Ryan Worley

1) Value iteration:

SuccProbReward:

	prob	state:	reward
-1:	80%	s-1	-5
	20%	s+1	-5
+1:	70%	s+1	-5
	30%	s-1	-5

ending states: (terminal)

	reward
+2	+100
-2	+20

Discount: 1

Pseudo Code:

- V['state']=0 for all states -2,-1,0,1,2 to start
- V[+2,-2] are end states, $\pi[2]=0$ $\pi[-2]=0$

a) Iteration 0: Iteration 1 Iteration 2:

state	V_{opt} i=0	V_{opt} i=1	V_{opt} i=2
-1	0	-4	2.69
1	0	11.0	40.34
0	0	66.5	66.5

b) π_{opt}:

$V_{\pi_{opt}}(-1)$ = add
$V_{\pi_{opt}}(0)$ = add
$V_{\pi_{opt}}(1)$ = add

- Done via python code

~~2a) See code~~

2b) For acyclic search problems, can use dynamic programing to store the optimum value for each node. Dynamic programing would calculate optimum path through the acyclic mdP, and all optimum policies from each node would be stored in cache, all non-optimum policies wald not be stored. Since there is no probability of returning to the prev state from future states (acyclic) dynamic programing will return a cache of optimum policies from each state.

2c) We want to define $T'(s,a,s')$ and $Reward'(s,a,s')$ in terms of OG problem w/ $\gamma = 1$.

normal transition

Probs added together

Special new state, end state

$$T'(s,a,s') = \gamma T(s,a,s') + (1-\gamma)\, T(s,a,o)$$

- w/ prob $1-\gamma$, MDP terminates @ any state.

$$V_{opt}(s) = \sum_{\substack{max \\ a \in actions}} T(s,a,s')\left[Reward(s,a,s') + \gamma V_{opt}^{t+1}(s')\right]$$

$$V_{opt}(s) = \gamma T(s,a,s')\left[\frac{1}{\gamma} Reward(s,a,s') + V_{opt}^{t-1}(s')\right]$$

- thus -

if $V_{opt}'(s) = V_{opt}(s)$

$$\boxed{T'(s,a,s') = \gamma T(s,a,s')} \quad \boxed{Reward' = \frac{1}{\gamma} Reward(s,a,s')}$$

4a) Code

4b) For small MDP, $2/27$ policies are different, very close to optimal algorithm found for Q-learning.

For large MDP, $894/2745$ states are different.

For large MDP, state space is too large to fully explore, not using generalization enough, need better features to describe states.

- See code for description on how this conclusion was reached.

4c) Code

4d) Expected reward from Fixed RLA alg: 8.52

Expected reward for Q-learning: 9.225

Reward for Q-learning higher because Q-learning creates new policy that fits new game played, where Fixed RLA policy is no longer the optimum for the new game. The rewards for both trials are somewhat close (Fixed reward still high because MDP didn't change too much) however, new Q-learning policy gives better reward