1 Insertion Sort (CLRS §2.1, 2.2)

1.1 Intro and Examples


[1]

Example

8	2	4	9	3	6
2	8	4	9	3	6
2	4	8	9	3	6
2	4	8	9	3	6
2	3	4	8	9	6
2	3	4	6	8	9

[1] Try it! https://www.lemmaplay.com/sort/insertion-sort.html

1.2 Pseudocode (& General Pseudocode Conventions and Notation)

```
for j =
    key =
    i =
    while
    A[i+1] =
```

1.3 Proof of Correctness

1.3.1 Aside: Proof by Induction and Loop Invariants

- Proof by induction: We want to prove some statement $P(n)$ for integers $n \geq 0$.
  1. Base case:
  2. Inductive step:
  3. Conclude $P(n)$ holds for all integers $n \geq 0$.
- Loop invariant: A property that holds throughout the execution of the algorithm
  1. Initialization:
  2. Maintenance:
  3. Termination: When the loop terminates, invariant gives useful property that helps show that the algorithm is correct.

Our loop invariant:

1.3.2 Proof of correctness of insertion sort

- Initialization:

- Maintenance:

- Termination:

1.4 Running time

Analyzing running time of an algorithm - each basic operation takes constant time: e.g. addition, assigning a variable, checking next number in array, etc.

What is t_j?

- If key $> A[j-1]$, then
- If $A[j] < A[1]$, then

Overall running time:

- In the worst case,
- In the best case,
- What about average case?

Final note:

2 Order of Growth and Asymptotic Behavior (CLRS §3.1)

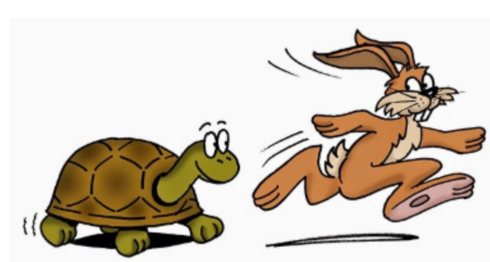
Goal: Establish notation that enables us to compare relative performance of different algorithms.

Definition For a function $g : \mathbb{N} \to \mathbb{R}^+$,

- $\Theta(g(n)) =$

- $\mathcal{O}(g(n)) =$

- $\Omega(g(n)) =$



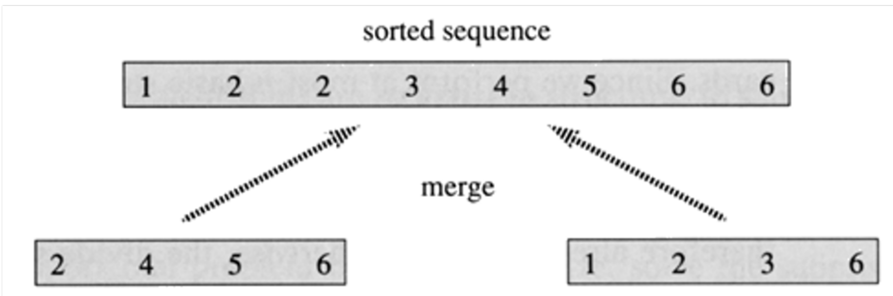
Example

- polynomials
- $n \log n$
- $2^n \log n$
- 3^n
- $3^n + 2^n$

Claim. $f \in \mathcal{O}(g(n))$ if and only if $g \in \Omega(f(n))$
Proof.

3 Divide and Conquer

3.1 Mergesort (CLRS §2.3)



3.1.1 The Merge Subroutine

```
i = 1
j = 1
for k = 1 to
    if
        C[k] =
        i =
    else
        C[k] =
        j =
```

3.1.2 Proof of Correctness of Merge

Loop invariant:

- Initialization:

