Rohit Ravishankar
rr9105@rit.edu

Home Work 1

CSCI - 630 Foundation of Intelligent Systems

1. Read this article. It includes the comment: "So tell me: Will AI machines procrastinate?" Answer this question, in the context of the article's discussion of developing AI systems. Your answer should be a decent-length paragraph and should include some of the arguments from the article as well as your own personal opinions.

   Ans. In my opinion AI machines will **not** procrastinate. My view stems from the idea that the use for AI machines to perform the same human tasks, is to do so in an accurate and more efficient way than humans could or would. If AI machines performed the same tasks as humans, but also experience the same feelings as humans do such as the need to procrastinate, then we aren't benefiting by the use of AI machines.

   The author argues that "promises of AI are based on a flawed premise: that we understand human intelligence and consciousness." and further mentions of how our understanding of intelligence, consciousness and human mind is limited. A view on the article which we can confidently concur with is that while we may be able to create a brain, it does not imply we can create a mind or consciousness. My conjecture is closely aligned with the view that if we do not understand consciousness or the human mind, we cannot recreate it. Hence, we cannot recreate the sense or the need for a feeling such as procrastination in an AI machine.

2. For each of the following, give a PEAS description of the task and given solver of the task. There may be several reasonable answers, but the key is that all four parts of your answer go together.

2.1. Figuring out your optimal schedule for next semester

Performance	• No overlap of classes
Environment	• List of all courses offered by the university • Days/times the courses are held • Location of the classes • Identify if you are on the waitlist or enrolled • Number of credits for each course
Actuators	• Enroll into non-overlapping classes prior to end of add/drop period • Courses to count towards graduation credits and/or specialization • Enroll in swap for a different course
Sensors	• Overlapping courses should be prohibited • Courses with overlapping exams should be prohibited • Define syllabus of the course before the end of add/drop period • Review all the courses enrolled in prior to end of add/drop period

2.2. Solving a crossword puzzle

Performance	• Time it takes to solve the crossword • Accuracy of the solved crossword • Theme of crossword based on all solutions
Environment	• Language used • Vertical/Horizontal orientation of crossword • Crossword is on paper or digital in nature
Actuators	• Fill in alphabets • Fill in words based on the hints provided for a given number
Sensors	• Identify the hints • Does your guess word fit in the number of spaces • Do all the words make sense together to come up with the final word solution/theme of crossword

2.3. Human playing soccer

Performance	• Goals scored • Successful passes • Goals prevented/saved • % ball possession • Number of fouls committed
Environment	• Soccer field of specific dimensions with line demarcations • Soccer ball of specific dimension • Weather conditions
Actuators	• Motion(forward/backward/turn) • Passing/shooting the ball • Saving/defending the ball • Tackling the opponent
Sensors	• Vision/sight to track the ball • Communication with your teammates • Reflex to tackle/pass/shoot

2.4. Small robots playing soccer

Performance	• Goals scored • Goals prevented/saved • % ball possession • Battery capacity of the robot
Environment	• Soccer floor(green colored) with line demarcations • Specific size/colored soccer ball • Off-field wireless communication with the referee
Actuators	• Motion(forward/backward/turn) • Passing/shooting the ball • Saving/defending the ball from the opponents
Sensors	• Proximity sensors • Transmitter/Receiver • Boundary sensors/line sensors • Ball tracking(vision system) sensor

3. On the British game show Countdown, the Numbers round consists of contestants getting six numbers and having to use a subset of those numbers together with the four standard arithmetic operations to come up with a given target value. (See example at the link.) In this question, you are to consider a program that will solve the numbers round (if possible). Note that each final solution can be thought of as a sequence of pairwise operations (e.g. for the

solution given to the Wikipedia example, 75 + 50 = 125; 125 - 8 = 117; 117 * 7 = 819; 3 * 2 = 6; 819 - 6 = 813).

