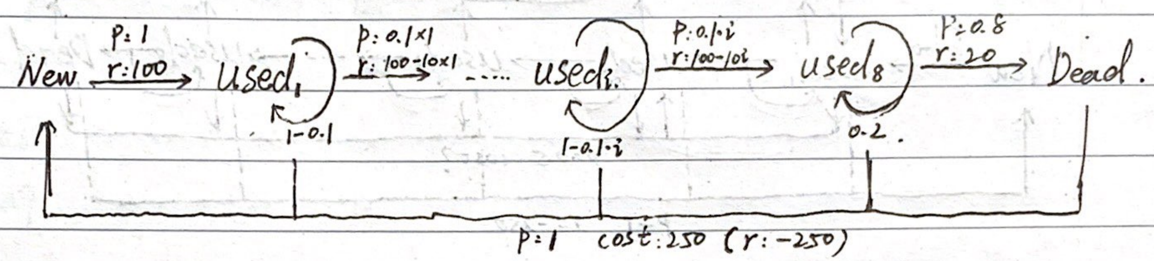
Question 2. Markov Decision Process.

~~The~~ According to the description of the problem, we can transform it to the flow graphy below:



[p represents possibility and r represents reward]

(a)
(b) ⟶ According to the flow graphy above, now we have ~~then~~ update prosses as below given random initial utility: $U(New) = U(used_i) = U(Dead) = 0$

$U_{n+1}(New) = 100 + \beta U_n(U_1)$

for $i = 1$ to 7:

$U_{n+1}(used_i) = \max\left[\left(100 - 10\cdot i + \beta\left((1-0.1\times i)\cdot U_n(used_i) + 0.1\times i\cdot U_n(used_{i+1})\right)\right), -250 + \beta U_n(new)\right]$

$U_{n+1}(used_8) = \max\left[\cancel{100}\, 20 + \beta\left(0.2\cdot U_n(used_8) + 0.8\cdot U_n(Dead)\right), -250 + \beta U_n(new)\right]$

$U_{n+1}(Dead) = -250 + \beta U_n(New)$

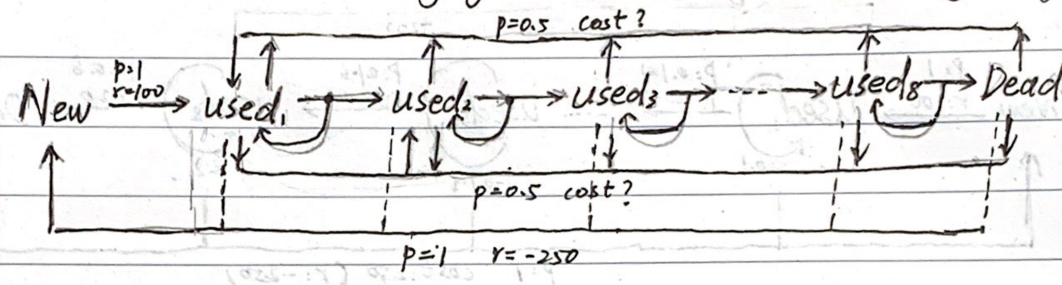
follow this process until results converge:

We define 'converge' like this: when $\max[U_{n+1}$ of all states $- U_n$ of all state$] < 0.001$

for question (a)(b)

I write a small code to solve this, it ~~stops~~ converges at around 120~140 times of iterations. The final utility and optimal policy are shown in the ~~log.txt!~~ log_ab.txt. As we can see, when it converges, the policy for 'New' and 'used₁'–'used₅' is 'use', and the policy for 'used₆'–'used₈' and 'Dead' is 'replace'.

(c) The option to buy a ~~new~~ used_machine instead of buying a new-machine is changing the flow graph in following way:



According to this graphy, we now have a new update process.

$U_{n+1}(New) = 100 + \beta U_n(u_1)$

for $i=1$ to 7: $U_{n+1}(u_i) = \max\{(100-10i) + \beta((100-0.1i)U_n(u_i) + 0.1i \cdot U_n(u_{i+1})), -250 + \beta U_n(new),$

$-cost + \beta(0.5U_n(u_1) + 0.5U_n(u_2))\}$

$U_{n+1}(u_8) = \max\{(100-10i) + \beta(0.2\,U_n(u_8) + 0.8U_n(Dead)), -250 + \beta U_n(New),$

$-cost + \beta(0.5U_n(u_1) + 0.5U_n(u_2))\}$

$U_{n+1}(Dead) = \max\{-250 + \beta U_n(New), -cost + \beta(0.5U_n(u_1) + 0.5U_n(u_2))\}$

We constantly change the cost of buying a used_machine, and we find that the threshold is 169-170. When the cost changes from 169 to 170, the policy for $u_6/u_7/u_8$/Dead changes from 'Buying used_machine' to 'Buying new-machine'. So the highest price shoud be set to 169 for which used-machine is the rational choice. (More data detail in log_c.txt)

(d) we change the β and record the policy when flow converges. And Below is the table of policy.

β	New	used1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	Dead.
0.1~0.8	u	u	u	u	u	u	u	u	u	re
0.85	u	u	u	u	u	u	u	re	re	re
0.9	u	u	u	u	u	u	re	re	re	re
0.93	u	u	u	u	u	re	re	re	re	re.
0.95	u	u	u	u	u	re	re	re	re	re.
$\boxed{0.96\sim}$	u	u	u	u	re	re	re	re	re	re.

[u:'use' . re:'replace']

I ~~base~~ do tests on the base of problem (a) & (b). During the test, we can find that after β gets bigger than 0.96, the policy becomes stable, and the best policy is shown above.

A little thought here: When β is small, it pays more attention to current reward, so each state takes 'use' as optimal policy, for it imediately get rewards. When β is bigger, it gradually takes future ~~[illegible]~~ utility into account. When β is big enough, the future utility compensate for the current reward, and the optimal policy turns from 'use' to 'replace'

Bonus. When the cost of a new machine is 250, ~~as~~ the long term discounted value is 800.53 as we computed in part (a) & (b)

We continue raising the cost of new machine, And we ~~find~~ compute the corresponding utility of new machine. We find that around ~~749~~, cost of 749, the flow reaches 0 gain (the utility of new machine $\approx$ the cost of a new machine). Below that cost, we are operating at a net gain. Above that cost, we are operating at a net loss.