**Intro To AI:**

Part A – ImprovedGreedy:

1. The definition of the game is where:

In our case, it was given that . Therefore:

1. In order to define a smart heuristic that leads to a good outcome, we’re going to take into account the following variables of the environment into consideration:
2. The number of steps we have left until the end of the game in relation to the Manhattan distance from the nearest package drop off location, we’ll call this parameter: feasibility. What is meant by this, is: feasibility = num\_steps – drop\_off\_cost.
3. The Manhattan distance from the drop off location of the package in case the agent is currently holding a package, we’ll call this parameter: delivery\_cost.
4. The distance of the adversary from the nearest package in terms of Manhattan-distance in case both agents are currently looking for packages, we’ll call this parameter: pick\_up\_cost.

And the formula we’re going to use is:

1. The main disadvantage of using Greedy as opposed to Minimax is, in Greedy we make locally optimal decisions at every step for each of our agents whereas in Minimax we explore the game tree such that at each step we try maximizing the utility of our agent and then minimize the utility of our adversary at the next step. Thus, a possible outcome is that While using Greedy we might take a step that maximize the utility of our agent but doesn’t minimize the utility of our adversary and therefore result in a different result than that of Minimax’s.

Part B – RB-Minimax:

1. Using an easy to calculate heuristic in RB-Minimax is advantageous because it doesn’t waste much time on computing the heuristic and is able to delve deeper into the game and obtain results that are better than those computed at a lower depth. However, it could also be disadvantageous in some cases where the heuristic is misleading and wastes the precious time we have on bad steps as compared to the more informed but harder to calculate heuristic that gives us the actual good steps.
2. The behavior in Dana’s algorithm doesn’t necessarily mean she has a bug. For example, in case there are 2 goals such that one is 1 step away and the second goal is 2 steps away such that the utility that’s gained from the second goal is a lot bigger than that of the first goal. In such a case the algorithm will choose to take the path to the second goal even though it was possible to reach one of the goals in one step.