Introduction to Algorithms

Date: 3/6 (Thursday)

Instructor: 유준수

Assignment

- Read 1.1, 1.2, 2.1
- Problems:
  - 1.1절 – 1, 2, 4
  - 1.2절 – 2, 3
  - 2.1절 – 1,2,3,4

Chapter 1. The Role of Algorithms in Computing

- 1.1 Algorithms
- 1.2 Algorithms as a technology

What is Algorithm?

$I \rightarrow \boxed{f} \rightarrow O$

precise descriptin

pseudocode

Example of Algorithm: Sorting

$$\langle a_1, a_2, \cdots, a_n \rangle$$

$$\Big\downarrow A$$

$$\langle a_1', \cdots, a_n' \rangle \quad \text{s.t.}$$

$$a_1' \le a_2' \le \cdots \le a_n'$$

Toy Example of Sorting

$$\langle 31, 71, 51, 41, 81 \rangle$$

$$\downarrow \mathcal{A}$$

$$\langle 31, 41, 51, 71, 81 \rangle$$

Toy Example of Sorting

Correctness

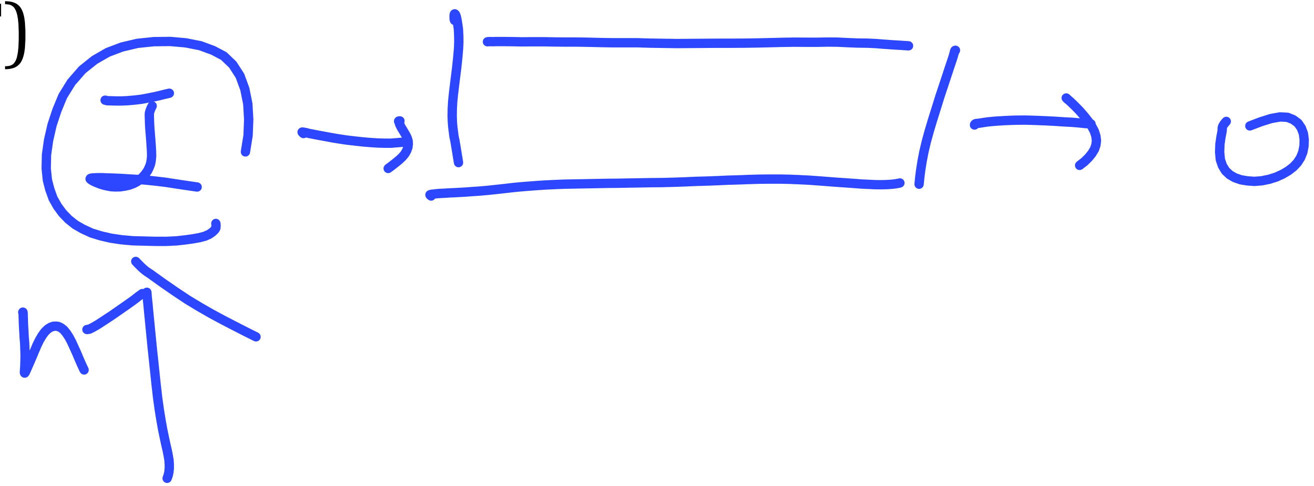
1. Halt
2. Correct Output

$$2 \to \boxed{+3} \to \cancel{4}$$

Many area of application:

$a^b \bmod p$

- Information Retrieval from Large DB on Internet
- Public Key & Digital Signature (RSA)
- Shortest Path Problem (Graph)
- Fast Fourier Transform (FFT)
- Machine Learning
- Deep Learning



Efficiency

- Insertion Sort: $c_1 n^2$
- Merge Sort: $c_2 n \log n$

n

c_1, c_2

$n^2 = n \times n$

$n \lg = n \times \log n$

Efficiency Example

$c_1 n^2$

$c_2 n \log n$

- Data size: $n = 10^7$

Name	Computer	Algorithm
Bob	A - 10^{10} $\left(\frac{instruction}{s}\right)$	Insertion Sort ($2n^2$)
Alice	B - 10^7 $\left(\frac{instruction}{s}\right)$	Merge Sort ($50\ nlogn$)

