Assignment 2 part 2

Task 1 :

- ***Markov decision process (MDP)*** is a discrete time stochastic control process. It provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker. MDPs are useful for studying optimization problems solved via dynamic programming and reinforcement learning.

- ***Value Iteration Algorithm*** : Value iteration is a method of computing an optimal Markov Decision Process policy and its value.Value iteration starts at the "end" and then works backward, refining an estimate of either Q* or V*. There is really no end, so it uses an arbitrary end point. Let Vk be the value function assuming there are k stages to go, and let Qk be the Q-function assuming there are k stages to go. These can be defined recursively. Value iteration starts with an arbitrary function V0 and uses the following equations to get the functions for k+1 stages to go from the functions for k stages to go:

$$U_{i+1}(s) = \max_{a \in A(s)} P(s` \mid s, a)\,(R(s` \mid s, a) + \gamma\, U_i(s`))$$
$$P_{i+1}(s) = \operatorname{argmax}_{a \in A(s)} P(s` \mid s, a)\,(R(s` \mid s, a) + \gamma\, U_{i+1}(s`))$$

Task 2 :

Inference from Results of Task 1 :

- Total Number of Iterations for Convergence : 111.
- Whenever the stamina of Lero is 0 he always prefers to Recharge in order to increase his stamina and save his life.
- When the Stamina of Lero is non zero but the number of arrows are 0, then Lero has 2 options it can either Recharge or Dodge, but the final policy suggests that Lero always prefers to Dodge. This can be attributed to the fact that dodging may increase the number of arrows that Lero has which can be used in the future to attack the enemy and gain final reward.
- Whenever Lero has non zero arrows and non zero stamina he always Shoots in order to decrease life of Mighty Dragon.
- There are some exceptions like when the state is (4,1,1) according to above rules Lero should shoot but Lero prefers to Dodge according to optimal policy because enemy is too far away from killing so it is more optimal to gain arrows instead of shooting.
- Another exception is for the state (4,3,1), going by the general rule Lero should Shoot but as the enemy is much far from losing and number of arrows is also not low Lero tries to get some Stamina before trying to eliminate the enemy.
- So these are optimal states which can help Lero to live for the max time and kill Mighty Dragon.

Inference from Results of Task 2 part 1 :

- Total Number of Iterations for Convergence : 100
- The exceptions that we found in Task-1 do not arise in this task because the penalty for the Shoot action is much lesser (less negative) than other actions, so whenever Lero can, he Shoots, so no exceptions arise.

Inference from Results of Task 2 part 2 :

- Total Number of Iterations for Convergence : 5
- In this task basically value of gamma is very less as compared to part 1 and gamma is the discount factor which represents the priority value it gives to future states. If the value of the discount factor is less then it means that the agent is not giving preference to the future and he wants to do everything now only in present time and does not want to have any effect of the future on his actions. But more discount factor value means that the agent is giving high priority to the future and takes action considering the effect what happens in future. So in case of the high discount factor future has a very large effect on current actions of the agent.
- So in this part there are only 5 iterations whereas in part1 there were 100 iterations which is a huge difference. This is because in this part the agent is not thinking of the future and he wants to do everything now only and does not want to have any effect of future on his current action. Therefore without thinking anything about the future he converges very early.

Inference from Results of Task 2 part 3 :

- Total Number of Iterations for Convergence : 12
- Number of iterations increased to 12 as delta is made very small which implies more precision is required and hence the number of iterations has increased.