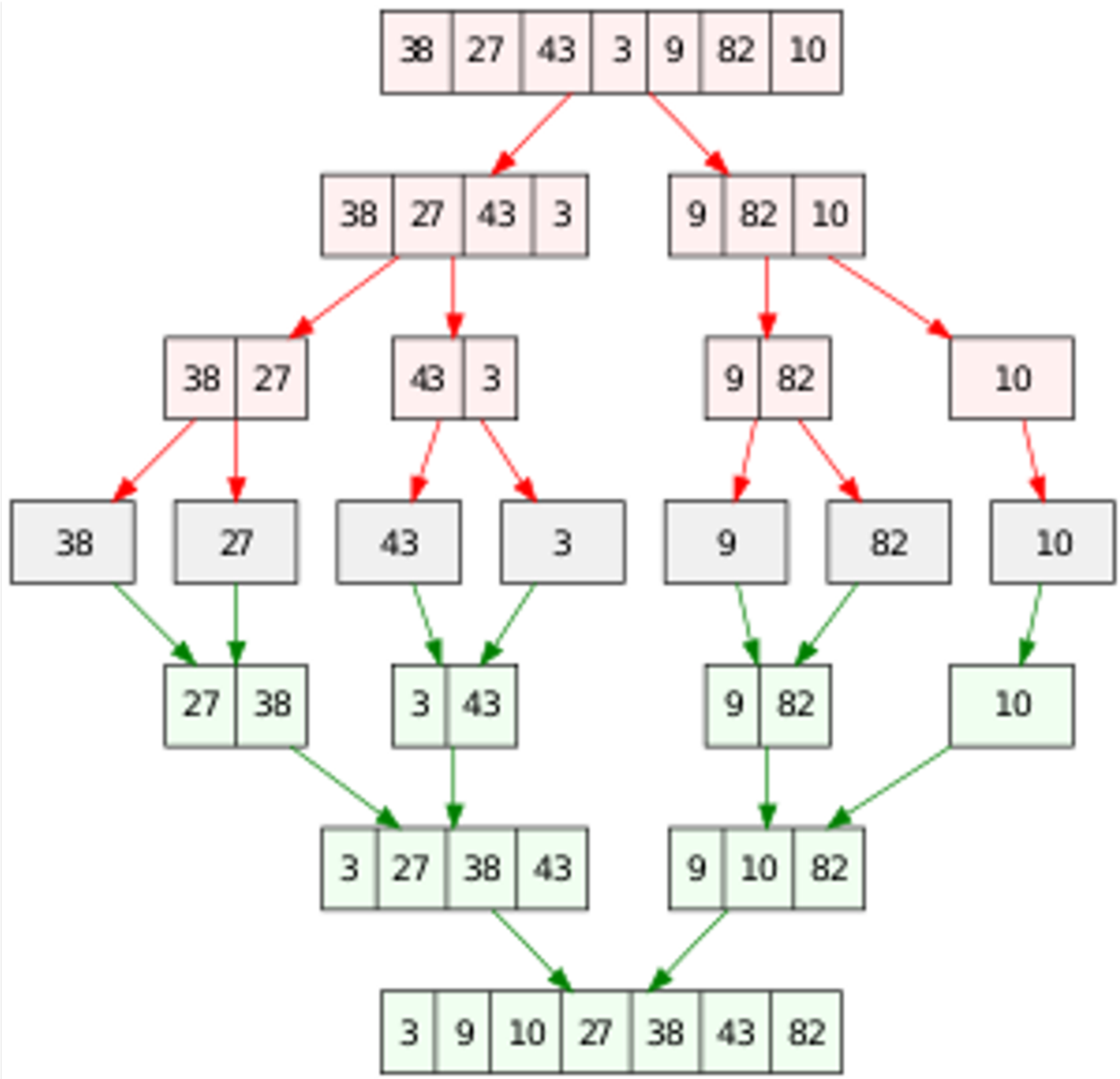
- Maintenance:

- Termination:

3.1.3 Running Time of Merge

3.1.4 Back to Mergesort: Correctness, Running Time, Recursion Tree

Example


38 27 43 3 9 82 10
38 27 43 3
9 82 10
38 27
43 3
9 82
10
38
27
43
3
9
82
10
27 38
3 43
9 82
10
3 27 38 43
9 10 82
3 9 10 27 38 43 82

3.2 Intro to Solving Recurrences (CLRS §4.3, 4.4)

3.3 Quicksort (CLRS §7.1, 7.2)

Idea:

Example

```
k=PARTITION
QUICKSORT
QUICKSORT
```

3.3.1 Partition

```
pivot =
i =
for j = 1 to n-1
    if A[j]

        i =

RETURN i
```

3.3.2 Correctness of Partition (and Quicksort)

Loop Invariant:

3.3.3 Running Time of Partition and Quicksort