

Appendix A

Alias method

The methods to sample from a discrete probability for n events, presented in Algorithms 1 and 2, become inefficient when n is large. To devise an efficient algorithm for large n , we must distinguish the case where the list of probabilities is constantly changing from the case where it remains unchanged. In the latter case, there are several ways to boost efficiency by performing a modestly expensive operation once and then using more efficient operations in subsequent samplings. One such approach is to sort the list of probabilities, an operation generally requiring $\mathcal{O}(n \log n)$ operations, and use a bisection method, requiring $\mathcal{O}(\log n)$ operations, to select the events from the list. What we really want, however, is a method requiring only $\mathcal{O}(1)$ operations. Walker's alias algorithm (Walker, 1977) has this property.

Suppose we need to repeatedly choose one of five colors randomly – red, blue, yellow, pink, and green – with weights 6, 1, 3, 2, and 8 (Fig. A.1). We first generate five sticks with lengths 6, 1, 3, 2, and 8 and paint each stick with the color that it represents. Next, we define a linked list of the sticks that are longer than the average (which is 4), and another for the sticks that are shorter than the average. In the present case, the long-stick list is (“red” \rightarrow “green”), and the short-stick list is (“blue” \rightarrow “yellow” \rightarrow “pink”). We pick the first item from each list and cut the longer (red) stick in two pieces in such a way that if we join one of the two to the shorter (blue) stick, we obtain an average-length stick. As a result, we are left with a red stick of length 3 and a joint stick of length 4. Since the original red stick was made shorter than the average length, we remove it from the long-stick list and append it to the short-stick list. On the other hand, the original blue stick has become an average-length stick, so we remove it from the short-stick list. Then, we pick the first item from each list and repeat the same operations again and again. When finished, we have five sticks of average length, some with a single color and others with two colors.

At each step at least one of the lists is shortened by one. Because the length of the list is finite, the whole procedure must come to an end within a finite number

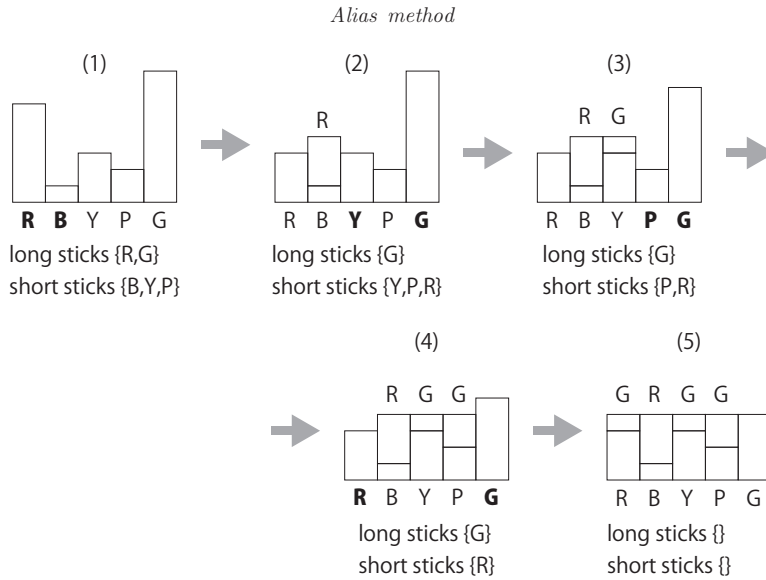


Figure A.1 Constructing aliases in Walker's method.

of operations. While constructing the joint stick-lists takes $\mathcal{O}(n)$ operations, this scaling does not matter in most cases, because once we have constructed the lists we can use them many times with little additional cost.

Now, to use these lists efficiently in the present case, we generate a random number $r \in [0, 20]$ (20 is the sum of the weights). Suppose it is 9.6. Instead of accumulating and comparing as in Algorithm 1, we simply divide r by the average stick length, that is, 4. This number means that our tentative choice is the third stick. But this time “third” does not necessarily mean “yellow.” It can be an alias of “green.” In the present case, the third stick, which was originally yellow, is now a composite of a yellow piece of length 3 and a green piece of length 1 so we have to choose among these two colors with probabilities $3/4$ and $1/4$. What is important is that this operation is $\mathcal{O}(1)$.