$3n$ is $O(n)$

$\exists c > 0, k \geq 1.\ \forall n \geq k.\ 3n \leq cn$

$\exists c > 0.\ \forall n \geq 1.\ 3n \leq cn$

$\boxed{\forall n \geq 1.}\ 3n \leq 3n$

$n \leq n$

$3n \leq 3n$

$\forall n \geq 1.\ 3n \leq 3n$

$\exists k \geq 1.\ \forall n \geq k.\ 3n \leq 3n$

$\exists c > 0, k \geq 1.\ \forall n \geq k.\ 3n \leq cn$

$\Rightarrow 3n$ is $O(n)$

$\lim_{x \to \infty} \frac{f(x)}{g(x)} = c$ for constant c

Quick Sort

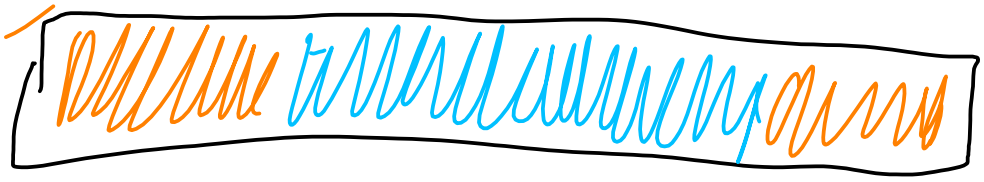
Randomized QS

Worst-case: $O(n^2)$ $O(n^2)$

Expected WC $O(n \log n)$

bad input
normal luck

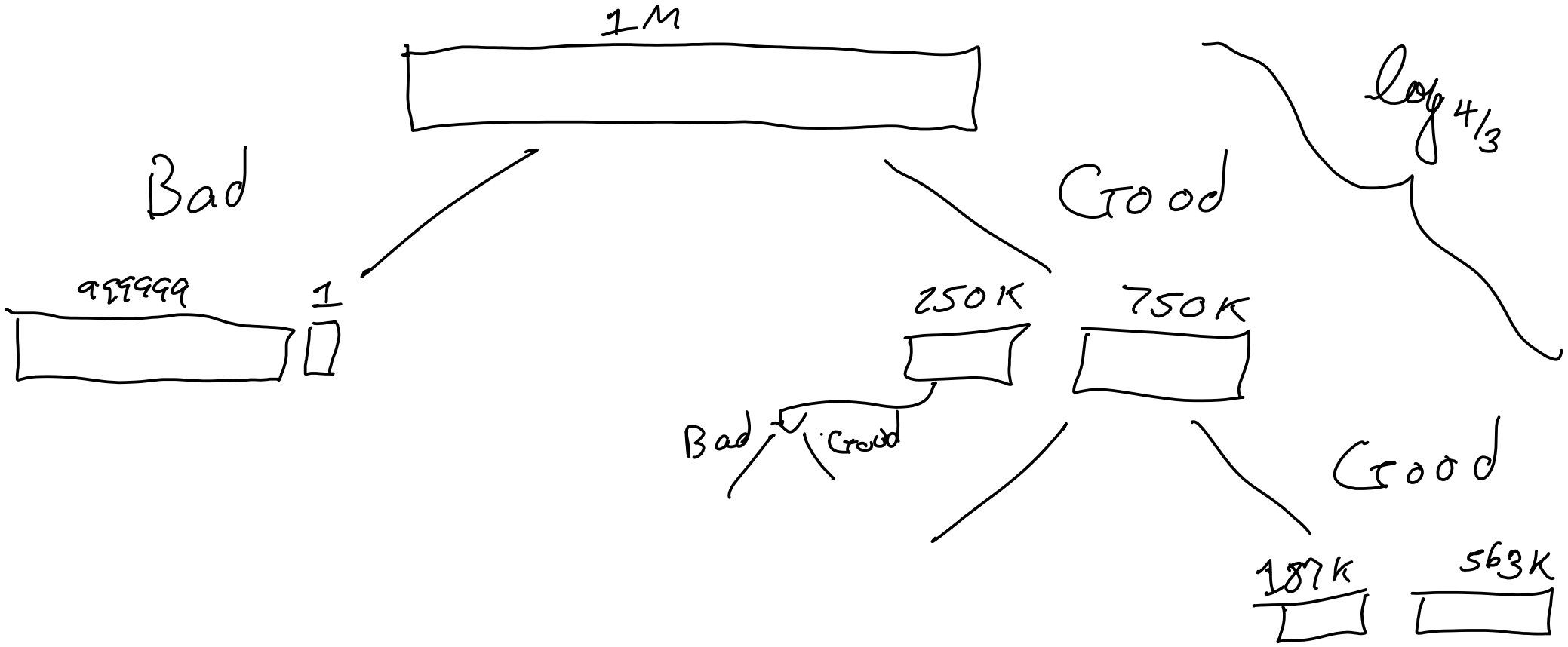
bad input
bad luck

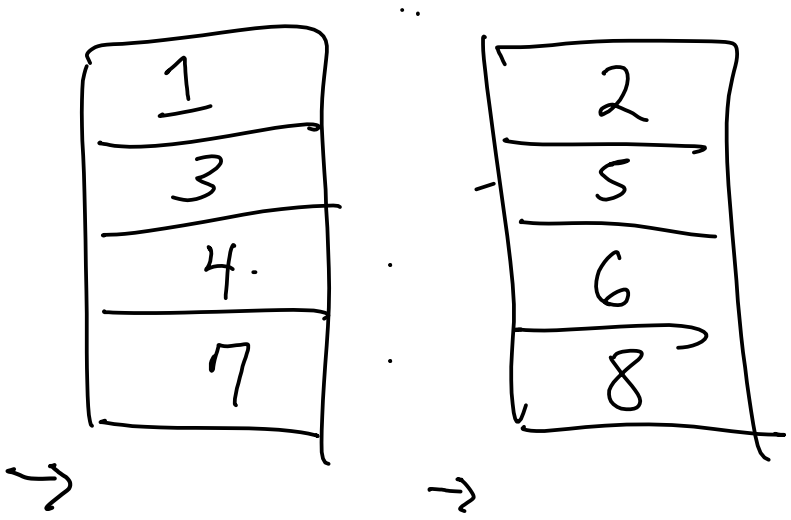


$\log_2 n$

Good Pivot: closer to middle (in values)

Bad Pivot: closer to outside.





