C(0,1) + 0.1*(-1000)], //action = Go Right

[-1 + 0.9* C(0,0) + 0.1*C(1,1)]) //action = Go Up

Question 3)

General Equation for C(x,y) will involve a few things.

1) C(x,y) will be the expected cost for a specific cell (x,y)
2) C(next) will be the expected cost of the cell you are taking an action to move into (assuming no slip)
3) C(ortho) will be the expected cost of a cell which is orthogonal to the chosen action direction sending you to C(next), indicating a cell you could slip into
4) This max equation will have as many distinct terms (a single term is encapsulated by the brackets) as there are valid neighbors of C(x,y)

Case 1:

If (x,y) is a location in the middle of the map, it would have 4 available actions, and each action would have 2 orthogonal slipping locations:

C(x,y) = max([-1 + (0.8) * C(next) + (0.1) * C(ortho_1) + (0.1) * C(ortho_2)],

for all four C(next) which are valid neighbors of C(x,y))

Case 2:

If (x,y) is a location on a wall, it would have 3 available actions, with two actions having one orthogonal slip and one action having two orthogonal slips:

C(x,y) = max([-1 + (0.8) * C(next) + 0.1 * C(ortho_1) + 0.1 * C(ortho_2)], //move away from wall

[-1 + (0.9) * C(next) + 0.1 * C(ortho)] //move parallel to wall

for the two C(next) which are on the same wall as C(x,y))

Case 3:

If (x,y) is a location on a corner of the map, it would have 2 available actions with each action having one orthogonal slip:

C(x,y) = max([-1 + (0.9) * C(next) + 0.1 * C(ortho)]

for the two C(next) which are valid neighbors of C(x,y))

Question 4)

If you knew the values of C(x,y) for all spots on the board, the best direction for moving from C(0,0) would be the choice of direction which has the smallest absolute value as your choice of action from C(0,0).

Example:

C(0,0) = max([-1 + 0.9* C(1,0) + 0.1*C(0,1)], //action = Go Down

[-1 + 0.9*C(0,1) + 0.1*C(1,0)]) //action = Go Right

Assume C(1,0) = -20 and C(0,1) = -19 for example. Then we could say the value of C(0,0) would be:

C(0,0) = max([-1 + 0.9* (-20) + 0.1*(-19)], //action = Go Down

[-1 + 0.9*(-19) + 0.1*(-20)]) //action = Go Right

C(0,0) = max(-20.9, //action = Go Down

-20.1) //action = Go Right

C(0,0) would have an optimal policy saying it should **go right** due to the lower absolute value attached to that directional calculation.

Question 5)

The value of C(0,0) computed by my code was **-28.453082** when the ravine cost was **-1000**

Question 6)

The optimal actions for each cell on the board would look like this when the ravine cost is -1000:

	0	1	2	3	4
0	RIGHT	RIGHT	DOWN	DOWN	DOWN
1	UP	RIGHT	DOWN	DOWN	DOWN
2	hole	RIGHT	DOWN	DOWN	DOWN
3	hole	RIGHT	DOWN	DOWN	DOWN
4	hole	RIGHT	DOWN	DOWN	LEFT
5	hole	RIGHT	DOWN	LEFT	UP
6	hole	RIGHT	DOWN	LEFT	hole
7	DOWN	DOWN	DOWN	LEFT	hole
8	RIGHT	DOWN	DOWN	LEFT	hole
9	UP	RIGHT	DOWN	LEFT	hole
10	hole	RIGHT	DOWN	LEFT	hole
11	hole	RIGHT	DOWN	DOWN	DOWN
12	hole	RIGHT	DOWN	DOWN	DOWN
13	hole	RIGHT	DOWN	DOWN	DOWN
14	hole	RIGHT	DOWN	DOWN	DOWN
15	DOWN	DOWN	DOWN	DOWN	DOWN
16	goal!	LEFT	LEFT	LEFT	LEFT

Bonus Question)

By using **binary search** to identify a decent approximation of the tipping point between diving straight into the ravine and skirting around the side, I found that when the cost of jumping into the ravine is above **-25.912487219294565**, the optimal policy on the board is to move two steps down from the start and jump right into the ravine.

When the ravine cost switches from that value to anything above ~-46, the optimal policy becomes a path which bounces near the ravine, but sometimes moves away from it. Only after the ravine cost is below -46 does the optimal policy change to the picture shown in part 6.

```
     New Ravine Cost!
        0      1     2     3     4
0    DOWN   DOWN  DOWN  DOWN  DOWN
1   RIGHT   DOWN  DOWN  DOWN  DOWN
2    hole   DOWN  DOWN  DOWN  DOWN
3    hole   DOWN  DOWN  DOWN  DOWN
4    hole  RIGHT  DOWN  DOWN  LEFT
5    hole  RIGHT  DOWN  DOWN  LEFT
6    hole   DOWN  DOWN  LEFT  hole
7    DOWN   DOWN  DOWN  LEFT  hole
8   RIGHT   DOWN  DOWN  LEFT  hole
9   RIGHT   DOWN  DOWN  LEFT  hole
10   hole  RIGHT  DOWN  LEFT  hole
11   hole  RIGHT  DOWN  DOWN  DOWN
12   hole  RIGHT  DOWN  DOWN  DOWN
13   hole  RIGHT  DOWN  DOWN  DOWN
14   hole   DOWN  DOWN  DOWN  DOWN
15   DOWN   DOWN  DOWN  DOWN  DOWN
16  goal!   LEFT  LEFT  LEFT  LEFT
**For a ravine cost of -25.912487219294565,
are you dead in a ditch: False**

     New Ravine Cost!
>>>Dive into that ravine kiddo, it's cheaper
to die!<<<
        0      1     2     3     4
0    DOWN   DOWN  DOWN  DOWN  DOWN
1    DOWN   DOWN  DOWN  DOWN  DOWN
2    hole   DOWN  DOWN  DOWN  DOWN
3    hole   DOWN  DOWN  DOWN  DOWN
4    hole  RIGHT  DOWN  DOWN  LEFT
5    hole  RIGHT  DOWN  DOWN  LEFT
6    hole   DOWN  DOWN  LEFT  hole
7    DOWN   DOWN  DOWN  LEFT  hole
8   RIGHT   DOWN  DOWN  LEFT  hole
9   RIGHT   DOWN  DOWN  LEFT  hole
10   hole  RIGHT  DOWN  LEFT  hole
11   hole  RIGHT  DOWN  DOWN  DOWN
12   hole  RIGHT  DOWN  DOWN  DOWN
13   hole  RIGHT  DOWN  DOWN  DOWN
14   hole   DOWN  DOWN  DOWN  DOWN
15   DOWN   DOWN  DOWN  DOWN  DOWN
16  goal!   LEFT  LEFT  LEFT  LEFT
**For a ravine cost of -25.91248721929456,
are you dead in a ditch: True**
```