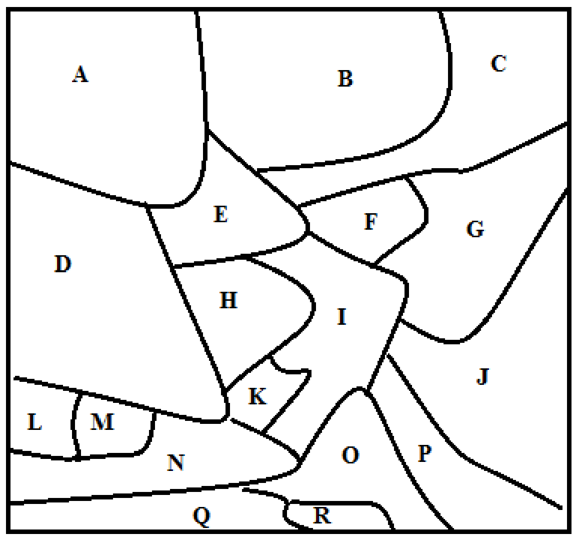
CSC 425/525 Homework #9 (Chapter 12-13)
Due: Wednesday, April 19

Word process all answers. Figures may be hand drawn. Undergraduates answer question 1 and four other questions. Graduate students answer all six questions.

1. Graph coloring is an optimization problem that asks what is the minimum number of colors we can assign to a graph such that each region is a different color from any region that is adjacent to it. For instance, the figure before has 18 regions (labeled A-R). What colors can we assign such that no region has a color of a neighboring region. Let's number the colors, say 1-10. Can we assign each of A-R a number such that no neighbor has the same number? One solution would be A = 1, B = 2, C = 1 (since A and C do not touch), D = 2 (B and D do not touch), E = 3, F = 2, G = 3, H = 1, I = 4, J = 1, K = 3, L = 1, M = 3, N = 4, O = 1, P = 2, Q = 2, R = 3. So our answer is 4 (colors). Is this the best we can do? We want to solve this problem using GAs. Answer the following questions.



   a. What would our chromosome look like? That is, specify how we represent a graph coloring problem as a chromosome.
   b. Of the three genetic operators, mutation, inversion, cross-over, which would make sense to apply? Explain.
   c. Define a fitness function to evaluate a chromosome. Would your GA be looking to minimize or maximize this value?
   d. Is this an optimization problem where we would know when we reached the optimal solution? Is so, how? If not, what would you recommend in terms of how long the GA should run (when would it stop)?
   e. What size population of parents and children would you select? How would you select the children? (strictly by fitness, or some other approach). Explain your choices.
   f. Should the same chromosome be allowed to appear twice in any single generation of parents? If not, should the same chromosome be allowed to appear twice at all during the entire run? Explain.

2. Assume we have a sorting function. Can we use genetic programming to evolve a better sorting function (that is, one that has a superior complexity)? Explain. Assume we have

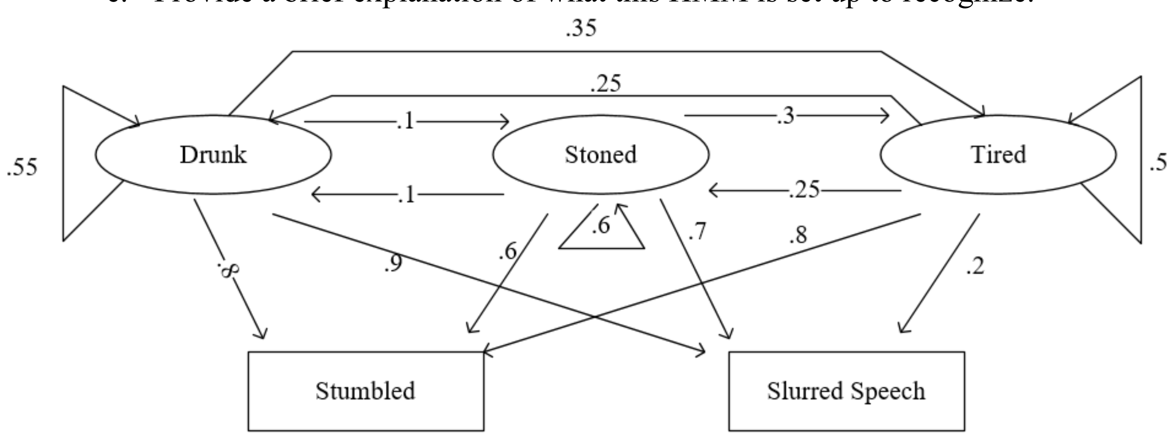
the following code for sequential search of an ordered linked list. Can we evolve a better set of code?

```
while(current!=null)
        if(current.getData()==target) return current;
        else current=current.getNext();
```

3. Given the data from question #4 on hw7 (student majors given their interests), compute the following. NOTE: this is not a matter of applying Bayes theorem, its just a matter of counting.
    a. P(CSC major)
    b. P(MIN Major | Problem solving = yes)
    c. P(DSC Major | Math = Y and Analysis = Y and Theory = N)
    d. P(Math = Y or Theory = Y | DSC Major or CSC Major)
    e. P(MIN Major or CIT Major | Money = N and Prog = N)

4. Consider a naïve Bayesian classifier used to categorize web pages using a "bag of words" approach. The NBC for the topic "artificial intelligence" was trained on 50 different pages, half of which are tagged as AI pages and half as not AI pages.
    a. Is this a form of supervised or unsupervised learning? Briefly explain.
    b. Would you consider this "training" as a form of learning? If so, what type from chapter 10 slide 3 would you consider it to be (e.g., rote, parameter adjustment, explanation based learning, analogical reasoning)? Provide a brief explanation.
    c. This NBC has not performed very well, so we trained it on 200 additional pages (half of which are AI pages, half of which are not). Would you consider this second round of training to be a forming of learning? If so, what type of learning (e.g., rote, parameter adjustment, explanation based, analogical reasoning, etc)? If not, why not? Again, briefly explain.

5. Given the following HMM, provide the following pieces of information.
    a. What are the hidden states?
    b. What are the observations?
    c. What are the transition probabilities?
    d. What are the emission probabilities?
    e. Provide a brief explanation of what this HMM is set up to recognize.



6. Similar to the "learned" organization for research articles as shown in slide 19 of the chapter 13 notes, provide your own Bayesian network for a Java for loop. It should include both normal for loops and iterator for loops. States should be generic such as "initialization", "comparison", etc but should permit variations like declaring your variables in the init step, having complex conditionals for comparisons, etc. For transition probabilities, make them up based on your own experience with Java for loops.