In [1]:
```python
import numpy as np
```

Exercise 3. Implement Algorithm 3.17 from the book. The input parameters are the integers M, L, and a set of weights $w_i \geq 0$, $i = 1, \ldots, M$, with $\sum_{i=1}^{M} w_i = 1$. The output of the algorithm are M integers $\bar{\xi}_i \geq 0$ which satisfy $\sum_{i=1}^{M} \bar{\xi}_i = L$. Verify your algorithm by checking that $\bar{\xi}_i / L \approx w_i$ for $L \gg M$.

Algorithm 3.27 (Multinomial samples) The integer-valued variable $\bar{\xi}_i$, $i = 1, \ldots, M$, is set equal to zero initially.

For $l = 1, \ldots, L$ do:

(i) Draw a number $u \in [0, 1]$ from the uniform distribution U[0, 1].

(ii) Determine the integer $i^* \in \{1, \ldots, M\}$ which satisfies

$$i^* = \arg\min_{i \geq 1} \sum_{j=1}^{i} w_j \geq u.$$

(iii) Increment $\bar{\xi}_{i^*}$ by one.

In [2]:
```python
def multinomialSamples (M, L, w_arr):
    eta_bar = np.zeros(M)
    # immediately drawing a list with uniform samples of size L
    u_list = np.random.uniform(low=0.0, high=1.0, size=L)

    for current in range(L):
        u = u_list[current]
        w_sum = w_arr[0]

        # determine i_star
        i_star = 0
        while w_sum < u:
            i_star += 1
            w_sum += w_arr[i_star]

        # increment eta_bar[i_star] to get the right distribution
        eta_bar[i_star] += 1
    return eta_bar
```

In [3]:
```python
L = 10000
w_arr = np.asarray([0.2, 0.1, 0.05, 0.25, 0.33, 0.07])
M = len(w_arr)

eta_bar = multinomialSamples(M, L, w_arr)

print("Compare the two arrays for verification:")
print(w_arr)
print(eta_bar/L)
```

```
Compare the two arrays for verification:
[0.2  0.1  0.05 0.25 0.33 0.07]
[0.2061 0.0974 0.0453 0.2488 0.3318 0.0706]
```