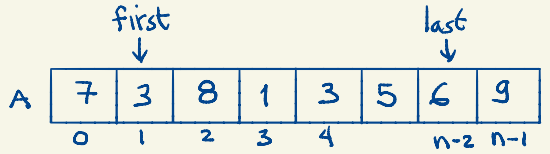
```
Algorithm prefixSums(A,n)
In: Array A storing n integers
Out: ??
if n = 0 then return A[0]                    } C1
else {                                       } C2
    A[n] ← A[n] + prefixSums(A,n-1)
    return A[n]
}
```

Let $f(n)$ = # operations performed by the algorithm when input has size $n$

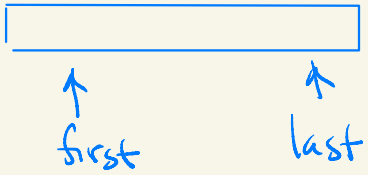
$$f(0) = c_1$$

$$f(n) = c_2 + f(n-1), \quad n > 0$$




Algorithm rev (A, first, last)
In: Array A integers first, last denoting
indices of first and last values in A.
Out: ??

$c_1$ { if first ≥ last then return
else {
$c_2$ tmp ← A[first]
A[first] ← A[last]
A[last] ← tmp
rev (A, first+1, last-1)
}




Let f(n) = # operations performed by the algorithm when input has size n

$f(0) = c_1$

$f(1) = c_1$

$f(n) = c_2 + f(n-2) \quad n > 1$

```
Algorithm test (A,n)
In: Array A of size n ≥ 1
Out: ??
if n = 1 then A[n] ← 0                                   } C1
else {                                                   } C2
    min ← 0
    for i ← 0 to n-1 do {                                } C3 + f(n/2)
        if A[i] < A[min] then min ← i                    } C3
        A[0] ← A[min]
        test(A, n/2)                                     } f(n/2)
    }
}
```

Let $f(n)$ = # operations performed by the algorithm when input has size $n$

$$f(1) = C_1$$

$$f(n) = C_2 + n\left(C_3 + f\left(\frac{n}{2}\right)\right) \quad n \geq 1$$