Assignment 02 key

1. (20 points) Explain the difference between a problem state and a search state.
   A problem state is a representation of a world (problem) space. As an example, the N-Puzzle state represents the current position of the tiles and blank slide. In contrast, a search state is a node in a search tree. It contains a problem state, but it also contains other information such as the action, cost to arrive at the node and possibly the estimated cost to a goal node. Its relationship to other search states in the tree indicate what set of actions were required to arrive at this state.

2. (20 points) Work problem 4.1 from your book. For 4.1 c, ignore that the simulated annealing algorithm presented in figure 4.5 would terminate immediately when the temperature is zero.

   a. Local beam search with beam width of 1: Takes the single best option. Equivalent to hill climbing (gradient) search. (It is similar to depth first search, but there is no backtracking as nodes outside the beam are discarded.)
   b. Local beam search with beam width of ∞: All children are expanded. As each child is processed in the next round, this is like a breadth first search.

   For c and d, it may be helpful to recall that simulated annealing transitions to a new state based on the change of fitness between a child and the current state and the current temperature:

```
ΔE=fitness(child)-fitness(state)
if ΔE > 0:
     state = child # life is better
else:
     # might be a bad state, but perhaps take a
chance
     roll = random number on interval [0,1]
     if roll < 1+exp(ΔE/temperature):
          state = child
```

   c. Simulated annealing, T=0 at all times. When the temperature is 0, the exponential term will be driven high (this actually goes over 1 – it is not a valid probability) when $\Delta E>0$ and to zero when $\Delta E<0$. This means the first choice that is explored that improves the fitness will be accepted. This is a first choice hill climbing search.

d. Simulated annealing, $T=\infty$ at all times. Regardless of the change in fitness, the probability of accepting the first choice to be generated will be accepted. This is a bit like a stochastic search where the first node has a probability of 1 and the others have a probability of zero, although it is unrelated to steepness. Alternatively, we could consider it to be similar to a random walk.

e. Genetic algorithm with population size N=1. Crossover will have no effect. Instead, there is only a probability of mutation which will allow a random change of state. Of the algorithms we covered, this is perhaps most closely related to random restart. (This could also be considered similar to a random walk that we did not cover but you may have seen in other courses.)

3. (20 points) Consider the problem of managing the reproduction of a critically endangered species in a captive environment. One fitness function could be the population size although more sophisticated ones might take into account indicators such as the number of births, mortalities, genetic diversity, and behavioral indicators of stress, fertility, etc. Assume that you are managing this population and that your facility has the following resources:

- N_m males
- N_f females
- N_d units of food/day
- N_e enrinchement items (toys to prevent boredom)

Devise a state representation that partitions the food, animals, and enrichment items amongst three habitat units. Propose a crossover function that is different from the one used on the N-queens problem that could be used in a genetic algorithm search if we could predict the fitness from your state.

Answers may vary. Sample answer:

State is represented as a 12-tuple with one attribute for each resource:

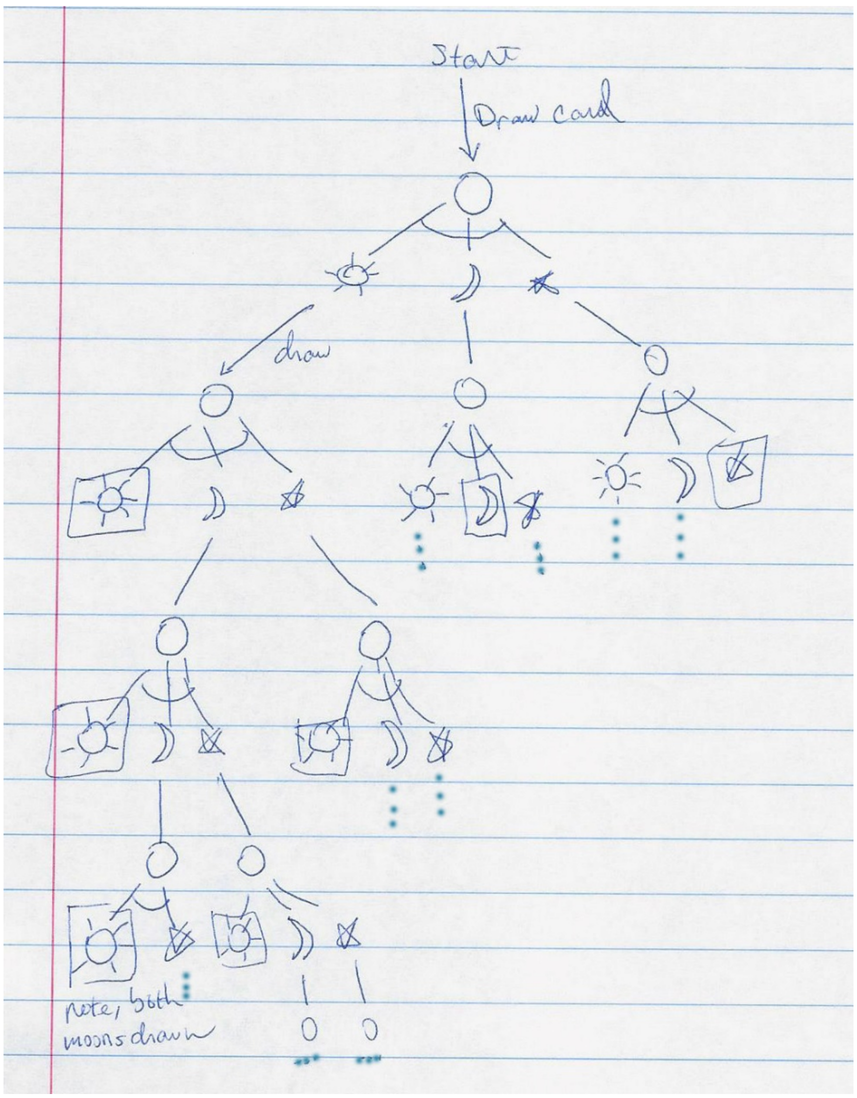
$$\left(\underbrace{m_1, f_1, d_1, e_1}_{\text{habitat}_1}, \underbrace{m_2, f_2, d_2, e_2}_{\text{habitat}_2}, \underbrace{m_3, f_3, d_3, e_3}_{\text{habitat}_3}\right)$$

```
crossover(s1,s2) {
   attributes = pick 1 to 3 attributes at random
    from resource domains
   for a in attributes {
      swap(s1[a], s2[a]);
    }
```

```
    return (s1, s2)
}
```

4. (20 points) Consider the fairly simply problem of selecting 2 matching cards from a deck of cards with pictures on them. The pictures consist of moon, sun, and stars, and there are two instances of each card (6 cards in total). Legal moves are pick a card that has not been picked, and the goal state is matching the first card drawn. Sketch an and-or search tree for this problem. You do not need to draw the full tree, just enough to make it clear that you understand what the full tree would look like.

```
Answers will vary, but should look something like this:
```




```
Goal nodes have boxes around them.  (note: bottom, 2nd left should
have the "smiley face" connector binding arcs)
```