مقدمه و نقش الگوریتم‌ها

- شروع استفاده از Framework ها برای توصیف و تحلیل الگوریتم ها
- بررسی دو الگوریتم برای مسئله sorting
- نحوه توصیف الگوریتم‌ها توسط pseudocode
- استفاده از asymptotic notation برای تحلیل و بیان running time
- آشنایی مختصر با یکی از تکنیک های طراحی الگوریتم در context مربوط به یک الگوریتم

The sorting problem

Input : A sequence of n numbers $(a_1, a_2, \ldots, a_n)$

Output : A permutation $(a'_1, a'_2, \ldots, a'_n)$ of the input sequence such that $a'_1 \leq a'_2 \leq \ldots \leq a'_n$.

- دنباله ها معمولاً در آرایه‌ها قرار دارند و براساس اندیس و مکان قرارگیری sorted هستند.
- معمولاً به اعداد در زبان الگوریتم keys گفته می‌شود.
- به همراه هر key ممکن است یکسری اطلاعات اضافه هم باشد که به آنها satellite data گفته می‌شود.
- برای حل مسئله sorting، چندین و چند راه وجود دارد.
- به هر راه برای حل مسئله یک الگوریتم گفته می‌شود.

- An algorithm :
a well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output

نحوه بیان یک الگوریتم :

- یک الگوریتم باید به گونه ای نوشته شود که بیشترین شفافیت و بیشترین اختصار را داشته باشد.
- زبان انگلیسی بعضی وقتها بهترین راه است.
- برای بیان موارد و دستورات کنترلی ، بهترین راه استفاده از pseudo code است.
- pseudo code شبیه به زبانهای C ، C++ ، پاسکال و java است.
- pseudocode زبان بیان و توصیف الگوریتم برای انسان است .
- pseudo code موارد سطح برای مهندسی نرم افزار مانند data abstraction ، modularity و error handling را اغلب نادیده می گیرد.
- در pseudo code استفاده از جملات کامل انگلیسی رایج است .
- الگوریتم نوشته شده توسط pseudo code برای ماشین مناسب نیست .

Insertion sort:

- یک الگوریتم خوب برای sorting یک مجموعه کم از اعداد است.
- درست به همان روشی که یک دست از پاسورها را مرتب می کنید .
- ابتدا همه کارت ها را به پشت بر روی میز قرار دارند و دست چپ شما خالی است .
- یک کارت از روی میز برمی دارید و آن را در جای مناسب sort شده در دست چپ خود قرار می دهد .
- برای یافتن جای مناسب آن را با همه کارتهای موجود در دست چپ یکی یکی مقایسه می کنید (از راست به چپ)

- در هر زمان، تمام کارتهای موجود در دست چپ شما مرتب شده است.

5 2 4 6 1 3

5 | 2 4 6 1 3

5 2 | 4 6 1 3

2 5 4 | 6 1 3

2 4 5 6 | 1 3

2 4 5 6 1 | 3

1 2 4 5 6 3 |

1 2 3 4 5 6

```
INSERTION-SORT(A)
for j ← 2 to n
    do key ← A[j]
       ▷ Insert A[j] into the sorted sequence A[1 . . j − 1].
       i ← j − 1
       while i > 0 and A[i] > key
           do A[i + 1] ← A[i]
              i ← i − 1
       A[i + 1] ← key
```

```
INSERTION-SORT(A)                                          cost   times
for j ← 2 to n
    do key ← A[j]
       ▷ Insert A[j] into the sorted sequence A[1 . . j − 1].
       i ← j − 1
       while i > 0 and A[i] > key
            do A[i + 1] ← A[i]
               i ← i − 1
       A[i + 1] ← key
```

```
INSERTION-SORT(A)                                          cost   times
for j ← 2 to n                                             c1
    do key ← A[j]                                          c2
       ▷ Insert A[j] into the sorted sequence A[1 . . j − 1]. 0
       i ← j − 1                                           c4
       while i > 0 and A[i] > key                          c5
            do A[i + 1] ← A[i]                             c6
               i ← i − 1                                   c7
       A[i + 1] ← key                                      c8
```

INSERTION-SORT(A)	*cost*	*times*
for $j \leftarrow 2$ **to** n	c_1	n
do $key \leftarrow A[j]$	c_2	$n-1$
$\triangleright$ Insert $A[j]$ into the sorted sequence $A[1\,..\,j-1]$.	0	$n-1$
$i \leftarrow j-1$	c_4	$n-1$
while $i > 0$ and $A[i] > key$	c_5	$\sum_{j=2}^{n} t_j$
do $A[i+1] \leftarrow A[i]$	c_6	$\sum_{j=2}^{n} (t_j - 1)$
$i \leftarrow i-1$	c_7	$\sum_{j=2}^{n} (t_j - 1)$
$A[i+1] \leftarrow key$	c_8	$n-1$

The running time of the algorithm is

$$\sum_{\text{all statements}} (\text{cost of statement}) \cdot (\text{number of times statement is executed}) .$$

Let $T(n) =$ running time of INSERTION-SORT.

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

Analysis of insertion sort

- زمان اجرای statement خط i ام را که مقداری ثابت است c_i فرض می کنیم.
- سطر سوم یک comment است و running time آن صفر است.
- در هر تکرار forloop، که از مقادیر ۲، ۳، ... ، n می گیرد t_j تعداد تکرار مربوط به while loop است. البته while loop test همیشه یک مرتبه بیشتر از while loop body اجرا می شود که $t_j - 1$ است.
- همواره header یا test تمام loop ها یک مرتبه بیشتر (iteration) اجرا می شوند.

running time وابسته به t_j است که مقدار دقیق آن را قبل از دریافت هر ورودی نمی دانیم