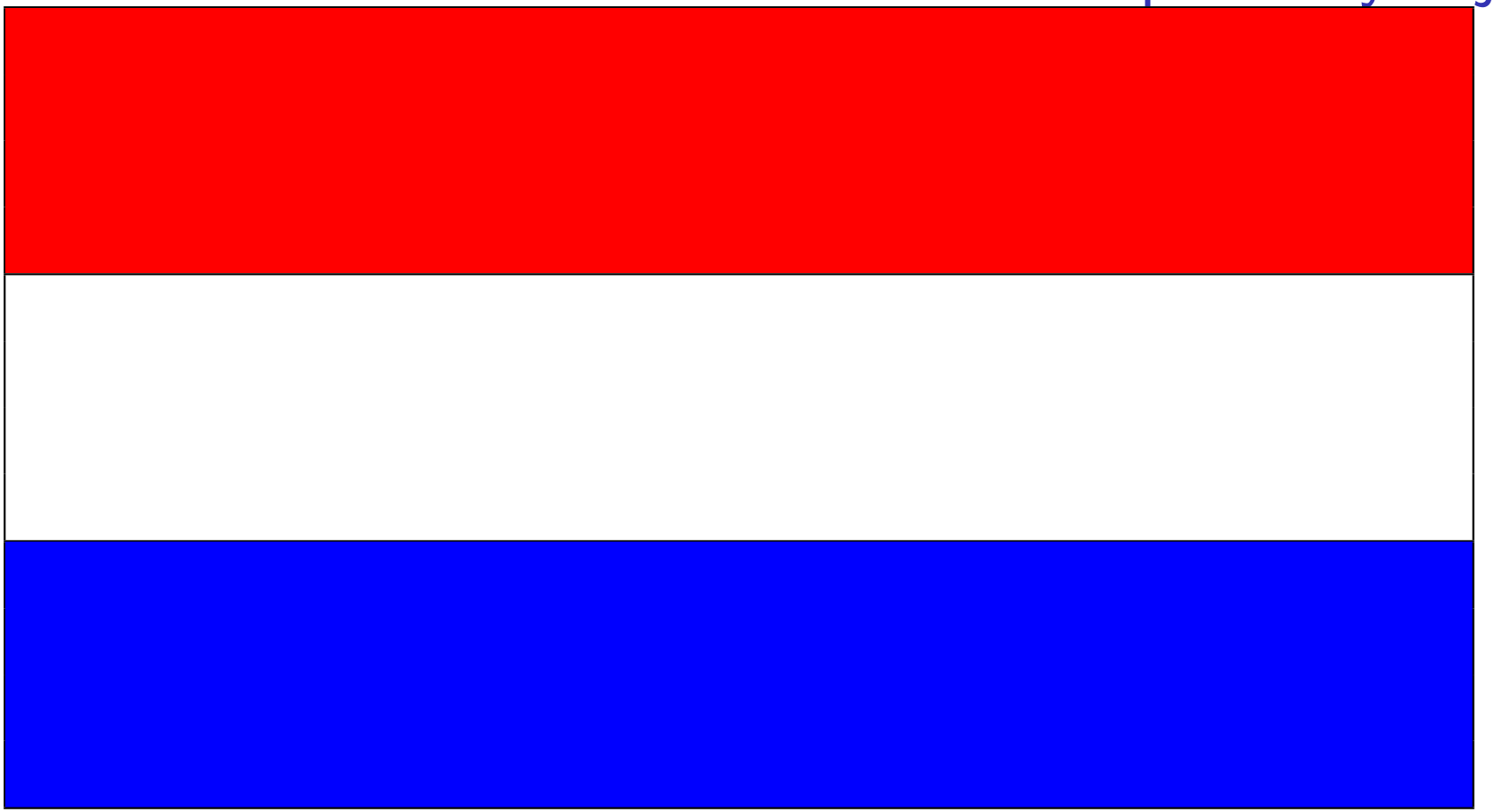
Understanding partitioning, which is central to Quick Sort: DUTCH NATIONAL FLAG PROBLEM posed by Dijkstra



