CIS 575. Introduction to Algorithm Analysis
Material for February 14, 2024

Dutch National Flag Algorithm: the Problem

1 Problem Statement

Input An array of items, each having exactly one of the properties **red**, **white**, or **blue** (in addition to other properties)

Output A permutation of the items such that all red items precede all white items, which in turn precede all blue items.

(In addition, we return the number of red items, the number of white items, and the number of blue items.)

Graphically, the desired effect is to rearrange the items so that they form (except there could be many more items of one color than of another color) the *Dutch National Flag*:

Given that many other flags in the world are formed by 3 stripes, perhaps most famously the French, one may suggest that *Tricolore* would be a more apt name for this problem. And indeed, that could very well have been the chosen term, had the problem not been proposed by a famous and tremendously influental Dutch computer scientist, Edsger Dijkstra.

2 Application

The algorithm we are about to develop may, as we shall see later in this course, be used in several contexts. For now, we shall consider the *selection* problem where we must find the

k'th smallest element of a given array; one approach to solving it is:

1. Choose, perhaps randomly, a so-called *pivot*, to be called p.
2. Apply the Dutch National Flag algorithm to partition the numbers into
   (a) those that are $< p$, considered **red**
   (b) those that are $= p$, considered **white**
   (c) those that are $> p$, considered **blue**.
3. Determine in which partition the k'th smallest element is, and do a suitable recursive call on that partition (unless it's the middle one in which case we know that p is the kth smallest element).

We shall illustrate this approach by an example: consider the 25 numbers:

37, 22, 42, 11, 17, 48, 12, 16, 20, 45, 61, 24, 47, 53, 33, 44, 35, 19, 10, 50, 13, 16, 30, 54, 23

and assume we want to find the 17th smallest element. Suppose that **23** has been picked as our pivot; then the Dutch National Flag algorithm will give us 3 partitions:

1. the 10 elements that are < 23
2. the 1 element that is $= 23$
3. the 14 elements that are > 23.

To find the 17th smallest element of the given numbers, we thus need to make a recursive call to find the 6th smallest element of the last partition.

3 Naive Approach

The Dutch National Flag problem may appear easy to solve:

1. compute the number r of red items, w of white items, b of blue items
2. create a new array, with the first r slots reserved for red items, the next w reserved for white items, and the last b reserved for blue items
3. traverse the original array, moving each item into the first available slot in the area reserved for its color.

While this solution accomplishes the desired rearrangement, even in time proportional to the number of items, it takes (due to step 2) also *space* proportional to the number of items, violating the

Extra Requirement The algorithm we develop must be *in-place*, with items rearranged only by *swapping*.

We shall next see how to accomplish that.