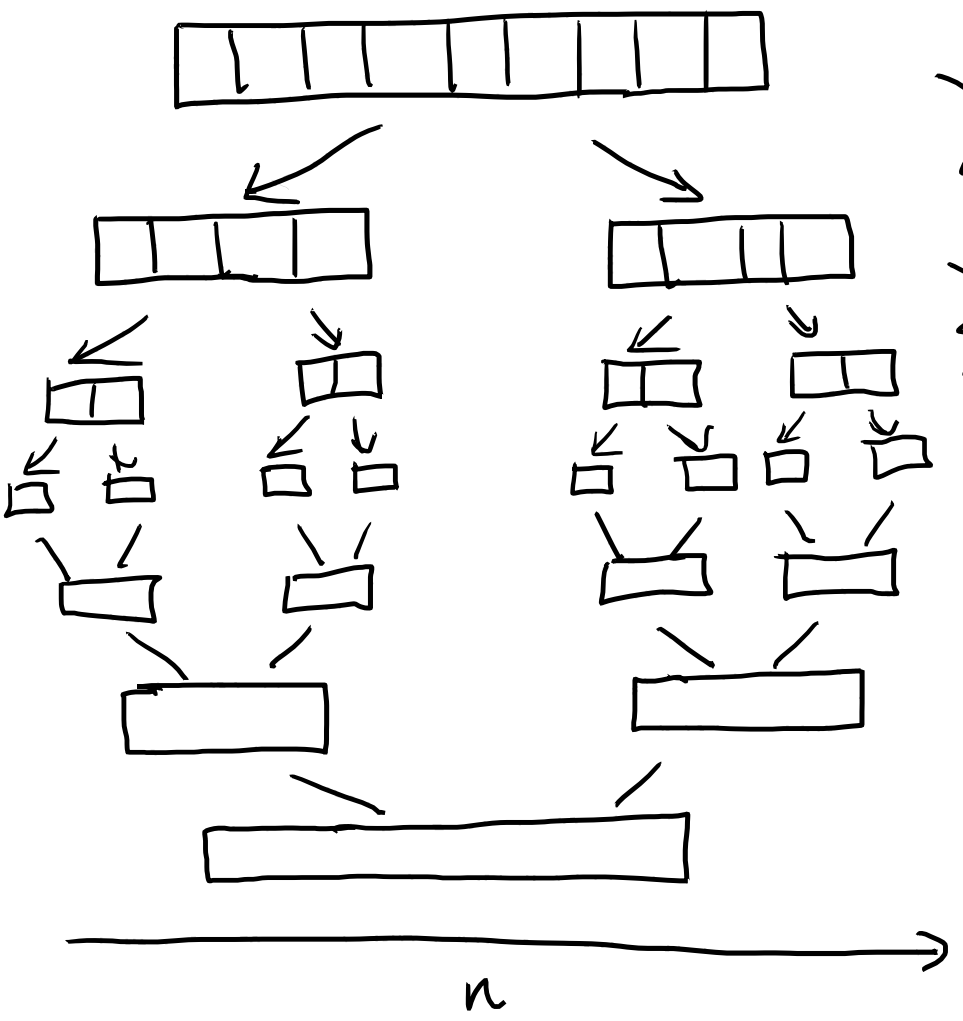
DIVIDE + CONQUER

Requires allocating array space for merging



$2^k = n$

$k = \log_2(n)$

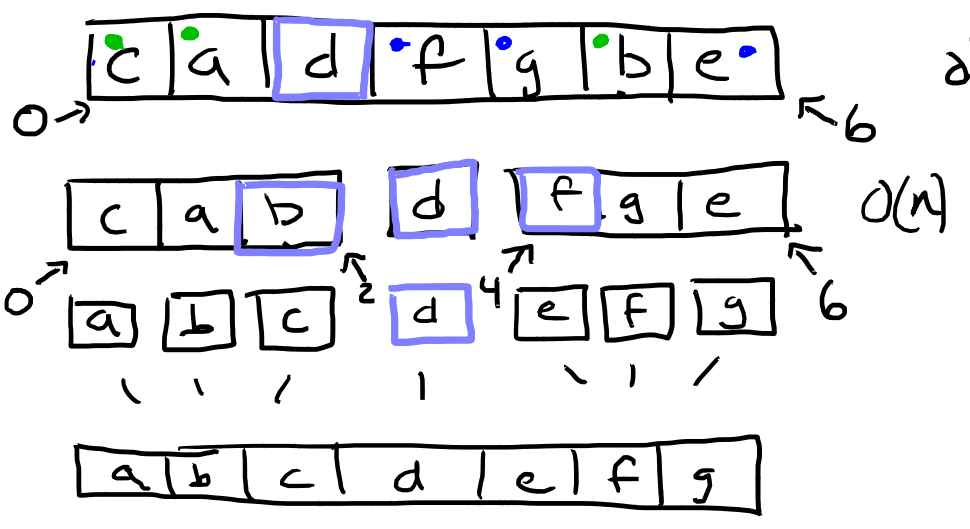