Problem Statement: **Given an array of `N` objects (`a[1...N]`) coloured red, white or blue, sort them so that objects of the same colour are adjacent, with the colours in the order red, white and blue.**

Dutch National Flag (3 Color partioning) problem

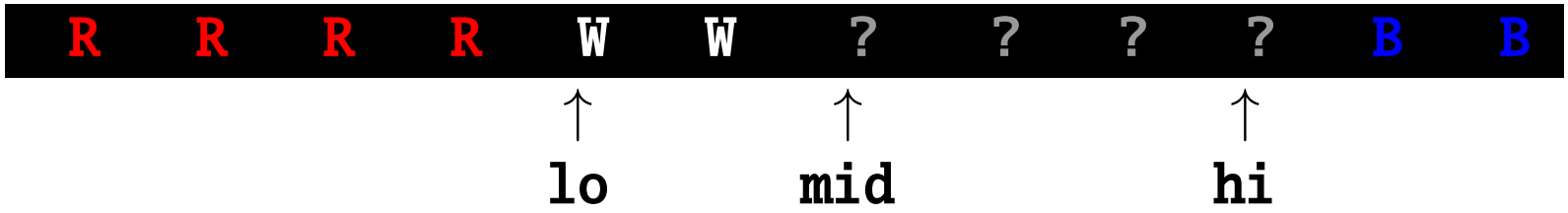
Problem Statement: **Given an array of `N` objects (`a[1...N]`) coloured red, white or blue, sort them so that objects of the same colour are adjacent, with the colours in the order red, white and blue.**

Conceptually, divide this array `a[1...N]` into **four sections**:

`a[1...Lo-1]` section groups the Reds ('R')
`a[Lo...Mid-1]` section groups the Whites ('W')
`a[Mid...Hi]` section groups the (YET) UNKNOWNS ('?')
`a[Hi+1...N]` section groups the blues ('B')

- The basic idea is to start with **`Lo=1, Mid=1, Hi=N`**
- Current the Unknown section is the whole array
- Shrink **Unknown section** while maintaining the conditions listed in the block above.

Dutch National Flag (3 Color partioning) problem – cont'd

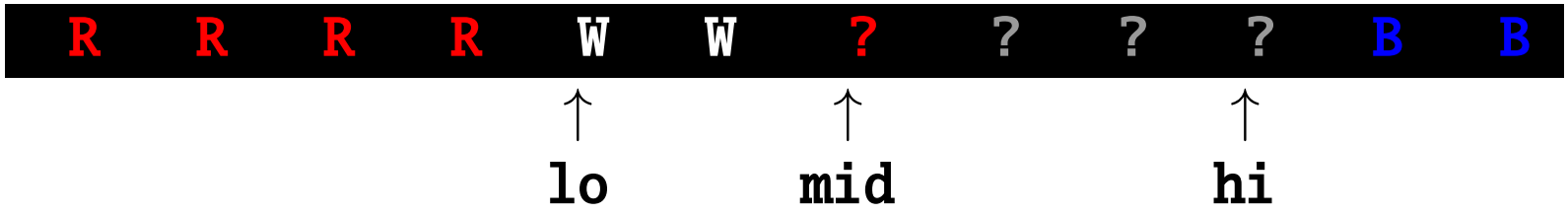


Three cases arise:

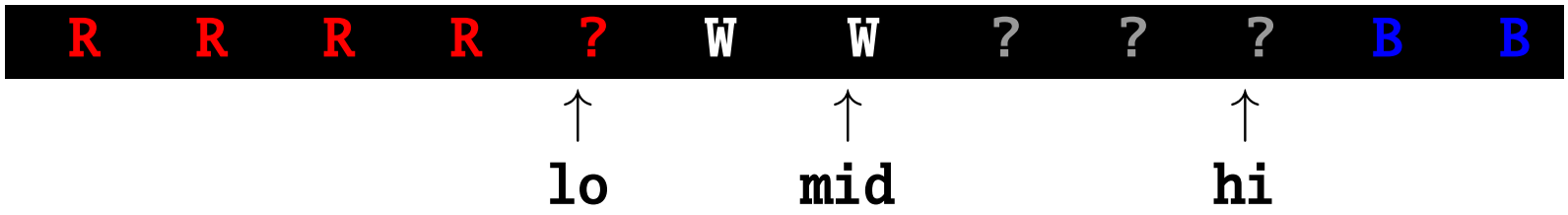
- **a[mid] is Red**
- **a[mid] is White**
- **a[mid] is Blue**