- What would be the state representation you would use for this problem?
  Ans. (a, b, c, d, e, f) such that {a, b, c, d, e, f} in ${}^{20}C_X$ and ${}^{4}C_Y$ and $X + Y = 6$

- What is the initial state for the first example given at Wikipedia?
  Ans. (75, 50, 2, 3, 8, 7)

- What are your successor functions? Be specific.
  Ans. +, -, *, / are the operations that can be performed on any 2 given numbers

- What is the branching factor of your approach?
  Ans. ${}^{6}C_2 * 4 = 60$ [From the given set of numbers the way to pick 2 numbers is possible in ${}^{6}C_2$ and 4 possible operations on these numbers]

- What is the maximum search depth of your approach?
  Ans. 5 levels [There are 6 numbers and the maximum search depth would be 5]\

- Imagine that instead you wanted to generate all possible three-digit numbers that can be achieved with a given initial set of six numbers, instead of just reaching a target. Would the answers to the previous parts change? If so, how? If not, why not?
  Ans. No, it won't change. Even if we had to find all possible numbers, we would need to continue generating all possible numbers and not need to stop as unlike number round there is no fixed target number to be reached.

4. Write a Python program to perform the search problem mentioned on the top of page 73 in the textbook (Knuth's conjecture). Note that while Python allows arbitrarily large integers (handy for factorials!), its math libraries are built on C code, so you cannot compute square roots of numbers over 10^300 - this will raise an exception, so don't try - and you also don't need to bother computing factorials that will result in numbers that large.

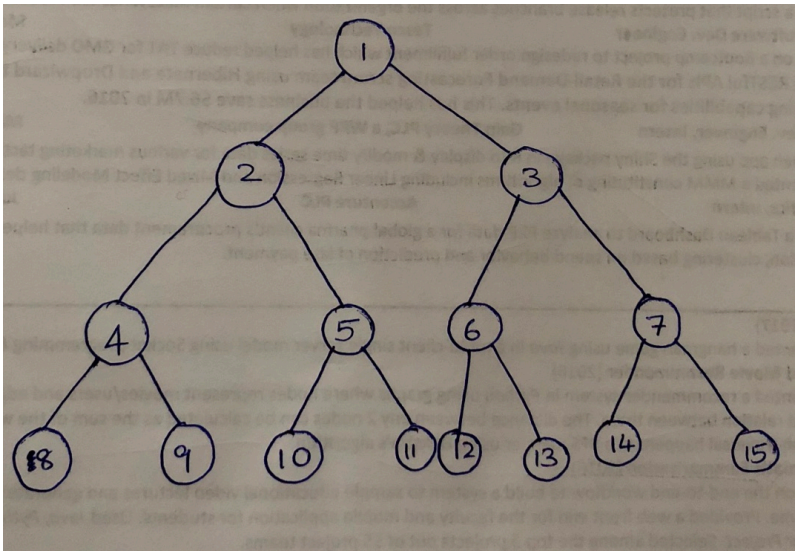
   The function you write, `bfs`, should take a single argument, the goal integer, and return the (minimal) sequence of numbers required to reach that goal with the given operations (you are welcome to write other functions as well). You do not need to list the operations themselves.

   Turn in your code along with results for `bfs(5)`, `bfs(8)`, `bfs(13)`, and another positive integer of your choice.

   Submitted as a part of the code.

5. R&N 3.15 a-d

3.15 a



3.15 b

Breadth-First Search: 1 2 3 4 5 6 7 8 9 10 11

Depth-Limited Search: 1 2 4 8 9 5 10 11

Iterative-Deepening Search: 1; 1 2 3

1 2 4 5 3 6 7; 1 2 4 8 9 5 10 11

3.15 c

Bidirectional search works well on this problem because the successor of k in the reverse direction is $\left\lfloor \frac{k}{2} \right\rfloor$ (for example, the parent node of 5 is 2). This causes the search to be better focused. The branching factor is 2 in the forward direction and 1 in the reverse direction.

3.15 d

Yes; start at the goal, and perform single reverse successor action until you reach 1