1 Part 1

We have two states :Old box, New box.
The system is in "New box" when we have exactly one item in the current box that is being filled. The system is in "Old box" when the current box contains more than one item.
X_n is the random variable for the weight of the current box at time n.
W_n is the weight of incoming item at time n.

The transition probabilities are as follows:
P_{NN} is the probability of going from state "New box" to state "New box". This happens when the weight of the new box (that has a single item) plus the incoming item exceeds the maximum allowed. Hence, we have a transition from one new box to another new box.
P_{ON} is the probability of going from state "Old box" to state "New box". This happens when the weight of the current box plus the new item exceeds the maximum weight. Hence, we have a transition to a new box to accommodate the newest item.
P_{NO} is the probability of going from state "New box" to state "Old box". This happens when the new box is able to accept the newest item and thereby becoming an "Old box".
P_{OO} is the probability of going from state "Old box" to state "Old box". This happens when our current box (with 2 or more items) can accommodate the newest item while maintaining maximum weight. Therefore, there is not a need for a new box.

Mathematically, by conditioning on the item at time n we have:

$$\begin{aligned} P_{NN} = P_{ON} &= P(X_n > w_{max} - W_n) = \sum_{k=1}^{w_{max}} P(X_n > w_{max} - W_n | W_n = k) \cdot P(W_n = k) \\ P_{OO} = P_{NO} &= P(X_n \leq w_{max} - W_n) = \sum_{k=1}^{w_{max}} P(X_n \leq w_{max} - W_n | W_n = k) \cdot P(W_n = k) \end{aligned}$$

Now, $P(X_n \leq w_{max} - W_n | W_n = k)$ is the probability that the weight of the current box is larger than $w_{max} - k$. That is, $P(X_n \leq w_{max} - X_n | W_n = k) = P(X_n \leq w_{max} - k)$. In the limit as n approaches ∞:

$$P(X_n \leq w_{max} - k) = \sum_{i \leq k} \pi_i$$

In other words, the probability that the weight of the current box is less than or equal to $w_{max} - k$ is the sum of the stationary probabilities $P(W = w_1), P(W = w_2), ..., P(W = w_k)$. Similarly, we have

$$P(X_n > w_{max} - k) = \sum_{i > w_{max} - k} \pi_i$$

Finally, the long run average number of items in any box is the average time for a system in state N to return to state N. This is $E(T_{NN})$, where T is the random variable for hitting times.
To find $E(T_{NN})$, we follow the similar argument in the book.
First, we condition on the first stop U.

$$E(T_{NN}) = P_{N,O} \cdot E(T_{NN} | U = O) + P_{N,N} \cdot E(T_{NN} | U = N)$$

where,

$$\begin{aligned} E(T_{NN} | U = N) &= 1 \\ E(T_{NN} | U = O) &= 1 + E(T_{ON}) \end{aligned}$$

Simplifying the above, we arrive at

$$E(T_{NN}) = 1 + P_{NO} \cdot E(T_{ON})$$

By repeating the same argument for $E(T_{ON}$, we find

$$E(T_{ON}) = 1 + P_{OO} \cdot E(T_{ON})$$

Hence,

$$E(T_{ON}) = \frac{1}{1 - P_{OO}}$$

Putting this back in our previous expression of $E(T_{NN})$

$$E(T_{NN}) = 1 + \frac{P_{NO}}{1 - P_{OO}}$$

2 Part 2

3 Part 3

Looking at the system differently, we can model using a different Markov chain:
The states are $S_1, S_2, S_3, ..., S_m$, where m is w_{max} and S_i represents the state where w_i is the first item in the current box.

The transition probability P_{ij} of going from a state i to a state $j \neq i$ is equalled to the probability of needing a new box while in state i (first item is w_i) to store a new item weighing w_j (the first item of the new box - state j). This probability is:

$$P_{ij} = P(X_n + w_k > w_{max}) \cdot P(W_n = w_k) = \sum_{i > w_{max} - k} \pi_i \cdot P(W_n = w_k)$$

To find P_{ii}, we need to consider two independent cases when state i loops back on itself. First, the system remains in a state i if the next item does not cause it to exceed the maximum weight. This probability is:

$$\begin{aligned} P(X_n + W_i \leq w_{max}) &= P_{NN} \\ &= \sum_{k=1}^{w_{max}} P(X_n > w_{max} - W_n | W_n = k) \cdot P(W_n = k) \\ &= \sum_{k=1}^{w_{max}} \left(\sum_{i > w_{max} - k} \pi_i \right) \cdot P(W_n = k) \end{aligned}$$

In the second case, the system remains in state i if a 'w_i item' arrives which would push the current weight over the limit. This probability is:

$$P(X_n + w_i > w_{max}) \cdot P(W_n = w_i) = \sum_{i > w_{max} - w_i} \pi_i \cdot P(W_n = w_i)$$

Knowing the transition matrix, we can now solve the eigenvalue equation to get the stationary probabilities. π_i in this case gives the percentage of time the system spends in state i - that is the percentage of time that the -uh oh guys, as I'm typing this I think this is wrong!!!!!!!!!!!!!

4 Part 4

Given $W_{max} = 10$ and P(W = i) = ci/10, i = 1,2,...,10 with the sum equal to 1 you get c = 10/55 ≈ 0.18. Running the following code for the values Q4(TRUE,FALSE,FALSE), Q4(TRUE,TRUE,FALSE), and Q4(FALSE,FALSE,TRUE).

```
Q4 <- function(testQ1 = FALSE, testQ2 = FALSE, testQ3 = FALSE)
{
        vmax = 10
        values = c(1:vmax)
        prob = c(1:vmax)
        values_in_new_box = c(rep(0,length(values)))
        constant = vmax/sum(values)
        prob = constant*values/vmax
        current_weight = 0
        number_of_boxes = 1
        boxes_total_weight = 0
        items = 100000
        for(i in 1:items)
        {
                random_value = sample(values, 1, TRUE, prob)
                boxes_total_weight = boxes_total_weight + random_value

                current_weight = current_weight + random_value
                if( current_weight > vmax )
                {
                        number_of_boxes = number_of_boxes + 1
                        current_weight = random_value
                        values_in_new_box[random_value] = values_in_new_box[random_
                }
        }

        number_of_items_per_box = items/number_of_boxes
        average_weight_per_box = boxes_total_weight/number_of_boxes

        values_in_new_box = values_in_new_box/number_of_boxes

        if(testQ1 == TRUE)
        {
                return(number_of_items_per_box)
        }
        if(testQ2 == TRUE)
        {
                return(average_weight_per_box)
        }
        if(testQ3 == TRUE)
        {
                return(values_in_new_box)
        }
}
```

This code yields the following simulated values for 100,000 samples. For the average number of items per box we get 1.132131. For the average weight per box we get 7.933018. For the probability of each weight starting a box we get the following table.

i	1	2	3	4	5
P(Q = i)	0.004532167	0.017698112	0.035645494	0.059699971	0.083403204
i	6	7	8	9	10
P(Q = i)	0.111683927	0.136916767	0.160178114	0.184130617	0.206100297

Here is where we compare these results to our numerical answers from Q1, Q2, and Q3.