Dutch National Flag (3 Color partioning) problem – cont'd

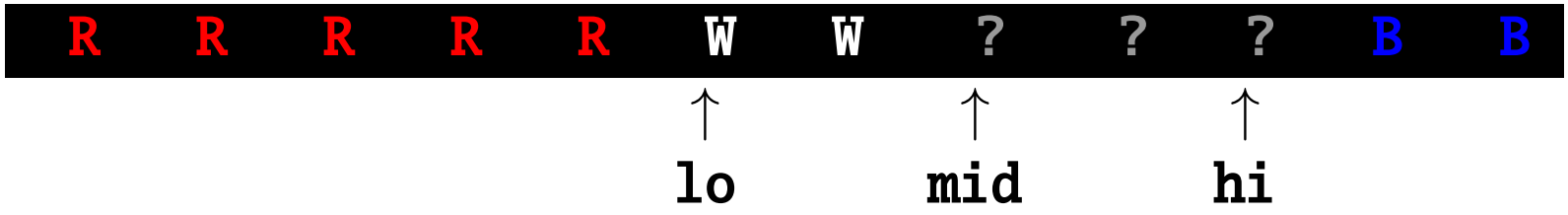
Case 1: If **a[mid]** is Red...



...**swap a[mid]** with **a[lo]**...

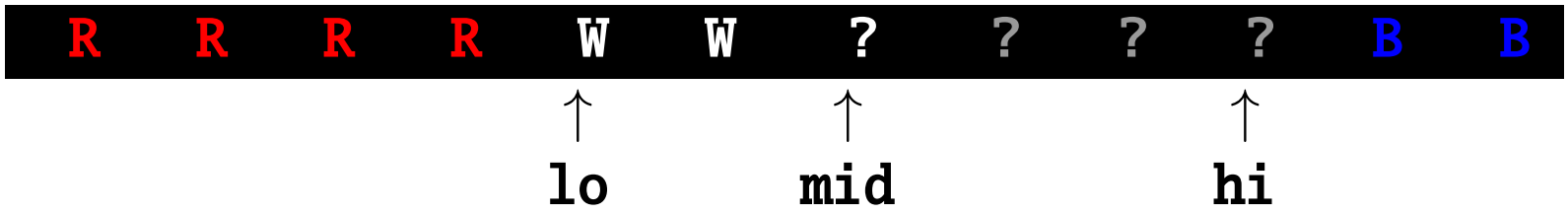


... and **increment by one lo** and **mid** pointers.



Dutch National Flag (3 Color partioning) problem – cont'd

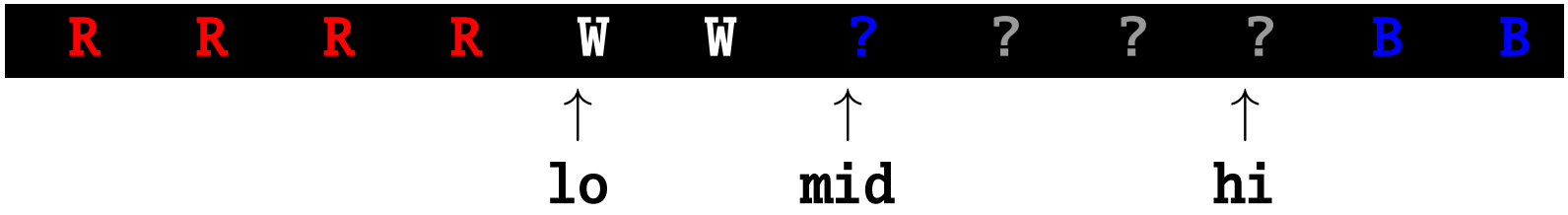
Case 2: If **a[mid]** is White...



