Problem 1

$V_{OPT}^{(0)} = [0, 0, 0, 0, 0]$

States $\rightarrow \{-2, -1, 0, 1, 2\}$.

$$V_{OPT} = \begin{cases} 0 & \text{if isEnd} \\ \max_a Q_{OPT}(s,a) & \text{else} \end{cases}$$

$$Q_{OPT} = \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V_{OPT}(s')\right]$$

Computing Q values at $t=1$.

$Q^{(1)}(-1,-1) = 0.8(20+0) + 0.2(-5+0) = 15$

$Q^{(1)}(-1,+1) = 0.3(-5+0) + 0.7(20+0) = 12.5$

$Q^{(1)}(0,-1) = 0.8(-5+0) + 0.2(-5+0) = -5$

$Q^{(1)}(0,+1) = 0.3(-5+0) + 0.2(-5+0) = -5$

$Q^{(1)}(1,-1) = 0.8(-5+0) + 0.2(100+0) = 16.$

$Q^{(1)}(1,+1) = 0.3(100+0) + 0.7(-5+0) = 26.5$

$V_{OPT}^{(1)}(-2) = 0$, $V_{OPT}^{(1)}(-1) = 15$, $V_{OPT}^{(1)}(0) = -5$, $V_{OPT}^{(1)}(1) = 26.5$

$V_{OPT}^{(1)}(2) = 0.$

Computing Q values at $t=2$

$Q^{(1)}(-1,-1) = 0.8(20+0) + 0.2(-5+-5) = 14$

$Q^{(1)}(-1,+1) = 0.3(-5+-5) + 0.7(20+0) = 11$

$$Q(0,-1) = 0.8(-5+15) + 0.2(-5+26.5) = 12.3$$

$$Q(0,+1) = 0.3(-5+26.5) + 0.7(-5+15) = 13.45$$

$$Q(1,-1) = 0.8(-5+-5) + 0.2(100+0) = 12$$

$$Q(1,1) = 0.3(100+0) + 0.7(-5+-5) = 23.$$

$V_{OPT}^{(2)}[-2] = 0$, $V_{OPT}^{(2)}[-1] = 14$, $V_{OPT}^{(2)}[0] = 13.45$,

$V_{OPT}^{(2)}[1] = 23$, $V_{OPT}^{(2)}[2] = 0$.

Prob 1.(b)

The optimal policy: $\pi_{OPT}(s) = \underset{a}{\operatorname{argmax}}\, Q(s,a)$

$\therefore \pi_{OPT}[-1] = -1$, $\pi_{OPT}[0] = +1$, $\pi_{OPT}(1) = +1$

Problem 2

(b) Because it is an acyclic MDP, we can have the topological sort of the nodes such that for a node 'i', all its parents come before it in the topological array & all its children after it.

Given this topological array, we start at the end of the array, we know that $V_{OPT}(end) = 0$. $\forall$ node $\in$ End state

Now, go in order from last to first, and $\forall$ node compute the V_{OPT} score, as follows.

$$V_{OPT}(s) = \max_{a} \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_{OPT}(s') \right]$$

$[\forall s \notin \text{End state}]$.

In the above expression, s' are the children of 's' and hence come after it in the topological sort & hence we already know their value before we come to computing $V_{OPT}(s)$. Hence we can compute this in linear time w.r.t. # states 's'.

Hence, its very efficient!

Problem 2

(c) $Q(s,a) = \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V_{OPT}(s')\right]$

$= \sum_{s'} \gamma T(s,a,s')\left[\frac{R(s,a,s')}{\gamma} + V_{OPT}(s')\right]$

From the above eqn it is clear, what we need to do. We reweight all the rewards $R(s,a,s')$ by a factor of $\frac{1}{\gamma}$. We reweigh all the transition probabilities by a factor of γ. Now, we have for each (s,a) pair a probability $= 1-\gamma$, of ending at node '0' with a reward of 0 (zero). Note '0' is an "end" state. $\Rightarrow V_{OPT}('0') = 0$.
So with the above definition, of a new MDP, we have

$$\tilde{Q}(s,a) = \left(\sum_{s'} \gamma T(s,a,s')\left[\frac{R(s,a,s')}{\gamma} + V_{OPT}(s')\right] \right) + 1-\gamma\,[0+0]$$

{Note, here the discount factor is 1}.

$\Rightarrow \tilde{Q}(s,a) = Q(s,a)$

Hence, $\tilde{V}_{OPT}(s) = V_{OPT}(s)$

Hence, we can solve the above modified MDP problem with $\gamma = 1$ and get the soln for the original MDP directly.

Prob 4b

Comparing the policies, I see that for the smallMDP the Value Iteration and the Q-learning agent have the exact same policy! [re running it, I sometimes get a mismatch of 1/38 policies].
Note, I did not set exploration policy to zero, because I used

$$\pi_{OPT}(s) = \underset{a}{\text{argmax}}\ Q(s,a).$$

for computing the optimal policy, which should give the correct answer!

Running the same on largeMDP, I get a mismatch of ~ 900/2950 policies, which is ~30% error. This seems high.
I think this is high because the states of the large MDP are high, so probably the Q-learning agent wasn't able to explore them properly.

oblem 4

(d) For comparison purposes, I ran 30,000 trials of Value Iteration and Q-learning algorithm on the new MDP. The value iteration was trained on the original MDP.

Value Iteration score = 204,793 ~ 6.826/trial
(Value iteration was trained on original MDP).

Q-learning score = 272467 ~ 9.082/trial.

Explanation: Note sum of all cards in the deck is 12. Threshold obd for original MDP is 10 & for new MDP is 15. Hence Q-learning learns to never quit, and soon starts scoring very high (~12). In the beginning phase its exploring and learning so it quits sometimes and gets avg rewards ~6. Hence the total avg ~ 9.
But Value iteration never learns. It plays according to the 10 thoushold, where it is sensible to quit in the range of [6-10]. Hence it gets an average reward of ~7, and it does not try to improve!