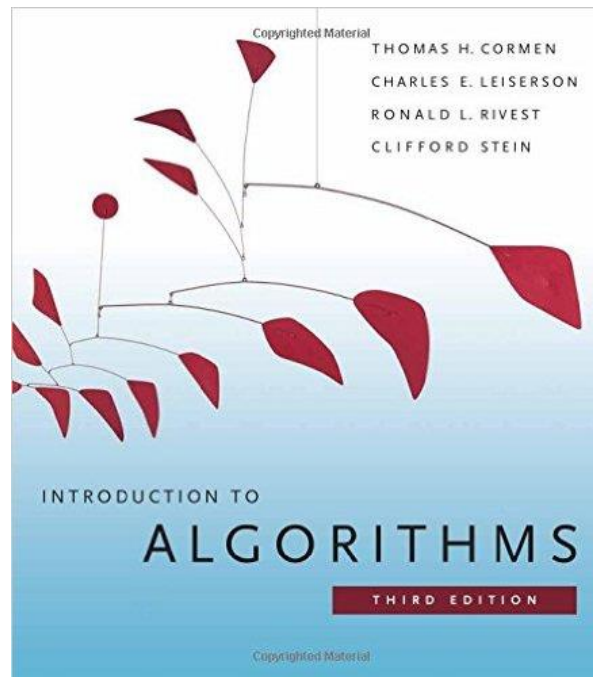


طراحی الگوریتم‌ها

فصل اول

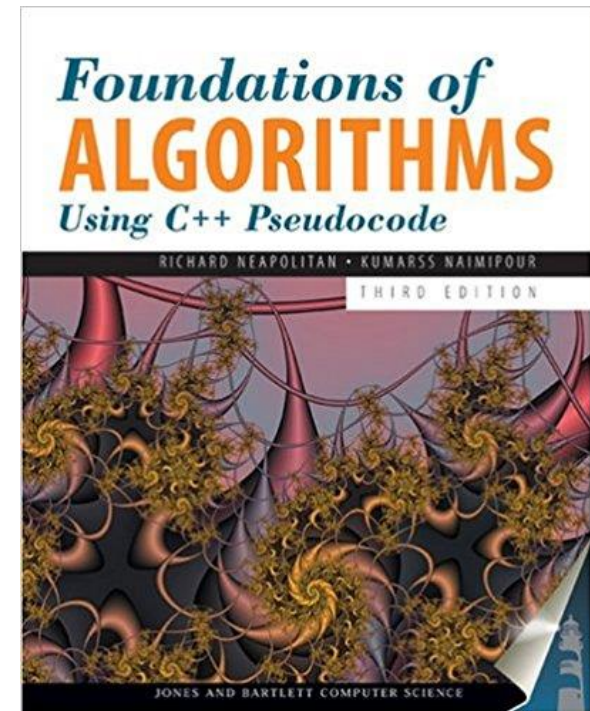
منابع

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, **Introduction to Algorithms** (3rd ed.). MIT Press, 2009.



منابع (ادامه)

- Richard Neapolitan, Kumarss Naimipour. **Foundations of Algorithms Using C++ Pseudocode** (3rd ed.). MIT Press, 2008.





ارزیابی

- تمرین: ۱۵ نمره
- میان ترم: ۲۵ نمره
- پروژه: ۲۵ + نمره
- پایان ترم: ۴۰ نمره
- مجموع: ۱۰۵

توجه ...

- تمرینات و پروژه باید تا **موعد مشخص** شده تحویل داده شوند.
- تمرینات و اطلاعیه‌ها در گروه تلگرام آپلود خواهد شد. (Telegram Channel: @Alg_UMA971)
- عنوان ایمیل فرستاده شده به فرمت زیر باشد:
- Subject: AlgUMA971_Topic & #_Student Name
- Example: AlgUMA971_HW2_AliHoseini
AlgUMA971_ProjectV3_HasaanAbam
AlgUMA971_Question_RezaAmiri
- ایمیل جهت ارتباط: S.Najjar.G@gmail.com

فهرست مطالب

- ❖ مقدمه‌ای بر الگوریتم‌ها و مفاهیم پایه
- ❖ معرفی پیچیدگی زمانی و حافظه‌ای و روشهای تحلیل مسائل
- ❖ روش‌های حل مسئله
- ❖ تقسیم و غلبه (حل)
- ❖ برنامه نویسی پویا
- ❖ روش حریصانه
- ❖ راهبرد عقبگرد
- ❖ شاخه و حد
- ❖ آشنایی نظریه NP و مسائل مربوط به آن

مقدمه‌ای بر الگوریتم‌ها و مفاهیم پایه

Algorithm

- **Definition**

An *algorithm* is a finite set of instructions that accomplishes a particular task.

