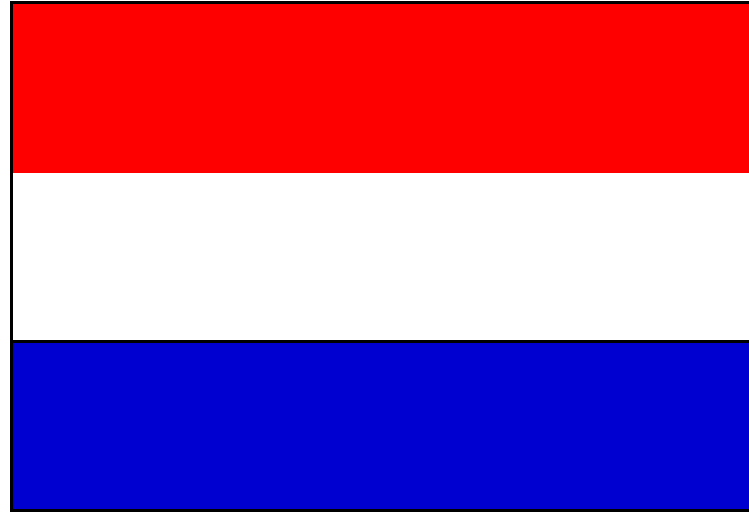


# Lecture 21 — Case Study: Dutch National Flag

David J. Pearce

*School of Engineering and Computer Science  
Victoria University of Wellington*

# Dutch National Flag Problem



- **Problem Statement:**

*“Given a quantity of items in three colours of the Dutch National flag, partition the items into three groups such that red items come first, then white items and, finally, blue items.”*

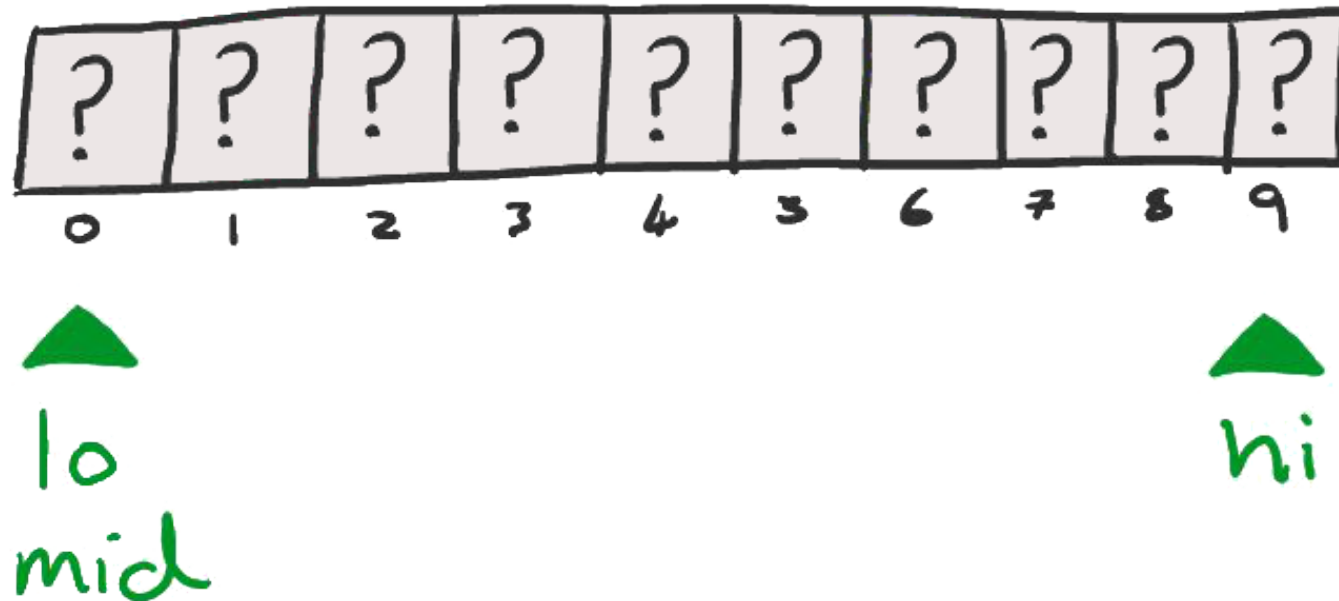
- Can we **specify** and **verify** an algorithm for doing this?