simple dividing

clever combining

merge

$O(n \lg(n))$

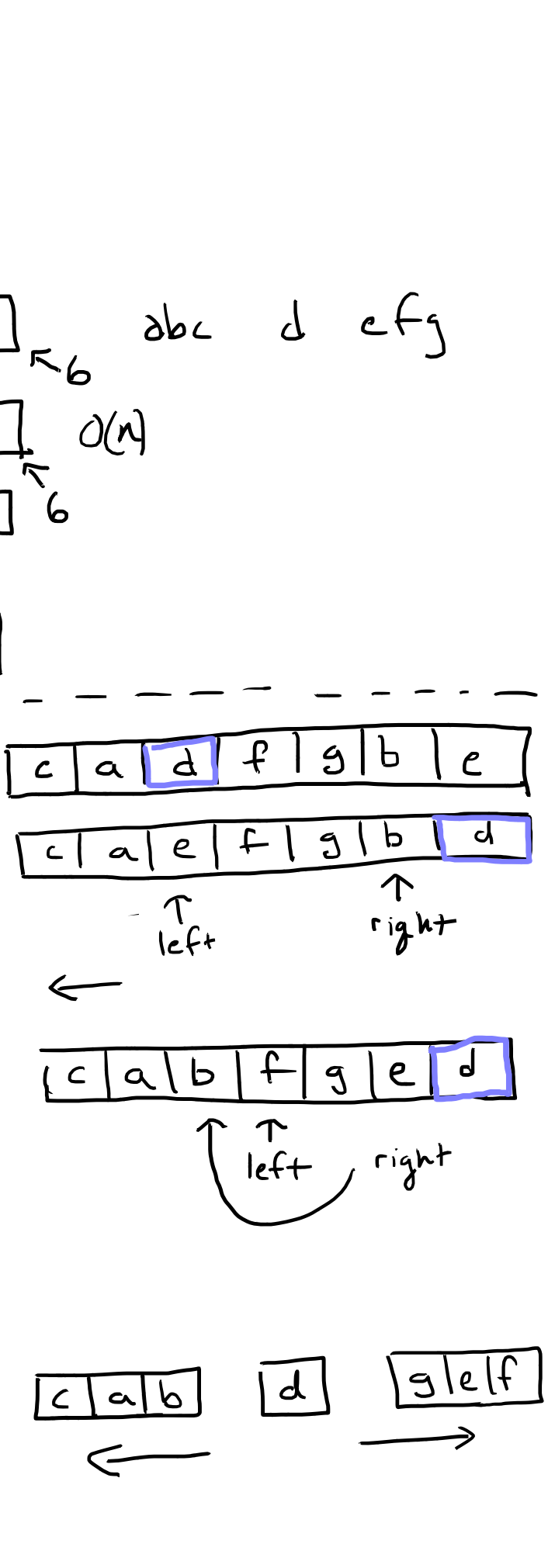
1. Pick "middle"
2. Partitioning elements around "middle"



