1. (a) $v_*(s) = \max q_*(s.a)$

(b) $q_*(s,a) = \max p(s',r|s,a) v_*(s)$

(c) $\pi(a|s) = \max q_*(s.a)$

(d) $\pi(a|s) = \max_a \sum_{s',r} p(s',r|s.a)[r+\gamma v_*(s)]$

(e) $v_\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s.a)[r(s.a) + v_\pi(s')]$

$v_*(s) = \max_a \sum_{s'} p(s'|s.a)[r(s.a) + v_*(s')]$

$q_\pi(s.a) = \sum_{s'} p(s'|s,a)[r(s,a) + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a')]$

$q_*(s,a) = \sum_{s'} p(s'|s.a)[r(s,a) + \gamma \max_{a'} q_*(s',a')]$

2. (a) The following algorithm is the entire defination for policy iteration.

```
1. input. $q_*(s.a) = 0$, $s \in S^+$. $a \in A$
2 # Policy evaluation.
  Loop:
     $\Delta \leftarrow 0$
     Loop for each $s \in S^+$:
        $v = V(s)$
        $V(s) = \sum_{s',r} p(s',r|s.\pi(s))[r+\gamma V(s')]$
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
```

```
Until $\Delta < \theta$
3. Policy Improvement
    Policy-stable ← true
    For each $s \in S$
        old action ← $\pi(s)$
        $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s',r|s,a)[r+\gamma V(s')]$
        If old-action $\neq \pi(s)$, policy-stable ← false.
    If policy-stable
        break
        return $V \approx v_*$  $\pi = \pi_*$
    else: go to 2.
```

(b)

$$q_{k+1}(s,a) = \sum_{s',r} p(s',r|s,a)\left[r+\gamma \max_{a'} q_k(s',a')\right]$$

3. (a) optimal policy at x: keep taking action c until get z

optimal policy at y: keep taking action b until get x

then keep taking action c until get z

(b) Policy evalution $\pi = c$

$V_x = [0.9 \times (-1+V_x) + 0.1 \times (-1+V_z)] = -1 + 0.9\ V_x$

$V_x = -10$

$V_y = [0.9 \times (-2+V_y) + 0.1 \times (-1+V_z)] = -2 + 0.9\ V_y$

$V_y = -20$

Policy Improvement:

$-1.8-16-2$

Iter1 - X: $V_{\pi c} = -1 + 0.9\ V_x = -10$

$V_{\pi b} = 0.8 \times (-2 + V_y) + 0.2 \times (-1 + V_x) = -1.6 + 0.8\ V_y - 0.2 + 0.2\ V_x$

$= -1.8 + 0.8\ V_y + 0.2\ V_x$

$= -19.8$

$\Rightarrow$ newaction $= \arg\max(V_{\pi c}, V_{\pi b})$

$= \pi_c$

$\Rightarrow$ newaction $=$ old action

$\Rightarrow$ policy stable

y: $V_{\pi c} = -2 + 0.9\ V_y$

$= -20$

$V_{\pi b} = 0.8 \times (-1 + V_x) + 0.2 \times (-2 + V_y) =$

$= -0.8 + 0.8 V_x - 0.4 + 0.2 V_y$

$= -1.2 - 8 - 4$

$= -13.2$

$\Rightarrow$ newaction $= \arg\max(V_{\pi c}, V_{\pi b})$

$= \pi_b$

$\Rightarrow$ newaction $\neq$ old action

$\Rightarrow$ policy not stable

Iter-2: y: $V_{\pi c} = -2 + 0.9 V_y$
$= -20$

$V_{\pi b} = 0.8 \times (-1 + V_x) + 0.2 \times (-2 + V_y) =$
$= -0.8 + 0.8 V_x - 0.4 + 0.2 V_y$
$= -1.2 - 8 - 4$
$= -13.2$

$\Rightarrow$ newaction $= \text{argmax}(V_{\pi c}, V_{\pi b})$
$= \pi b$
$\Rightarrow$ newaction $=$ old action

$\Rightarrow$ policy stable

(c) Policy evalution $\pi = b$

$V_x = -2 + 0.8 V_y - 1 + 0.2 V_x$

$V_x = -3 + 0.8 V_y + 0.2 V_x \quad \Rightarrow 0.8 V_x = -3 + 0.8 V_y$

$V_y = -1 + 0.8 V_x - 2 + 0.2 V_y$

$V_y = -3 + 0.8 V_x + 0.2 V_y. \quad \Rightarrow 0.8 V_y = -3 + 0.8 V_x$

$\Rightarrow$ No solution

add discounting γ

$\Rightarrow V_x = -2 + 0.8\gamma V_y - 1 + 0.2\gamma V_x$

$(1 - 0.2\gamma) V_x = -3 + 0.8\gamma V_y$

$\Rightarrow V_y = -1 + 0.8\gamma V_x - 2 + 0.2\gamma V_y$

$(1-0.2\gamma) V_y = -3 + 0.8\gamma V_x$

$\Rightarrow$ solution exist $\Rightarrow$ discounting help

if $\gamma = 0.1$ Policy evalution,

$0.98 V_x = -3 + 0.08 V_y$

$0.98 V_y = -3 + 0.08 V_x$

$\Rightarrow 12.005 V_y = -3 + 0.98 V_x$

$\Rightarrow 12.005 V_y + 3 = -3 + 0.08 V_y$

$11.925 V_y = -6$

$V_y = -0.503$

$0.98 V_x = -3 - 0.04$

$V_x = -3.1$

Policy improvement

Iter1 $\pi_1(x) = \mathrm{argmax}(V_{\pi c}, V_{\pi b})$

$= c$

$\pi_1(y) = \mathrm{argmax}(V_{\pi c}, V_{\pi b})$

$= b$

$\pi_1(y) = b = \pi_0(y)$

$\Rightarrow$ stable

Iter $\pi_2(x) = \mathrm{argmax}(V_{\pi c}, V_{\pi b})$

$= c$

$\pi_2(x) = c = \pi_1(x)$

$\Rightarrow$ stable

if $\gamma = 0.5$ Policy evalution,

$0.9 V_x = -3 + 0.4 V_y$

$0.9 V_y = -3 + 0.4 V_x$

$2.025 V_y = -6.75 + 0.9 V_x$

$2.025 V_y + 6.75 = -3 + 0.4 V_y$

Policy improvement

Iter1 $\pi_1(x) = \mathrm{argmax}(V_{\pi c}, V_{\pi b})$

$= c$

$\pi_1(y) = \mathrm{argmax}(V_{\pi c}, V_{\pi b})$

$= b$

$\pi_1(y) = b = \pi_0(y)$

$\Rightarrow$ stable

$1.625V_y = -9.75$

$V_y = -6$

$V_x = -9.56$

Iter $\pi_2(x) = \arg\max(V_{\pi c}, V_{\pi b})$

$= c$

$\pi_2(x) = c = \pi_1(x)$

$\Rightarrow$ stable

$\pi(\gamma = 0.5) = \pi(\gamma = 0.1)$

$\Rightarrow$ Optimal policy do not depend on γ

5(b)

The reward have been changed as following:

One car can be moved from lot1 to lot2 free;

If cars in parking lot1>10, additional $4 are charged, If cars in both parking lot1 and lot2 >10, additional $8 are charged.

The policy become non-linear because of the modification on reward, the major difference is that in previous policy, there is no action needed when parking number around 10. But since Jack need to pay for extra money if a location have>10 cars, there are actions needed when the number of cars in a parking lot is close to 10 to avoid penalty.