Bob $\frac{2 \cdot (10^7)^2}{10^{10}} = 2 \times 10^4$ s

Alice $\frac{50 \times \cancel{10^7} \times \log 10^7}{\cancel{10^7}}$ $= 50 \times 7 \times \log 10$ $\doteq 10^3$ s

3.xx

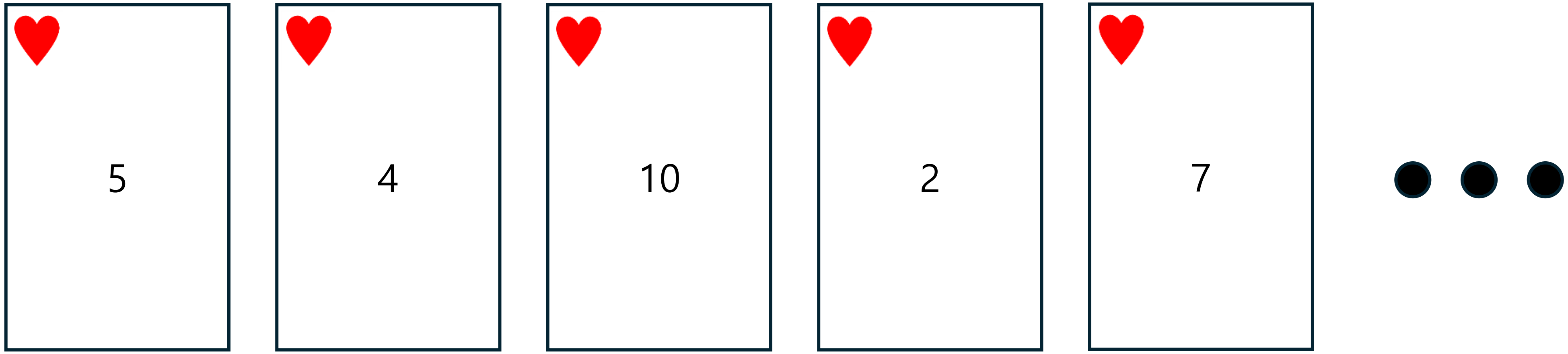
$\parallel$

$2 \times 10 = 20 \times$

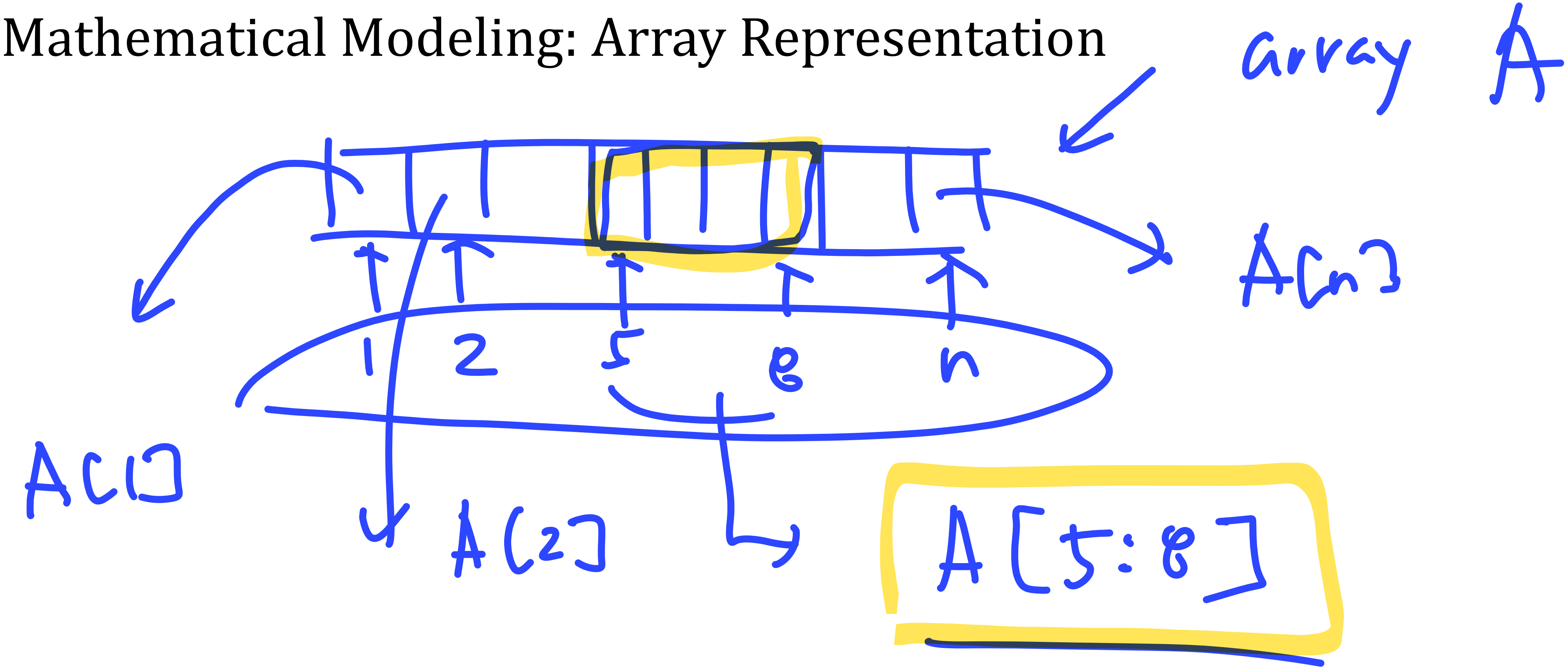
Chapter 2. The Role of Algorithms in Computing