1
2
3
4
5
6
7
8

Merge

1. Start a cursor on both piles
2. Move smaller element to a new array & move cursor up
3. Repeat until 1 cursor reaches end
4. Just add everything else from other array

Function MergeSort(A):

If length(A) ≥ 2:

$A_1, A_2 \leftarrow$ split into two halves(A)

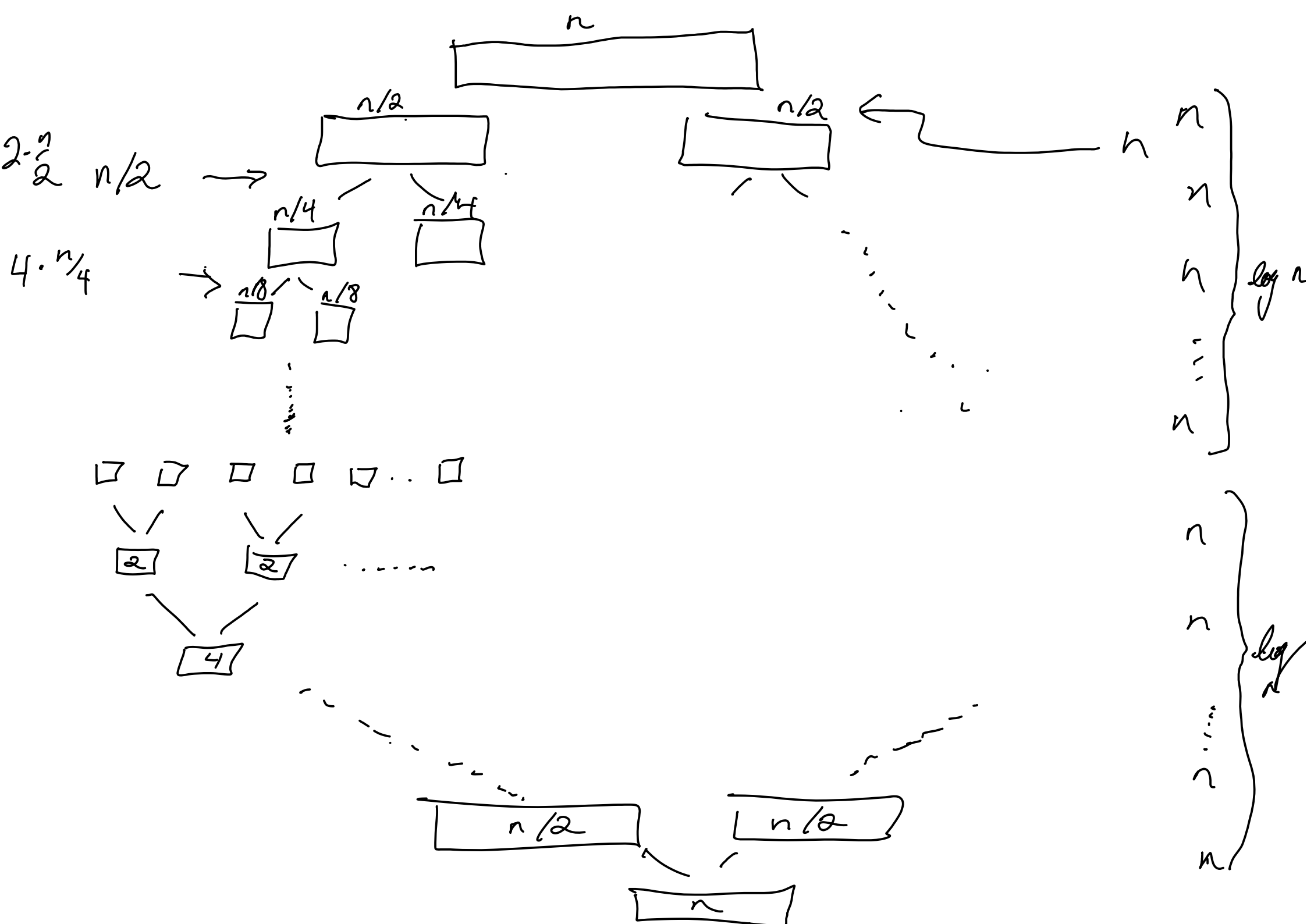
Split

MergeSort(A_1)

MergeSort(A_2)

Merge(A, A_1, A_2)

```
Function Merge(A, A1, A2)
    i ← 0        k ← 0
    j ← 0
    While i < length(A1) and j < length(A2):
        If A1[i] < A2[j]
            A[k] ← A1[i]
            i ← i + 1
            k ← k + 1
        Else
            A[k] ← A2[j]
            j ← j + 1
            k ← k + 1
        End If
    End While
    While i < length(A1)
        A[k] ← A1[i]
        i ← i + 1
        k ← k + 1
    End While
    While j < length(A2)
        A[k] ← A2[j]
        j ← j + 1
        k ← k + 1
    EndWhile
```


n
n/2
n/2
2 · n/2 n/2
n/4
n/4
4 · n/4
n/8
n/8
2
2
4
n/2
n/2
n
n
n
n
n
n
log n
n
n
n
n
log n