1.(a) $h_1(S) = 6$, $h_1(a) = 5$, $h_1(b) = 5$,
$h_1(c) = 4$, $h_1(d) < 3$, $h_1(e) = 2$, $h_1(G) = 0$

Thus, h_1 is admissible.

$h_2(S) = 6$, $h_2(a) = 5$, $h_2(b) < 5$
$h_2(c) < 4$, $h_2(d) < 3$, $h_2(e) < 2$, $h_2(G) = 0$

Thus, h_2 is admissible.

$h_3(S) < 6$, $h_3(a) < 5$, $h_3(b) < 5$,
$h_3(c) < 4$, $h_3(d) < 3$, $h_3(e) < 2$, $h_3(G) = 0$

Thus, h_3 is admissible.

$h_4(S) > 6$ (violation), $h_4(a) > 5$ (violation)
$h_4(b) > 5$ (violation), $h_4(c) < 4$,
$h_4(d) = 3$, $h_4(e) = 2$, $h_4(G) = 0$.

Thus, h_4 is not admissible.

(b) h_1: $S-a = 1$, $a-b < 1$, $a-d < 5$, $a-e = 3$
$b-c = 1$, $c-e = 2$, $d-e < 1$, $e-G = 2$

Thus, h_1 is consistent.

h_2: $S-a = 1$, $a-b = 1$, $a-d < 5$, $a-e > 3$ (violation)
$b-c > 1$ (violation), $c-e < 2$, $d-e < 1$, $e-G < 2$

Thus, h_2 is not consistent.

h_3: $S-a < 1$, $a-b = 1$, $a-d < 5$, $a-e = 3$,
$b-c > 1$ (violation), $c-e < 2$, $d-e < 1$, $e-G < 2$

Thus, h_3 is not consistent.

h_4: $S-a = 1$, $a-b < 1$, $a-d < 5$, $a-e > 3$ (violation)
$b-c > 1$ (violation), $c-e < 2$, $d-e = 1$, $e-G = 2$

Thus, h_4 is not consistent.

(c) OPEN: (S, 6)
(h_1) (a, 6)
(e, 6), (b, 7), (d, 8)
(G, 6), (b, 7), (d, 8)

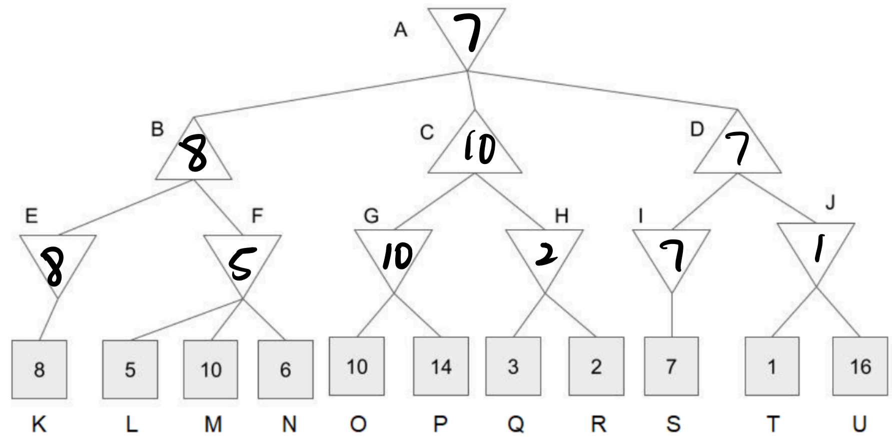
CLOSED: (S, 6), (a, 6), (e, 6), (G, 6)

(d) In each expansion step, we may add new nodes or update the value of the existed nodes. Thus, if the goal state node is not at the beginning of the open list, it may not be the smallest value and may be updated in the next expansion process. For example, in our lecture slide, we reached [r, 20] and updated it to [r, 14] in the next step. Then, we should keep [r, 14] not [r, 20].

