Algorithms

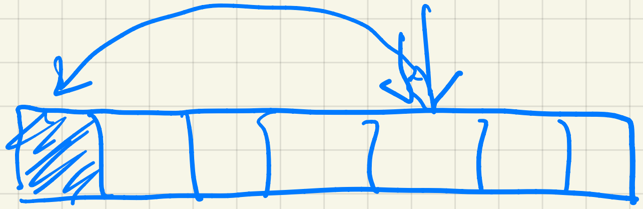
Book : Leiserson MIT

CLRS

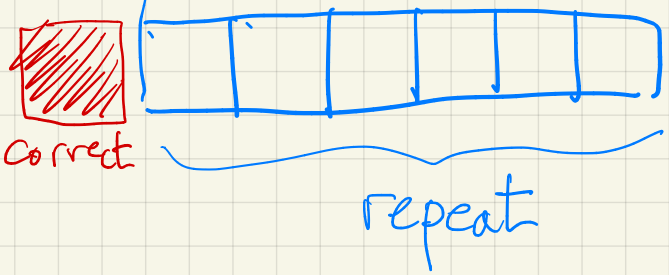
Chapter 2. (sorting)

Sorting Algorithm:

- Bubble Sort
- Selection Sort

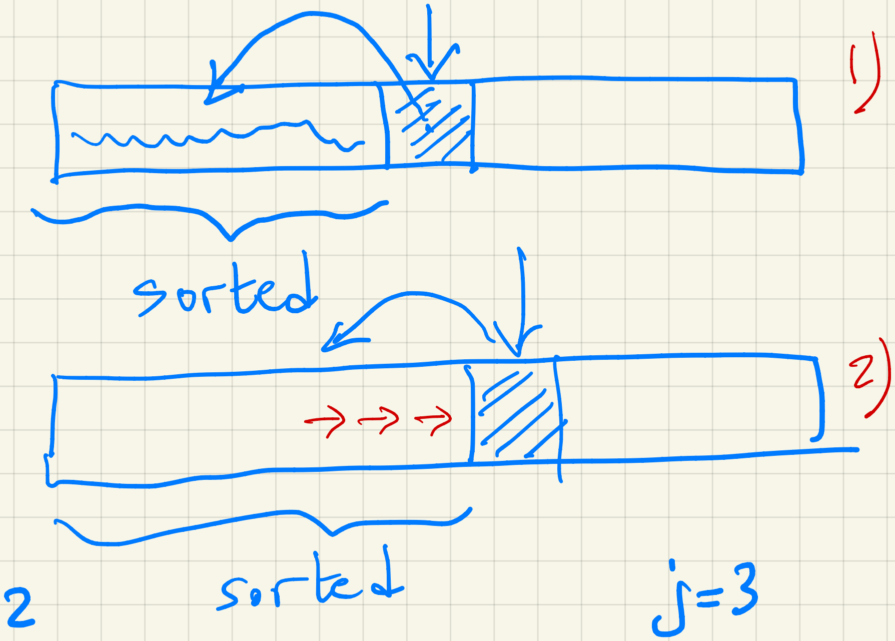


- Find smallest (scan entire array)
- switch with first



Amount of work: not sensitive to original order

Insertion Sort

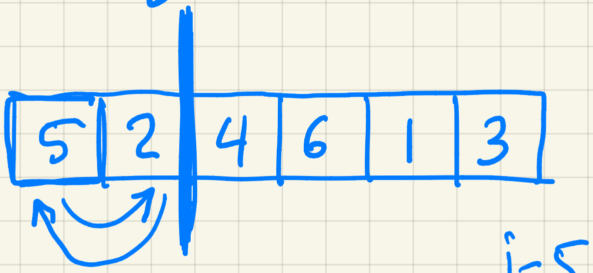


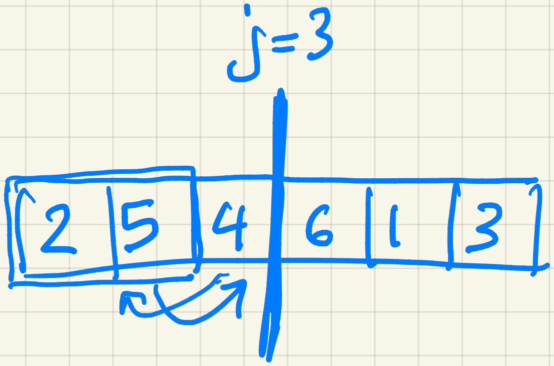
1) Assume A[1...j-1] sorted

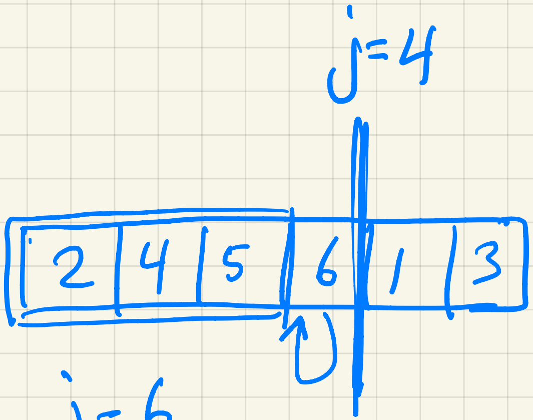
2) Insert A[j] in the correct position

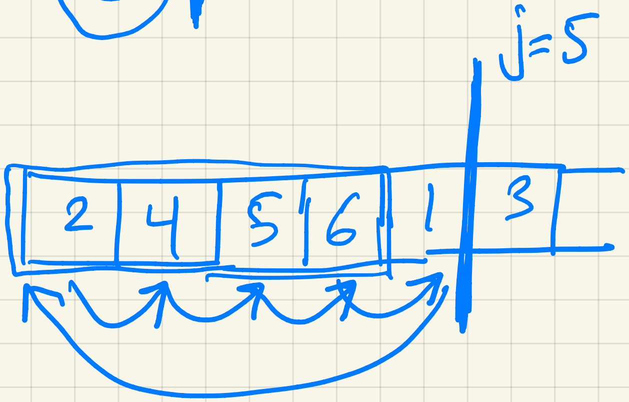
Example:

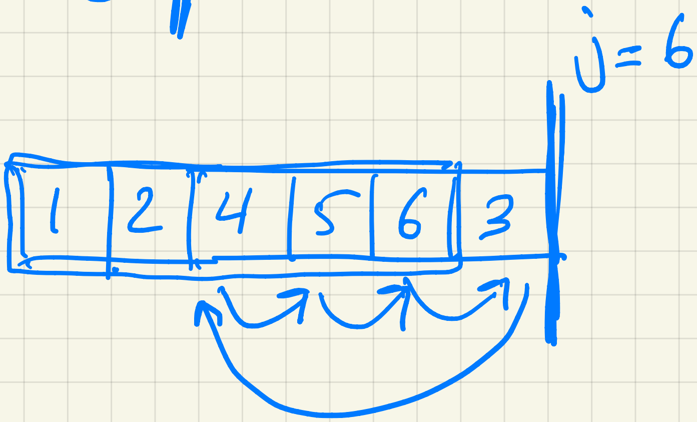
j = 2







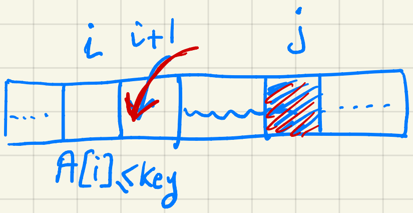




```
                                                              Costs     # times

Insetion Sort (A)
-> for j <- 2 to n                                               1        n
     do key <- A[j]                                              1        n - 1
        |> Insert A[j] into the sorted sequence A[1...j-1]
        i <- j-1                                                 1        n - 1
shifting  { while i > 0 and A[i] > key                           1        $\sum_{j=2}^{n} tj$
process   {      do A[i+1] <- A[i]                               1        $\sum_{j=2}^{n} (tj - 1)$
          {         i <- i-1                                     1        $\sum_{j=2}^{n} (tj - 1)$
insertion     A[i+1] <- key                                      1        n - 1
```

After while loop



loop invariant: Sub array A[1...j-1] at the start of each iteration consists of all the elements originally in A[1...j-1] in sorted order

Initialization: $A[1]$ consists of all elements that are originally in $A[1]$ and sorted.

Maintenance: ✓ think about it

Termination: $j = n+1$

$A[1 \ldots j-1] = A[1 \ldots n]$ consists of all elements that were originally in $A[1 \ldots n]$ in sorted order.

Total time of insertion sort.

$$n + (n-1) + \underbrace{(n-1) + (n-1)} + \sum_{j=2}^{n} t_j + 2\sum_{j=2}^{n} (t_j - 1)$$

$$2\sum_{j=2}^{n}(-1) \qquad -2(n-1)$$

$$n + n - 1 + 3\sum_{j=2}^{n} t_j$$

$$\boxed{2n - 1 + 3\sum_{j=2}^{n} t_j}$$