- 2.1 Insertion sort
- 2.2 Analyzing algorithms
- 2.3 Designing algorithms

Card Example in Sorting

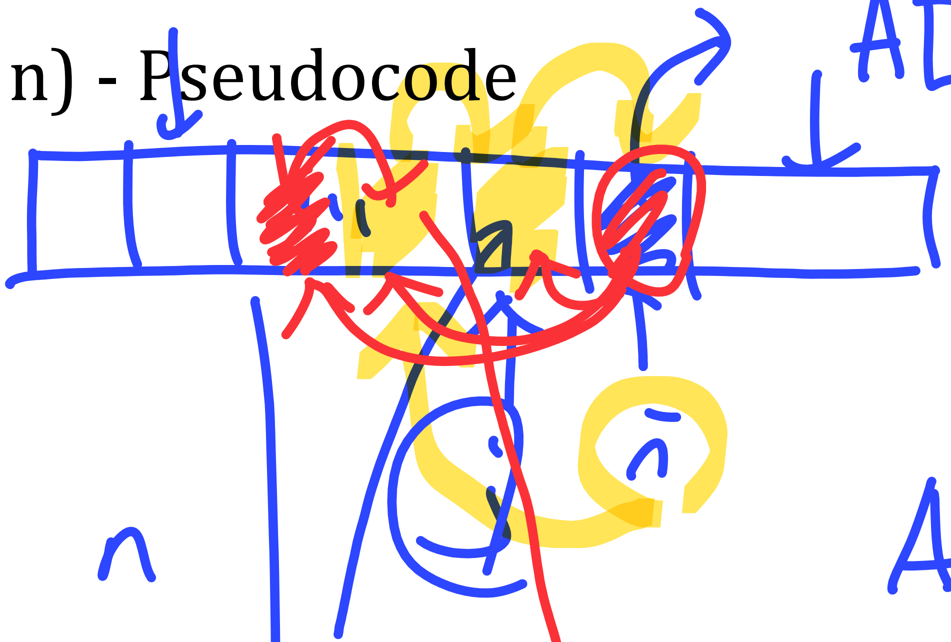


Mathematical Modeling: Array Representation



Insertion_Sort(A, n) - Pseudocode

$A[i] = key$



for $i=2$ to n

$key = A[i]$

$j = i - 1$

while $A[j] > key$, $j > 0$

$A[j+1] = A[j]$

$j = j - 1$

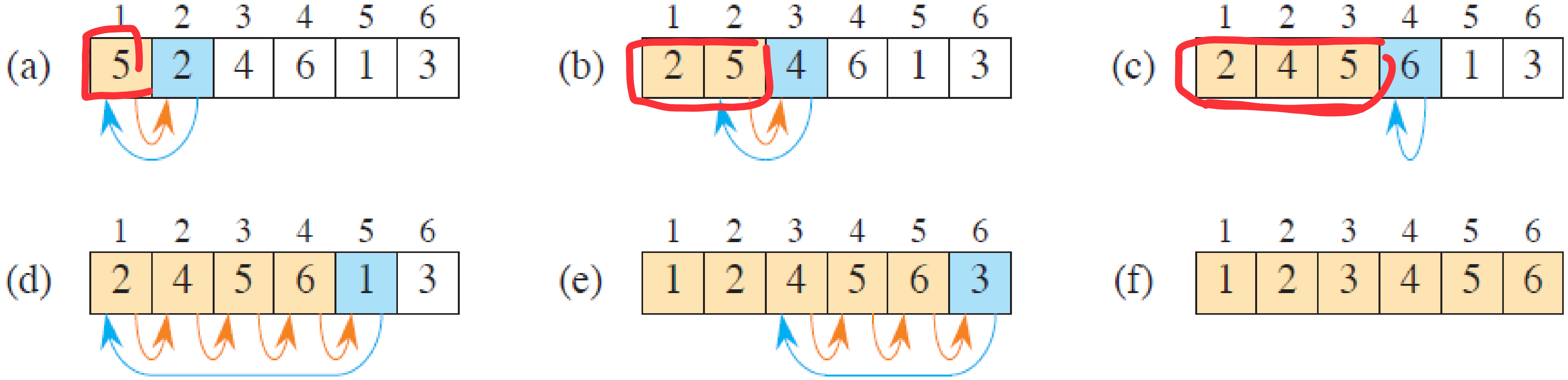
$A[j] > key$

$A[j+1] = A[j]$

$j = j - 1$

$A[j+1] = key$

Insertion_Sort(A, n) – Graphical Representation

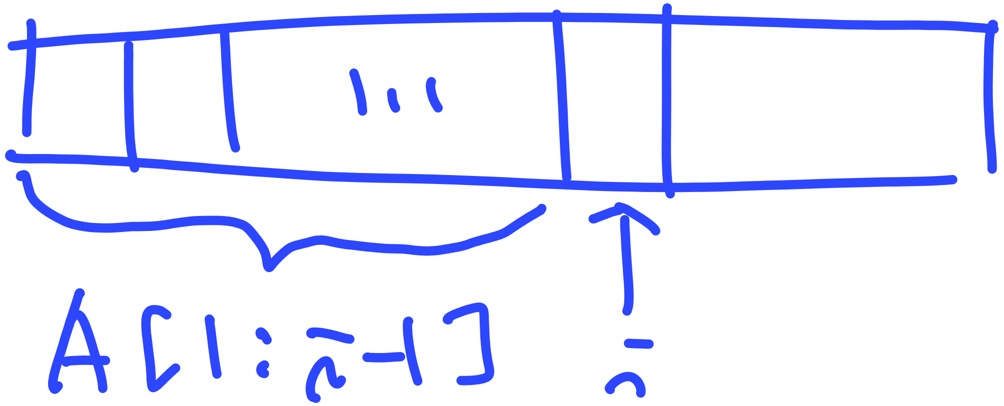


Algorithm Correctness

- Loop Invariant: A loop invariant is a property of a program loop that is true before (and after) each iteration

1. Initialization: It is true prior to the first iteration of the loop.
2. Maintenance: If it is true before an iteration of the loop, it remains true before the next iteration.
3. Termination: The loop terminates, and when it terminates, the invariant gives us a useful property that helps show that the algorithm is correct.

Insertion_Sort(A, n) - Correctness



Loop Inv

(A[1:i-1] is sorted)

Init   A[1]   sorted

Main   A[1:i-1] is sorted

A[i] will be positioned at the right place ⇒ A[1:i]

Ter A[1:n] is sorted is sorted

Question?