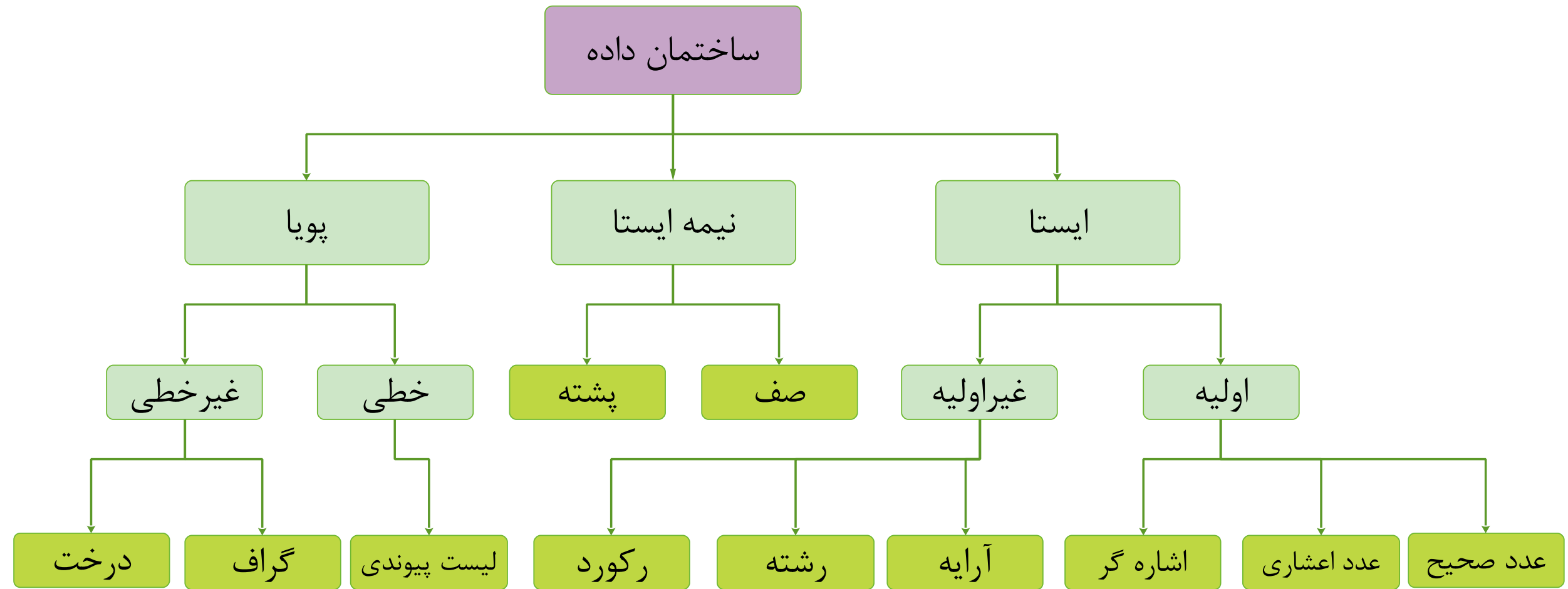
- **Criteria**

- Input
- Output
- Definiteness: clear and unambiguous
- Finiteness: terminate after a finite number of steps
- Effectiveness: instruction is basic enough to be carried out

Data Structure

- A *data structure* is a way to store and organize data in order to facilitate access and modification
- No single data structure work well for all purpose.
- It is important to know the strengths and limitations of several of them

Data Structure



Algorithms: Measurements

- Criteria
 - Is it correct?
 - Is it readable?
 - ...
- Performance Analysis (machine independent)
 - space complexity: storage requirement
 - time complexity: computing time
- Performance Measurement (machine dependent)