# Algorithm (Core)

```
function partition(int[] cols) -> (int[] r)
requires |cols| > 0:
    int lo = 0
    int mid = 0
    int hi = |cols|-1

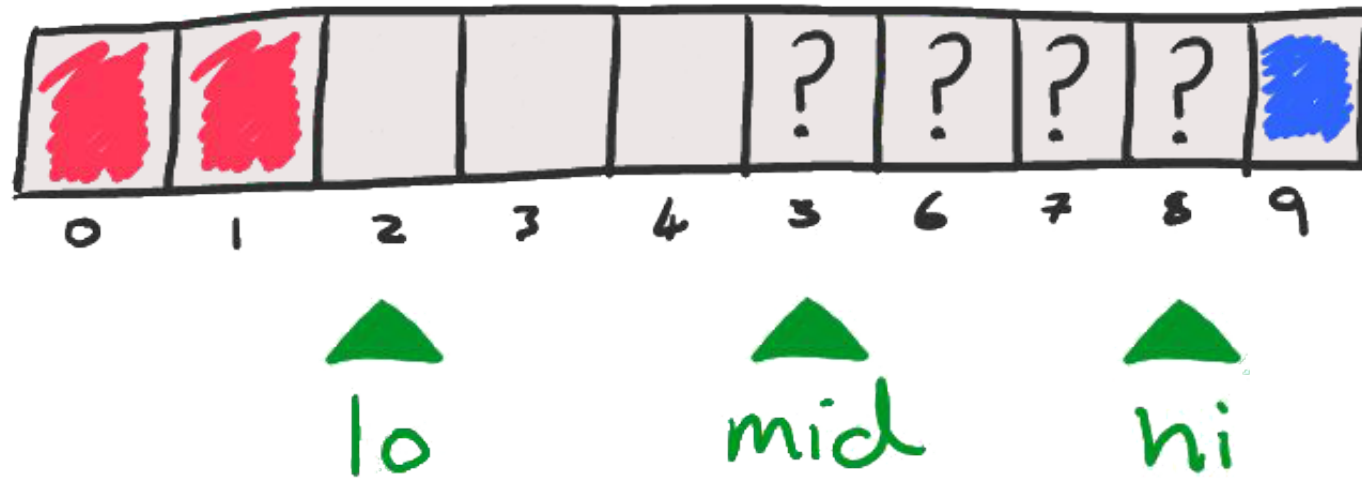
    while mid <= hi:
        //
        if cols[mid] == RED:
            int tmp = cols[lo]
            cols[lo] = cols[mid]
            cols[mid] = tmp
            lo = lo + 1
            mid = mid + 1
        else if cols[mid] == BLUE:
            int tmp = cols[mid]
            cols[mid] = cols[hi]
            cols[hi] = tmp
            hi = hi - 1
        else:
            mid = mid + 1
    //
    return cols
```

# Initial State



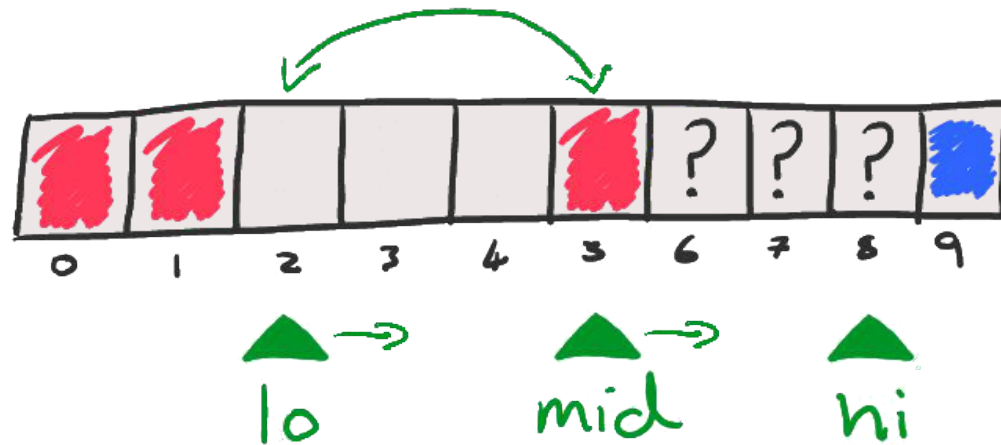
- Initially, all elements of array considered **unknown**
- **Three markers:** `lo`, `mid` and `hi`
- Markers `lo` and `mid` **start at zero**, `hi` starts at **last element**

# General State



- Marker `lo` — identifies **next position** for RED item
- Marker `mid` — identifies **next position** for WHITE item
- Marker `hi` — identifies **next position** for BLUE item

# Taking a Step



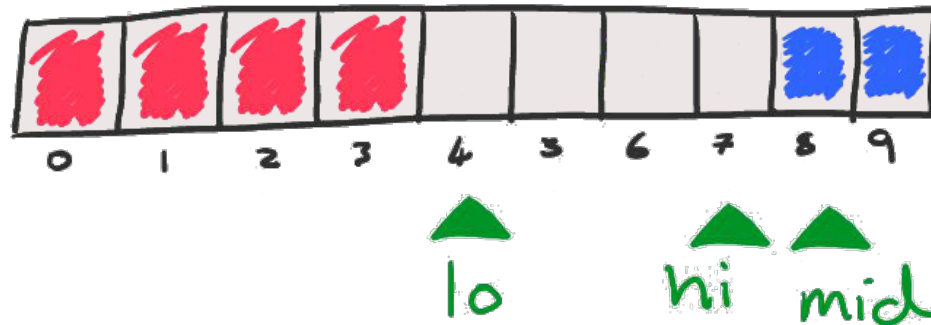
- To take a step, **examine** what is at `mid` position:

If **RED**, swap items at `lo` and `mid` and increment `lo` and `mid`

If **WHITE**, do nothing and increment `mid`

If **BLUE**, swap items at `mid` and `hi` and increment `mid` and decrement `hi`

# Terminating State



- To take a step, **examine** what is at `mid` position:

If **RED**, swap items at `lo` and `mid` and increment `lo` and `mid`

If **WHITE**, do nothing and `mid`

If **BLUE**, swap items at `mid` and `hi` and increment `mid` and decrement `mid`

# Dutch Flag Problem

```
while mid <= hi:  
    //  
    if cols[mid] == RED:  
        int tmp = cols[lo]  
        cols[lo] = cols[mid]  
        cols[mid] = tmp  
        lo = lo + 1  
        mid = mid + 1  
    else if cols[mid] == BLUE:  
        int tmp = cols[mid]  
        cols[mid] = cols[hi]  
        cols[hi] = tmp  
        hi = hi - 1  
    else:  
        mid = mid + 1
```

- What is the **loop invariant** here?