Erikson Sodergren
Intro to IS: Homework 2

1.
    a. Hill climbing
    b. Breadth first search?
    c. First-choice hill climbing
    d. Purely random walk
    e. Purely random walk

2.

1	Myth -> Immortal	given
2	!Myth -> (!Immortal ^ Mammal)	given
3	(Immortal V Mammal) ->Horn	given
4	Horn -> Magic	given
5	!Myth V Immortal	KB1,implication elimination
6	Myth V (!Immortal ^ Mammal)	KB2,implication elimination
7	!(Immortal V Mammal) V Horn	KB3,implication elimination
8	!Horn V Magic	KB4,implication elimination

a.

1	!Myth	Proof by contradiction
2	!Immortal ^ Mammal	1, KB6
3	Horn	2, KB7
4	Magic	3, KB8
5	No further steps possible.	Cannot prove

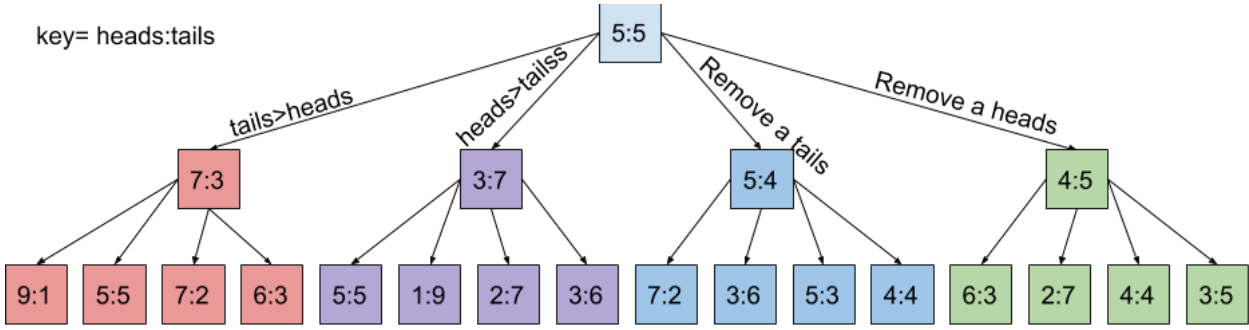
b.

1	!Magic	Proof by contradiction
2	!Horn	1, KB8
3	!(Immortal V Mammal)	2, KB7
4	!Immortal ^ !Mammal	3, De morgan
5	!Myth	4, KB5

6	!Immortal ^ Mammal	5, KB6
7	!Mammal ^ Mammal	4, 6

c.

1	!Horn	Proof by contradiction
2	!(Immortal V Mammal)	1, KB7
3	!Immortal ^ !Mammal	2, de morgan
4	!Myth	3, KB5
5	Myth	3, KB6

3. State consist of ingredients, each with an integer representing how much of it is in that recipe. Fitness function is how delicious the cookie is (impossible for a program to check currently, as such i will assume this is being done manually by cooking and tasting each recipe). A hill climbing algorithm would likely get caught on the large amount of plateaus or ridges you would get from recipes with very similar tastes due to small variation in ingredient quantities but would certainly improve with each variation,while the genetic algorithm would likely find the absolute most delicious cookie, but would require tasting many awful ones throughout the process, possibly out of nowhere due to mutation.
4. For simplicity sake, i will represent this game not as 10 different binary objects, but as 2 counts. One for the number of heads, 1 for the number of tails. Therefore the actions you can take are: subtract 1 from a count, or subtract 2 from 1 count and add 2 to the other.
   a. 

   b. Maximum branching factor is 4
   c. This tree has no maximum depth, as the players could choose to continue flipping coins each turn and never remove any.
   d. func(node) = [8, 0, 5, 3], [0, -8, -5, -3], [5, -3, 2, 0], [3, -5, 0, -2]
   e. We would prune the 3 rightmost leaves of the “max flips 2 heads to tails” subtree, the 2 rightmost leaves in the “max removes a tails” subtree, and the 2 rightmost leaves of the “max removes a heads” subtree
5. Seeding the python random module with a value of 15 creates a problem with Number Set: [4, 1, 9, 1, 3, 4, 1, 1, 3, 6, 4, 2, 6, 8, 6, 5, 7, 5, 6, 4, 4, 6, 6, 4, 5, 9, 7, 4, 8, 7, 8, 2, 8, 6, 8, 6, 8, 7, 2, 8, 1, 4, 3, 9, 3, 1, 5, 9, 8, 3, 2, 2, 7, 1, 2, 8, 9, 5, 7, 9, 4, 3, 9, 9, 6, 2, 4, 5, 5, 3, 6, 7, 8, 6, 8, 2, 5, 8, 3, 2, 8, 7, 7, 1, 5, 2, 5, 2, 1, 8, 1, 5, 3, 8, 8, 8, 1, 9, 1, 8]

Target: 7663

Trial 1: 0.5s runtime
4+1/9-1/3*4+1+1+3*6/4/2*6+8*6/5*7+5/6/4+4*6/6+4-5*1-7/4*8*7-8*2/8*6+8*6-9-7+2-8+1/4*3-9*3+1*5*9/8/3+2/2+7+1-2/8-9-5*7*9/4-3+8/9-6*2/4/5+5*3*6-7*8*6-8+2/5-8+3+2*8-7+7/9-5-2*5-2+1*8/3-5/1-8*8-8+1/9/1/8
distance from target: 0.003395061725314008

Trial 2: 1.5s runtime
5-4*8/3+2+6-9*1-2-1-5-8/2+9+2+3*8*4*3*7-8+3-9+8/9/6*3/8/2*2/6/5+5+1+8-3*1/3/1-9/7*2/4+8/8-6/6-8+8/8*1-9-1*3+6-8-5+4-8/7+2/5*7*1/4/9/5-1*4+2+6/5-8*2*1*4+4*5+3*7/5/5/6-7/6+8*4*6*8/9+6/8+4+1-1*9*7+2-7*7
distance from target: 2.1146661310922354e-0

Trial 3: 4s runtime
3-4/5-5/6*3+3*6+9-2+1*8/8*9-7+8/5+6-9*5/1-8/8/4-7-2*6-9-8*5-8+4*6/9*8*3/1-4/8*1/7/2+6/9*1/2+7/5*8*7*8/8+1*1+2+9*2+1/4*2*1/6+7+6/8*1+2/7*3/5*6/4+5/8/3/9+5/2*8*6-3+7+7+4-5-2+3*1+4-8*5*8+3*9+8-4-2+4-1/6
distance from target: 0.0