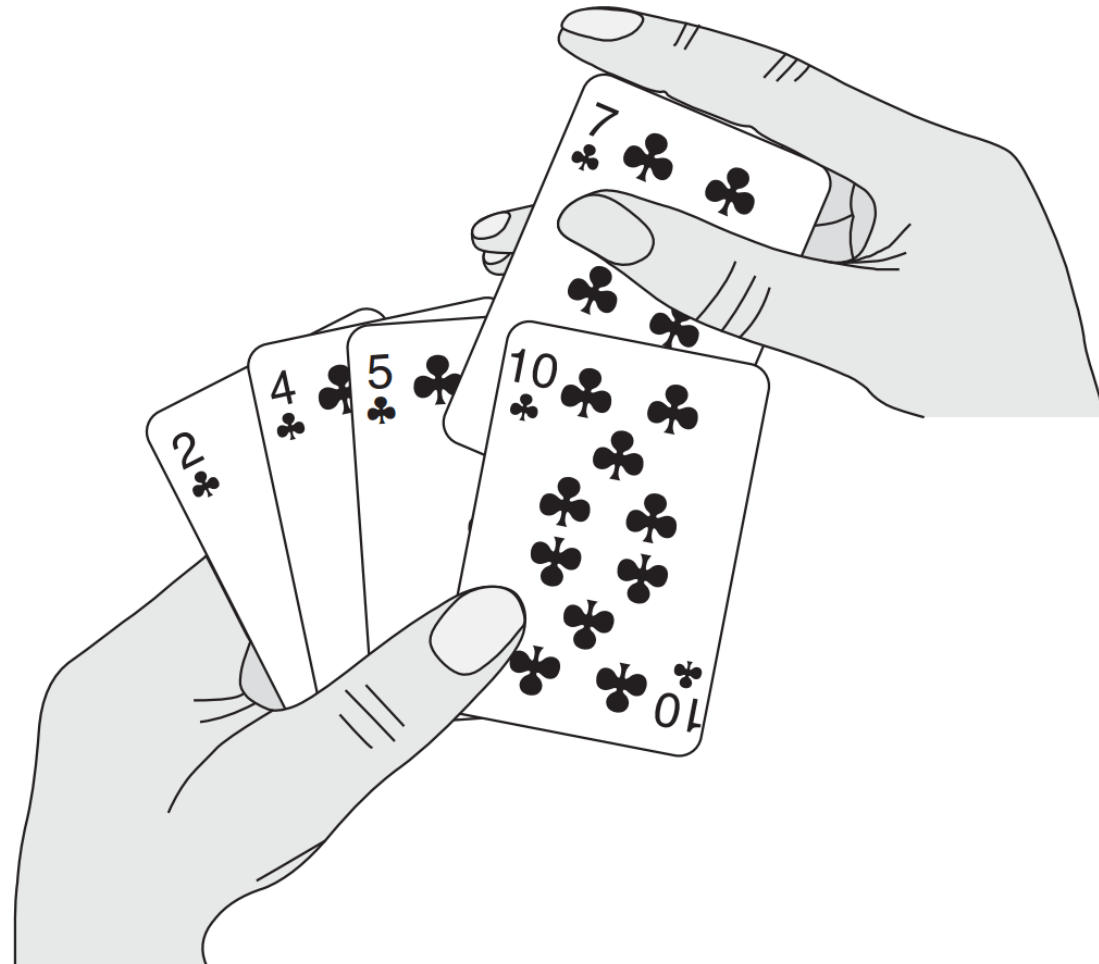
The goals of this course:

- How to **devise** algorithms?
- How to **express** algorithms?
- How to **validate** algorithms?
- How to **analyze** algorithms?
- How to **test** algorithms?

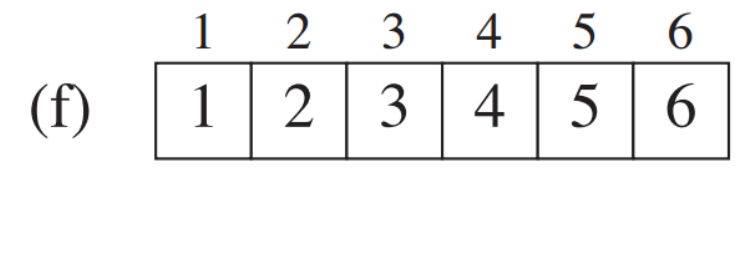
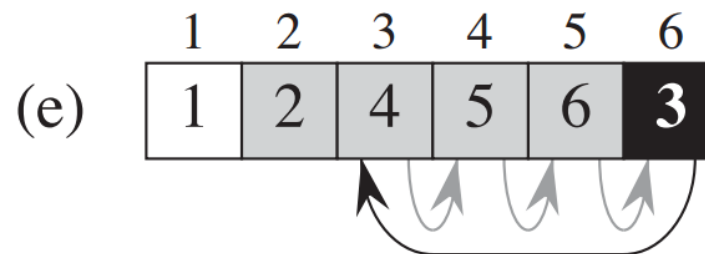
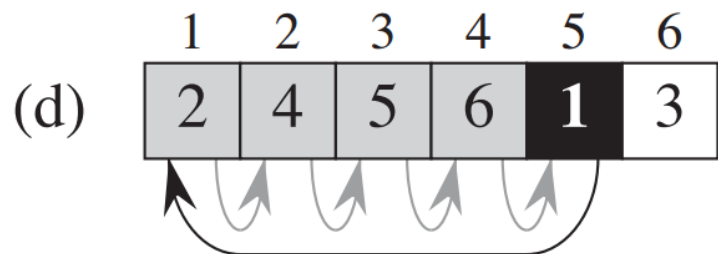
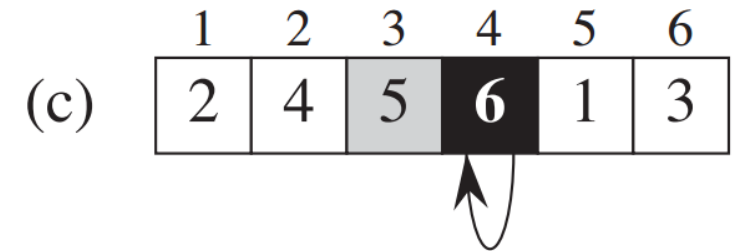
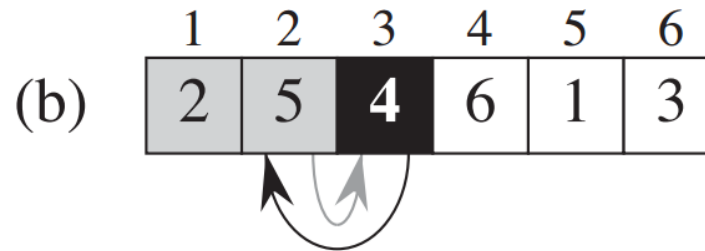
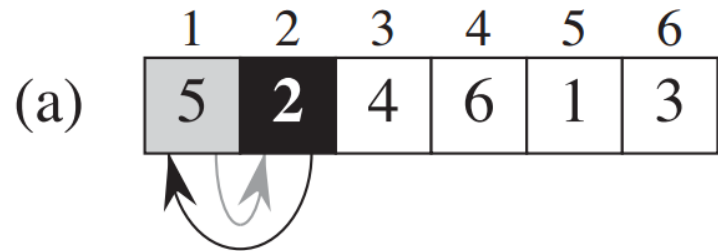
The following techniques will be discussed in this course:

- Divide and Conquer
- Dynamic Programming
- Greedy Algorithms
- Backtracking Algorithms
- Branch and Bound Algorithms

Insertion Sort: How to devise?



Insertion Sort: How to devise? (Cont.)



Insertion Sort: How to express?

INSERTION-SORT(A)

```
1  for  $j = 2$  to  $A.length$ 
2       $key = A[j]$ 
3      // Insert  $A[j]$  into the sorted sequence  $A[1..j-1]$ .
4       $i = j - 1$ 
5      while  $i > 0$  and  $A[i] > key$ 
6           $A[i + 1] = A[i]$ 
7           $i = i - 1$ 
8       $A[i + 1] = key$ 
```


Insertion Sort: How to validate?

- After the termination of InsertionSort algorithm, the input array A is sorted.
- Lemma: At the start of each iteration of the for loop of lines 1-8, the subarray $A[1:j - 1]$ consists of the elements originally in $A[1::j - 1]$ but in sorted order.

Insertion Sort: How to validate? (Cont.)

Proof based on induction (Loop Invariant, in this case).

- **Initialization:** $j = 2 \Rightarrow A[1::1] = A[1]$, which is sorted.
- **Maintenance:** $A[j]$ is inserted in the correct position, so $A[1::j]$ is sorted.
- **Termination:** This happens when $j = n + 1$. So $A[1::j - 1] = A[1::n]$ is an ordered array

Insertion Sort: How to analyze?

Computing the amount of resources (Time, Space, etc.) needed by the algorithm.

INSERTION-SORT(<i>A</i>)	<i>cost</i>	<i>times</i>
1 for <i>j</i> \leftarrow 2 to <i>length</i> [<i>A</i>]	c_1	n
2 do <i>key</i> \leftarrow <i>A</i> [<i>j</i>]	c_2	$n - 1$
3 ▷ Insert <i>A</i> [<i>j</i>] into the sorted sequence <i>A</i> [1 .. <i>j</i> - 1].	0	$n - 1$
4 <i>i</i> \leftarrow <i>j</i> - 1	c_4	$n - 1$
5 while <i>i</i> > 0 and <i>A</i> [<i>i</i>] > <i>key</i>	c_5	$\sum_{j=2}^n t_j$
6 do <i>A</i> [<i>i</i> + 1] \leftarrow <i>A</i> [<i>i</i>]	c_6	$\sum_{j=2}^n (t_j - 1)$
7 <i>i</i> \leftarrow <i>i</i> - 1	c_7	$\sum_{j=2}^n (t_j - 1)$
8 <i>A</i> [<i>i</i> + 1] \leftarrow <i>key</i>	c_8	$n - 1$

Insertion Sort: How to analyze? (Cont.)

Computing the amount of resources (Time, Space, etc.) needed by the algorithm.

$$T(n) = c_1 n + (c_2 + c_4 + c_8)(n - 1) + c_5 \sum_{j=2}^n t_j + (c_6 + c_7) \sum_{j=2}^n t_j - 1.$$

Insertion Sort: How to test?

- Just **implement** the pseudocode in any programming language and execute it with different instances of random arrays as input...

Homework ...

Answer to the five mentioned questions for the following problems:

- Bubble Sort
- Sequential Search
- Binary Search