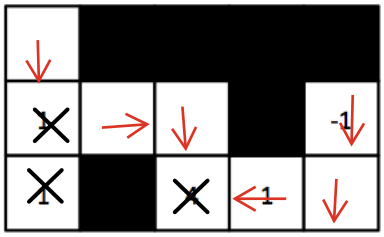
# Q1. MDPs: Reward Shaping

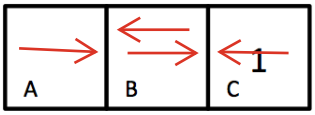
PacBot is in a Gridworld-like environment $E$. It moves deterministically Up, Down, Right, or Left, or at any time it can exit to a terminal state (where it remains). If PacBot is on a square with a number written on it, it receives a reward of that size **on Exiting**, and it receives a reward of 0 for Exiting on a blank square. Note that when it is on any of the squares (including numbered squares), it can either move Up, Down, Right, Left or Exit. However, it only receives a non-zero reward when it Exits on a numbered square.

**(a)** Draw an arrow in **each** square (including numbered squares) in the following board to indicate the optimal policy PacBot will calculate with the discount factor $\gamma = 0.5$. (For example, if PacBot would move Down from the square in the middle, draw a down arrow in that square.) If PacBot's policy would be to exit from a particular square, draw an X in that square.




In order to speed up computation, Pacbot computes its optimal policy in a new environment $E'$ with a different reward function $R'(s, a, s')$. If $R(s, a, s')$ is the reward function in the original environment $E$, then $R'(s, a, s') = R(s, a, s') + F(s, a, s')$ is the reward function in the new environment $E'$, where $F(s, a, s') \in \mathbb{R}$ is an added "artificial" reward. If the artificial rewards are defined carefully, PacBot's policy will converge in fewer iterations in this new environment $E'$.

**(b)** To decouple from the previous question's board configuration, let us consider that Pacbot is operating in the world shown below. Pacbot uses a function $F$ defined so that $F(s, a, s') = 10$ if $s'$ is closer to C relative to $s$, and $F(s, a, s') = 0$ otherwise (consider C to be closer to C than B or A). Let us also assume that the action space is now restricted to be between Right, Left, and Exit only.




In the diagram above, indicate by drawing an arrow or an X in each square, as in part (a), the optimal policy that PacBot will compute in the new environment $E'$ using $\gamma = 0.5$ and the modified reward function $R'(s, a, s')$.

**(c)** PacBot's utility comes from the discounted sum of rewards **in the original environment**. What is PacBot's expected utility of following the policy computed above, starting in state A if $\gamma = 0.5$? 0

**(d)** Find a non-zero value for $x$ in the table showing $F(s, a, s')$ drawn below, such that PacBot is guaranteed to compute an optimal policy that maximizes its expected true utility for **any** discount factor $\gamma \in [0, 1)$.

| | Value |
|---|---|
| $F(A, \text{Right}, B)$ | 10 |
| $F(B, \text{Left}, A)$ | $x$ |
| $F(B, \text{Right}, C)$ | 10 |
| $F(C, \text{Left}, B)$ | $x$ |

x <= -10

# 2 MDPs: Micro-Blackjack

In micro-blackjack, you repeatedly draw a card (with replacement) that is equally likely to be a 2, 3, or 4. You can either Draw or Stop if the total score of the cards you have drawn is less than 6. If your total score is 6 or higher, the game ends, and you receive a utility of 0. When you Stop, your utility is equal to your total score (up to 5), and the game ends. When you Draw, you receive no utility. There is no discount ($\gamma = 1$). Let's formulate this problem as an MDP with the following states: $0, 2, 3, 4, 5$ and a *Done* state, for when the game ends.

**(a)** What is the transition function and the reward function for this MDP?

R(s, Stop, Done) = s for s <= 5
R(s, a, s') = 0 oth

T(s, Stop, Done) = 1
T(0, Draw, s) = 1/3 for s in {2, 3, 4}
T(2, Draw, s) = 1/3 for s in {4, 5, Done}
T(3, Draw, s) = 1/3 for s in {5} 2/3 for s in {Done}
T(4, Draw, Done) = 1
T(5, Draw, Done) = 1
T(s, a, s') = 0 oth

**(b)** Fill in the following table of value iteration values for the first 4 iterations.

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $V_0$ | 0 | 0 | 0 | 0 | 0 |
| $V_1$ | 0 | 2 | 3 | 4 | 5 |
| $V_2$ | 3 | 3 | 3 | 4 | 5 |
| $V_3$ | 10/3 | 3 | 3 | 4 | 5 |
| $V_4$ | 10/3 | 3 | 3 | 4 | 5 |

**(c)** You should have noticed that value iteration converged above. What is the optimal policy for the MDP?

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\pi^*$ | Draw | Draw | Stop | Stop | Stop |

**(d)** Perform one iteration of policy iteration for one step of this MDP, starting from the fixed policy below:

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\pi_i$ | Draw | Stop | Draw | Stop | Draw |
| $V^{\pi_i}$ | 3 | 2 | 5/3 | 4 | 0 |
| $\pi_{i+1}$ | Draw | Stop | Stop | Stop | Stop |