# Assignment-2 Part-3

Jyoti Sunkara - 2018101044

Shradha Sehgal - 2018101071

## Run the script

Run the following command to obtain the `output.json` in `outputs` folder

```
python3 solution.py
```

## Procedure for making the A matrix

Each state is uniquely identified by a tuple of three values. The tuple is ordered as <Dragon health, Number Of Arrows, Hero Health>.

- The domains for the tuple values are as follows:

```
    Dragon health: 0, 25, 50, 75, 100
    Number Of Arrows: 0, 1, 2, 3
    Hero Health: 0, 50, 100
```

Hence, there are 60 states in total.

- We iterate over all possible combinations of ***<Dragon health, Number Of Arrows, Hero Health>***.
- For each such unique state, we check the possible actions between ***SHOOT, DODGE, RECHARGE, and NOOP***.
- Each state has its own index value based on incrementation in the order: hero health, number of arrows, dragon health.
- For every possible action, we make a vector of length equivalent to the number of states. All values in this vector are zero except that at the index of the current state and the indices of the next states to which the action takes us.
- Say, if an action can take you from state A to B with probability ***P***. The value of ***-P*** will be assigned at the index of state B. And ***P*** will be added to the index corresponding to state A.
- We append the created vector for a state **only** when at least one action is valid from a state.

## Procedure for finding policy

- A policy specifies the action selection mechanism and is represented by a sequence of ***STATE - ACTION*** pairs.
- A`, R and alpha are used to generate the X vector that helps decide the policy.
- The X vector represents the utility of the STATE-ACTION pair obtained by passing ***alpha***, ***A*** and ***R*** to the LPP solver function.

- ***alpha*** is vector that holds the initial probabilities of the states. It is initialized at the start of the algorithm based on the starting scenario given. It is a list of zeroes with a one at the index of the state representing maximum arrows, full dragon health and full hero health.
- ***A*** is the two-dimensional array that represents the flow of probabilities of valid actions from each state.
- The ***R*** vector holds the reward of taking a *valid* action from each state.
- A linear program is an optimization problem with a linear objective and inequality constraints. Here, the LPP is to:

```
maximize RX
subject to AX  = alpha,
           X >= 0
```

- Once we obtain the ***X*** vector as a solution to the above LPP we begin iterating over all the states by incrementing in the order: hero health, number of arrows, dragon health.
- For each state, we iterate over all the X values associated with the state.
- We obtain the highest value of X possible for the state and add the ACTION associated to that X value to the POLICY.

## Can multiple policies exist?

The answer is *yes*. This is because for a state, actions with the same X value are interchangeable.

1. To generate a different policy we can change the condition where find the maximum value of X for an action. We can change from np.argmax() to iterating as **x[i] >= x[i+1]** over all relevant indices. The former finds the first highest value of x in case many x's have the same largest value. The second one finds the last x with the highest value.

2. Another way to do this without changing the comparison function for X is to consider the actions in a different order. Currently we proceed in the order ***SHOOT, DODGE, RECHARGE, and NOOP*** - if we change that then some other action may lead to the first highest value than the one in the current set up. An example is to change the actions array to:

```
actions = ["RECHARGE", "DODGE", "NOOP", "SHOOT"]
```

- This does not affect the ***A*** matrix as that is created before solving the LPP.

- The ***alpha*** vector holds the initial probabilities of the states. This does not change during the algorithm as the start state of the scenario is fixed.

- The ***R*** vector holds the reward of taking a *valid* action from each state. This is determined using the step cost of the action before solving the LPP and will hence not be affected by the policy.

- The ***X*** vector is the solution to the LPP with A, alpha and R as parameters. Since neither of the three vectors change, X will not change either. However the actions that take place in order to produce X will differ for a different policy.

3. Other ways to change the policy is to change the external world. That is, we can change the starting state and hence alter alpha. We can also change the step costs which would change the R vector. These changes will obviously impact the policy as the LPP solver gets different variables. However, this changes the question altogether. If we want to solve the same question, yet generate a different policy, only points 1 and 2 are applicable.