3.1

(a) suppose, $P(X_t=j \mid X_1=i) = [U^{t-1}]_{ij}$ is true for $t \geq 2$

Then:

$$P(X_{t+1}=j \mid X_1=i) = \sum_k P(X_{t+1}=j \mid X_t=k, X_1=i) \cdot P(X_t=k \mid X_1=i)$$
$$= \sum_k P(X_{t+1}=j \mid X_t=k) \cdot P(X_t=k \mid X_1=i)$$
$$= \sum_k U_{kj} \cdot [U^{t-1}]_{ik}$$
$$= \sum_k [U^{t-1}]_{ik} U_{kj} = [U^t]_{ij}$$

And when $t=1$

$P(X_2=j \mid X_1=i) = U_{12}$, ∴ also true

Hence,

$P(X_{t+1}=j \mid X_1=i) = [U^t]_{ij}$ is true.

(b) from the part (a), we could see:

$$[U^t]_{ij} = \sum_k [U^{t-1}]_{ik} U_{kj},$$

which means, in fact, we only need the jth column of U_{kj} to compute. And for $[U^1]$ to $[U^t]$, we need do the multiplication t times.

∴ Time complexity is $O(n^2 t)$

Algorithm:

```
nx1 matrix: col = Uj; // the jth colum of U
for k= 1:t
    col= U.col
end for loop
return col
```

(c). when we see $\log_2 t$, we think of use the matrix U^t, where t is the power of 2.

Hence, we just do:

$$A^t = (A^{\frac{t}{2}})^2 = ((A^{\frac{t}{4}})^2)^2 \cdots$$

And for any matrix $n \times n$ multiplication, the time complexity is $O(n^3)$

$\therefore$ the time complexity in our case is $O(n^3 \log_2 t)$

Algorithm:

```
Assume m is a identity matrix
while (t > 0)
    if t % 2 == 1
        m = m * U
    U = U * U
    t = t / 2;
end while
return m;
```

(d). Recall $[U^t]_{ij} = \sum [U^{t-1}]_{ik} U_{kj}$, again we only need to know jth column of U. In our case, since $m \ll n$, we could open an list to store the non-zero elements.

Hence, we just need muliply the matrix with the list.

$\therefore$ The time complexity is $O(mnt)$