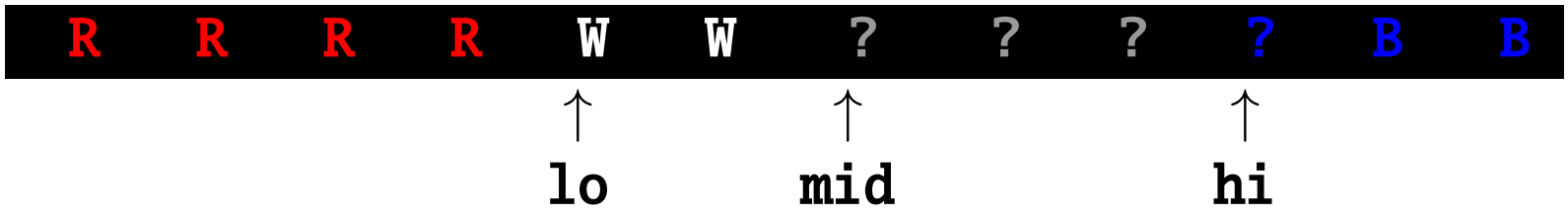
... simply **increment by one mid** pointer.

Dutch National Flag (3 Color partioning) problem – cont'd

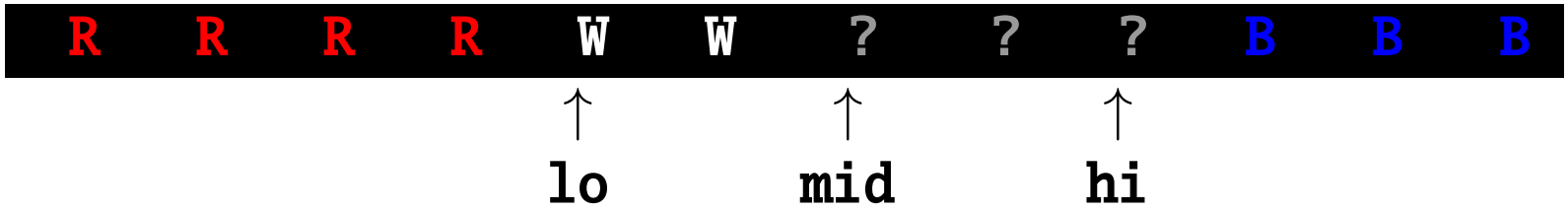
Case 3: If **a[mid]** is Blue...



...**swap a[mid]** with **a[hi]**...



... and **decrement by one hi** pointer.



Dutch National Flag (3 Color partioning) problem – Summary

```
1 Lo := 1; Mid := 1; Hi := N;
2 while Mid <= Hi do
3     //Invariance to be maintained in the loop:
4     // a[1..Lo-1]=R && a[Lo..Mid-1]=W && a[Hi+1..N]=B;
5     // a[Mid..Hi]=?
6     case a[Mid] in
7         0: swap a[Lo] and a[Mid]; Lo++; Mid++
8         1: Mid++
9         2: swap a[Mid] and a[Hi]; Hi--
```

Dutch National Flag problem is related to Quick Sort

- In quick sort, we partion the array (to be sorted) into **three** sections.
- To achieve this an *arbitrary* **`pivot`** is chosen from the array.
- Then **partition** this array using the **`pivot`** as follows:

(Left) Red	section with values smaller than the **`pivot`**
(Middle) White	section with values EQUAL to the **`pivot`**
(Right) Blue	section with values larger than the **`pivot`**

- Quick sort then **recursively sorts** the smaller and larger sections separately.

- Ideally, if **`pivot`** were chosen to be the **median** of the array, then the Red and Blue sections would be of **equal size**.
- However, median can only be found by scanning the whole array and this would slow the algorithm down.
- Therefore, in the **simplest versions** of **quick sort** an arbitrary element, typically the first element is used as **`pivot`**.