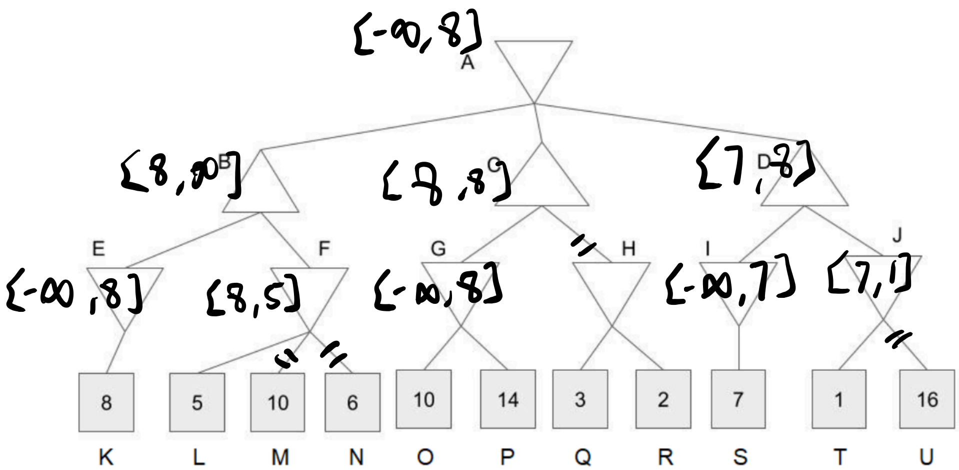
If the heuristic is not guaranteed to be admissable, we should expand all nodes making multiple goal state nodes with different values. Then we should choose the one with smallest value.

(e) let $h_3(c) = 2$, then h_3 is admissable and consistent. Then h_1 dominates h_3.
The special thing is $|h_1(S_i) - h_1(S_j)| \geq |h_3(S_i) - h_3(S_j)|$

2. (a)



(b)



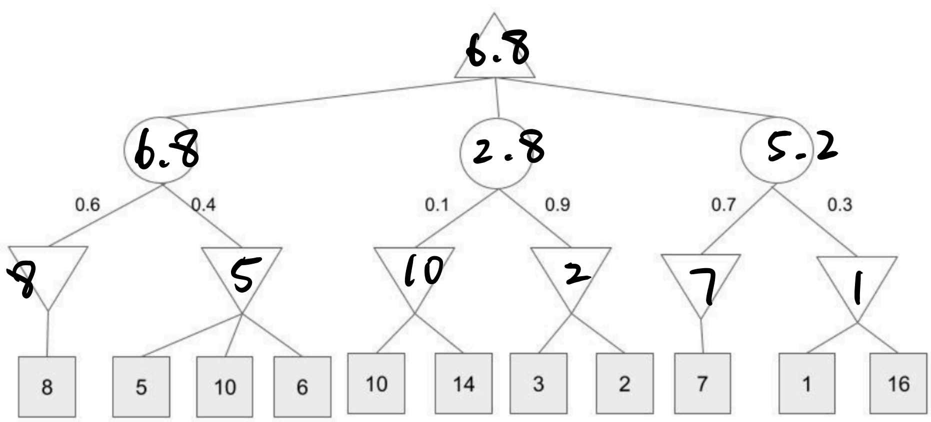
K	L	M	N	O	P	Q	R	S	T	U
		X	X			X	X			X

(c)

K	L	M	N	O	P	Q	R	S	T	U
X	X	X	X							

(d) The left-to-right ordering is more efficient. It prunes more.

(e)



3.(a) $T(s, a, s')$ means that when we are in state s and take an action a, we will have the probability of T to reach state s'. Since T is probability, $0 \le T \le 1$.

(b) NO (c) False

(d) The sooner we get a reward, the better. Thus we need a factor to decrease the untility of later rewards.

Besides, we need to determine when the algorithm terminate, and discount factor helps converge.

(e) $V = R(S_{now}) + \gamma R(S_{final})$

$= 0 + 0.9 \times 100 = 90$

(f) $V = R(S_{now}) + \gamma R(S_{1_2}) + \gamma^2 R(S_{final})$

$= 0 + 0.9 \times 30 + 0.9^2 \times 100$

$= 27 + 81 = 108$

(g) Using $\gamma = 0.8$, we get $V = 0 + 30 \times (0.8 + 0.8^2 + \cdots + 0.8^n) + 100 \times 0.8^{n+1}$

When the increase is smaller than , we think it converges. Then we get it should be 10 extra work days.

