VJ Davey

CTBA Fall 2016

Automate This! Assignment

1. **What is an algorithm?**

“An algorithm is a set of instructions to be carried out perfunctorily to achieve an ideal result” – P 54.

1. **What are the characteristics of a good algorithm and software implementation thereof? Why are these characteristics essential for an algorithm to be classified as good?**

A good algorithm is one which is written clearly, explicitly, and concisely, solves or approximates the solution to some problem, and terminates cleanly. A good software implementation of an algorithm is one written which meets these same standards while being written in the programming language of choice. These characteristics are essential for an algorithm to be classified as good for several reasons. Firstly, an algorithm must be written clearly and concisely because the algorithm is supposed to be a set of instructions to follow. There should be no room for interpretation with the steps of an algorithm. Machines can’t reason, at least yet, they must be told explicitly what to do. The second part about solving or approximating the solution to some problem is self-explanatory. Lastly, the algorithm must terminate cleanly. An algorithm which does not terminate serves to only eat up resources and never brings us to the solution or result we desire. It is important to ensure that algorithms terminate and do so cleanly.

1. **In which circumstances mentioned in the book did hardware need to be developed in order to accompany the algorithms embedded in software? In your experience can you recall any situations where algorithmic software works hand in hand with hardware?**

In the book, one situation mentioned where hardware needed to be developed in order to accompany algorithms embedded in software involved Thomas Peterffy’s creating a mechanical piston to manually type in trade requests solved for algorithmically to meet standards for participating in the Stock Exchange. In my own experience, I have seen numerous situations where algorithmic software works hand in hand with hardware. One which stands out to me right now would be automated sprinkler systems. Though I don’t personally know the exact algorithms used to make a sprinkler operate, I would assume that systems I have seen which can be programmed to water a person’s lawn at designated times during the day make use of simple algorithms which dictate when the sprinkler rotates and the amount of water it spurts out whenever the programmed time for the sprinkler to run approaches. The rotation of a sprinkler is something which is very clearly algorithmic in nature due to its simplicity and repetitiveness.

1. **What ethical issues or unintended consequences can result from implementing algorithms?**

Using algorithms to compete and gain an edge in domains typically reserved for human competition can be morally ambiguous, as was the case with Thomas Peterffy’s automated stock trading and the disdain he was met with when Jones came to visit him.

Using and implementing algorithms with technology can lead to efficient solutions to things like stock trading, sports training, daily activities and more. However, since everyone may not have the same access to technology, some people may be placed entirely out of the realm of competition against someone else who can make efficient use of algorithms and has access to technology.

Other ethical issues which can arise from implementing algorithms include the fact that algorithms are explicit in nature. This can be troublesome when algorithms involve automated hardware that has the potential to make life or death decisions. Problems include matters such as whether an automated car going at a high speed should hit a pedestrian that wandered into the road or swerve off the road and risk killing its passengers, and whether a bomb disarming robot should cut one wire or the other in a time sensitive situation. These kinds of problems are real problems that automated machines would have to face. Giving an explicit answer to these problems in some cases could be akin to giving a machine an “OK” to kill someone, as the answer to their problem was written in the code before the problem had arisen. Another problem also comes into play by answering these problems algorithmically. What happens when software is buggy? A machine can only do what it is programmed to do, and if giving the machine the green light to do something like run over a pedestrian is built into the code, then that means there is potential for that line of code to be executed at the absolutely wrong time, with extreme speed and precision. Therefore, it may be unethical to have such blocks of code included in the program at all. But then the problem remains of what the machine should do when it encounters that situation.