**Lab 8 – Genetic Algorithms & AI**

1. **Study the following paper that uses GA** to solve the Knapsack problem.

<http://www.micsymposium.org/mics_2004/Hristake.pdf>

Make sure you understand it very well. The main goal here is to enable you to understand some published paper well and relate it to the content of the lecture covered in the class. This paper will also enable you to do some research work as would be needed in future e.g. when you join an industry.

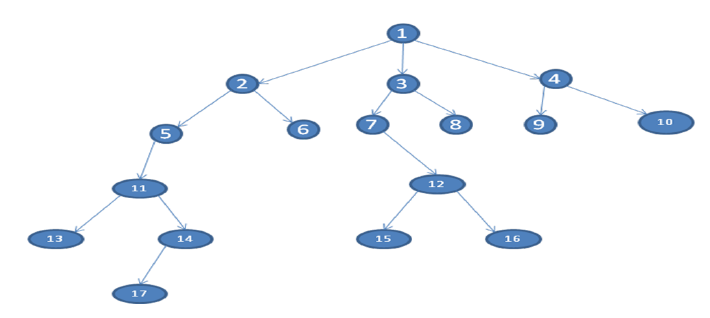
***Nothing is due for this BUT your understanding will be helpful to answer questions on GA in the Final exam. Also, your understanding of the paper will help you solve Problem #2 below.***

1. You have 5000 MP3 files sitting on your computer's hard disk. Unfortunately, the hard disk has started making noises, and you decide that you had better back up the MP3s. Equally unfortunately, you can only burn CDs, not DVDs, on your computer. You need to minimize the number of CDs that you use, so you decide to design a genetic algorithm to choose which MP3s to put onto each CD in order to fill each CD as completely as possible. Design a genetic algorithm to solve the problem. You will need to consider how you would encode the inputs, which genetic operators are suitable, and how you would make the genetic algorithm deal with the fact that you have multiple CDs, not just one CD. (Note: You can use some ideas from #1 above).
2. For each of the following Agents, develop a PEAS description of the task environment:
3. Robot soccer player
4. Internet book-shopping Agent
5. Autonomous Mars rover
6. Mathematician’s Theorem Proving Assistant
7. Consider a Natural Language based Medical Assistant Agent.

Draw its architecture along with the environment. Mention important goals and utility functions.

1. For the following tree, list the order in which the nodes are visited using Depth-First

Search.



1. The travelling salesman problem (TSP) is an NP-Complete (i.e. complex – needs exponential time to solve). TSP asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?

Try to solve this problem using GA. Then implement your solution in Python. You would need to import numpy, pandas, random and operator.