abc d efg

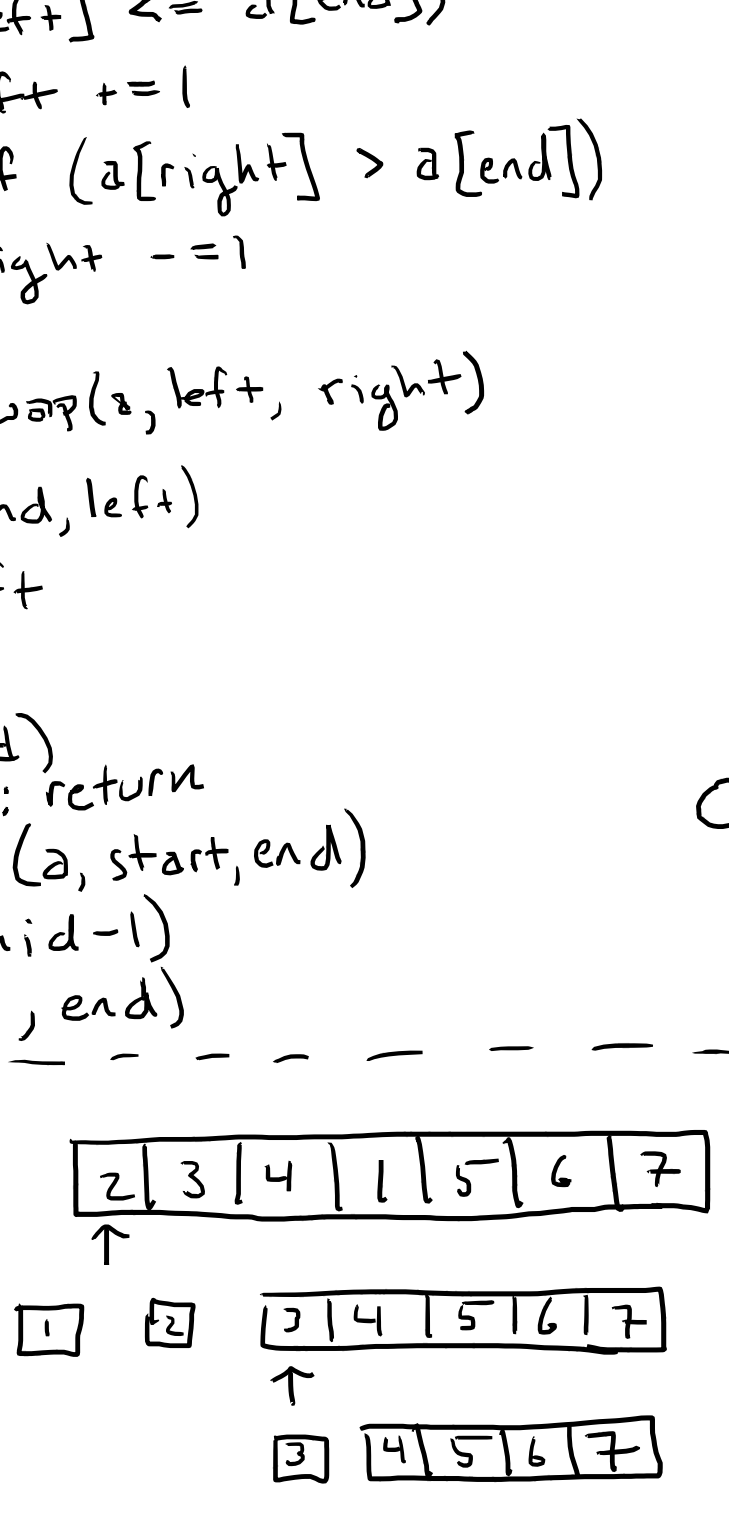
O(n)

```
partition(a, start, end)
    mid = chooseMiddle(a, start, end)
    swap(a, mid, end)
    left = start   right = end - 1
    while(left <= right)
        if (a[left] <= a[end])
            left += 1
        else if (a[right] > a[end])
            right -= 1
        else
            swap(a, left, right)
    swap(a, end, left)
    return left
```



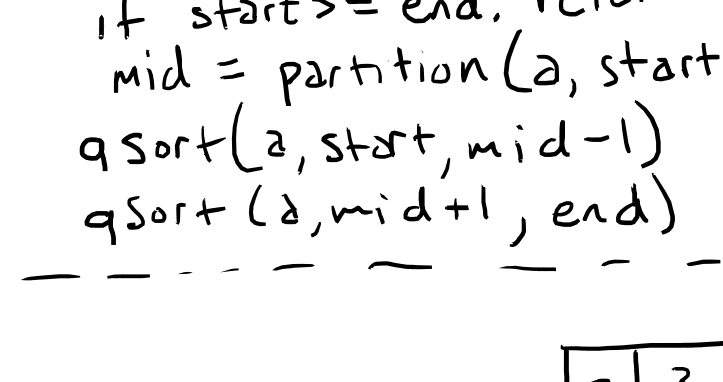

```
qsort (a, start, end)
    if start >= end: return
    mid = partition(a, start, end)
    qsort(a, start, mid-1)
    qsort(a, mid+1, end)
```

QuickSort



chooseMiddle
- O(n)
- median

$O(n^2)$



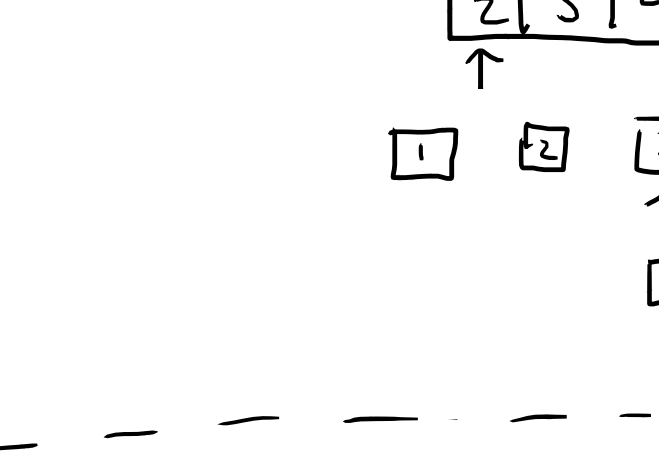
$(3/4)^k n = 1$

$n = (4/3)^k$

$\log_{(4/3)} n = k$

$\frac{\log_2(n)}{\log_2(4/3)}$

$O(n \log_2(n))$ $c \log_2(n)$



$(95/100)^k n = 1$

$\frac{\log_2(n)}{\log_2(100/95)}$ $c \log_2(n)$

99.99% likelihood, $4n \lg(n)$