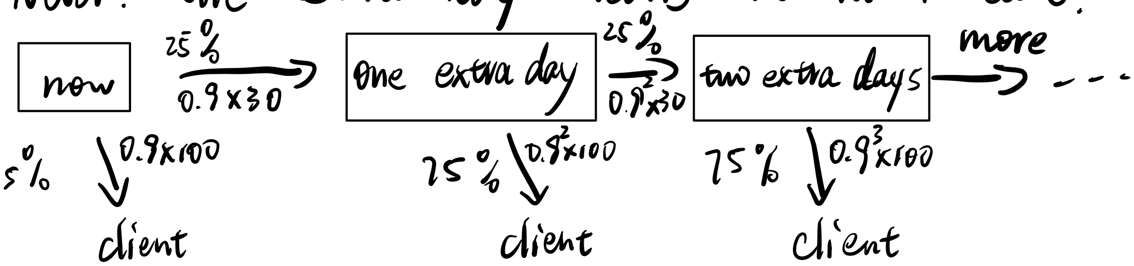
```
result = [24]
for i in range(1, 100):
    result.append(30*0.8**(i+1) + result[i-1])
for i in range(len(result)):
    result[i] = result[i] + 100*0.8**(i+2)
for i in range(len(result)):
    if result[i+1] - result[i] <= 1:
        print(i)
        print(result[i])
        print(result[i+1])
        break
```

```
9
115.70503270400002
116.56402616320001
```

(h) Send the art right now.

(i) Never. One extra day means 30 hr increase.

(j)



(k) $V(s) = 0 + 0.9 \times [0.75 \times 100 + 0.25 \times (30 + 0.9 \times 100)]$

$= 0.9 \times [75 + 30]$

$= 94.5$

4.(a) $V_{k+1}(S) \leftarrow \max \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma V_k(s')]$

Thus $V_0(S_1) = 0$

$V_1(S_1) = \max\{1 \times 100, 1 \times 30\} = 100$

$V_1(S_2) = \max\{0.1 \times 50 + 0.9 \times 100, 0.2 \times 0 + 0.8 \times 80\}$

$= \max\{95, 64\} = 95$

$V_1(S_3) = \max\{1 \times 20, 0.4 \times 20 + 0.3 \times 0 + 0.2 \times 0 + 0.1 \times 0\}$

$= \max\{20, 8\} = 20$

$V_1(S_4) = \max\{0.6 \times 20 + 0.4 \times 40, 1 \times 10\}$

$= \max\{28, 10\} = 28$

Then

	S_1	S_2	S_3	S_4
V_0	0	0	0	0
V_1	100	95	20	28

$V_2(S_1) = \max\{1 \times (100 + 0.9 \times 20), 1 \times (30 + 0.9 \times 95)\}$

$= \max\{118, 115.5\} = 118$

$V_2(S_2) = \max\{0.1 \times (50 + 0.9 \times 95) + 0.9 \times (100 + 0.9 \times 20),$

$0.2 \times (0 + 0.9 \times 95) + 0.8 \times (80 + 0.9 \times 28)\}$

$= \max\{119.75, 101.26\} = 119.75$

$V_2(S_3) = \max\{1 \times (20 + 0.9 \times 20), 0.4 \times (20 + 0.9 \times 20) +$

$0.3 \times (0 + 0.9 \times 28) + 0.2 \times (0 + 0.9 \times 95) +$

$0.1 \times (0 + 0.9 \times 100)\}$

$= \max\{38, 48.86\} = 48.86$

$V_2(S_4) = \max\{0.6 \times (20 + 0.9 \times 28) + 0.4 \times (40 + 0.9 \times 20),$

$1 \times (10 + 0.9 \times 28)\}$

$= \max\{50.32, 35.2\} = 50.32$

(b) $Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$

Then $Q_3(s,a) \leftarrow \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_2(s')]$

Thus $Q_3(S_1,W) = 1 \times [100 + 0.9 \times V_2(S_3)]$

$= 1 \times (100 + 0.9 \times 48.86) = 143.974$

$Q_3(S_2,W) = 0.1 \times [50 + 0.9 \times V_2(S_2)] +$

$0.9 \times [100 + 0.9 \times V_2(S_3)]$

$= 0.1 \times (50 + 0.9 \times 119.75) +$

$0.9 \times (100 + 0.9 \times 48.86)$

$= 145.3541$

$Q_3(S_3,W) = 1 \times [20 + 0.9 \times V_2(S_3)]$

$= 1 \times (20 + 0.9 \times 48.86)$

$= 63.974$

$Q_3(S_4,W) = 0.6 \times [20 + 0.9 \times V_2(S_4)] +$

$0.4 \times [40 + 0.9 \times V_2(S_3)]$

$= 0.6 \times (20 + 0.9 \times 50.32) +$

$0.4 \times (40 + 0.9 \times 48.86)$

$= 72.7624$