

طراحی و تحلیل الگوریتم ها

دکتر امیر لکی زاده

استادیار گروه مهندسی کامپیوتر دانشگاه قم

➤ تعریف الگوریتم کارا: efficient Algorithm

➤ یک الگوریتم را کارا گویند اگر در زمان چند جمله ای برحسب اندازه ورودی تصمیم پذیر قطعی باشد. (مسئله را حل کنید)

➤ مسائل را از لحاظ زمان لازم برای حل مسئله (زمان به عنوان تابعی از اندازه ورودی مسئله) به دو دسته تقسیم می شوند:

➤ ۱. مسائل تصمیم پذیر قطعی در زمان چند جمله ای:

Deterministic decidable in polynomial time (P)

➤ ۲. مسائل تصمیم پذیر غیر قطعی در زمان چند جمله ای:

NonDeterministic decidable in polynomial Time (NP)

منظور از مسایل تصمیم پذیر قطعی در زمان چند جمله ای (کلاس P):

برای حل آن یک الگوریتم کارا وجود دارد مانند مسئله مرتب سازی یا

الگوریتمی برای آن مسأله وجود دارد که در آن تعداد عملیات برای حل مسأله به صورت یک تابع چند جمله ای از اندازه ورودی مسأله می باشد.

```
int test (int n)
{
  int  $i = 0$  int  $s = 0$ ;
  for ( $i = 1; i \leq c; i++$ )
     $s = s + 1$ 
  return s;
```

$$\} \quad T(n) = 4 + (C + 1) + 2C = 5 + 3C$$

تعداد: محاسبات الگوریتم $T(n)$ ، مستقل از اندازه ورودی می باشد و همواره مقدار ثابتی است. ➤

- مسائل تصمیم پذیر غیر قطعی در زمان چند جمله ای (کلاس NP) الگوریتمی در زمان چند جمله ای وجود دارد که به إزای یک نمونه از آنها تصمیم پذیر است.

مسائل NP_C:

مسائلی هستند که هیچ الگوریتم تصمیم پذیر در زمان چند جمله ای تاکنون برای آنها ارائه نشده است و از طرفی ثابت نیز نشده است که چنین الگوریتمی برای آنها وجود ندارد.

- No efficient Algorithm for an NP – Complete Problem has been found and nobody has ever proven that efficient Algorithm for it can not exist

➤ مثال: فروشنده دوره گرد (Traveling Salesman Problem(TSP)

➤ توجه: تعریف مسائل NP – Complete (NP_C)

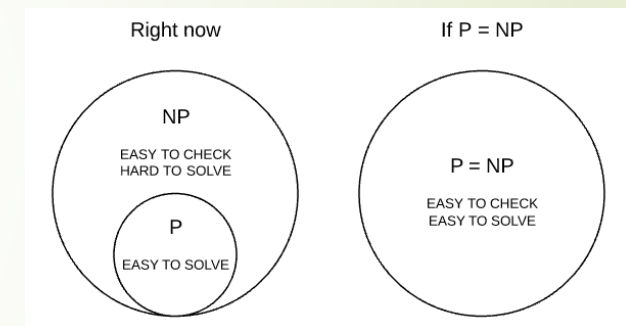
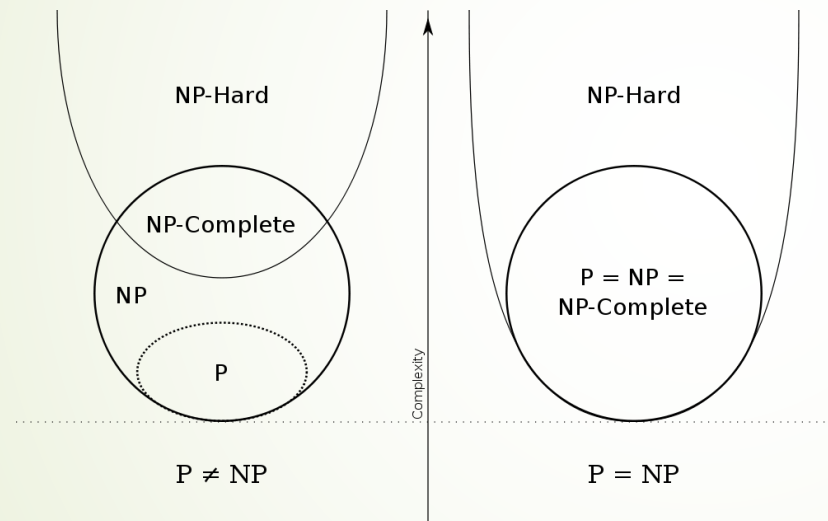
➤ یک مسأله را NP_C گویند اگر:

۱- NP باشد

۲- تمامی مسائل NP در زمان چند جمله ای قابل تبدیل به آن باشند: reducible in polynomial time

► چرا مسائل NP_C مورد علاقه هستند؟

- ۱- اگر یک الگوریتم تصمیم پذیر در زمان چند جمله ای برای یک مسأله NP_C یافت شود آنگاه راه حل فوق قابل تعمیم برای تمام مسائل NP خواهد بود و در نتیجه $P = NP$ می باشد.
- ۲- اگر ثابت کنیم یک مسأله NP_C است می توانیم به جای اینکه وقت خود را صرف پیدا کردن یک الگوریتم بهینه به هدر بدهیم سعی کنیم یک الگوریتم را که یک جواب خوب و نزدیک به بهینه به دست می دهد، ارائه کنیم.



تحليل الگوریتم ها

➤ مقایسه کارایی (efficiently) دو الگوریتم

➤ Faster Computer or Faster Algorithms

Problem: Sorting,

Alg1: Insertion sort $T(n) = 2n^2$

Alg2: Merge sort $T(n) = 50n \cdot \log n$

Suppose have two computers:

A: 10^9 Ins/sec , B: 10^7 Ins/sec, A is 100' faster

$$\frac{2 \cdot (10^6)^2 \text{ instructions}}{10^9 \text{ instructions/second}} = 2000 \text{ seconds ,}$$

while computer B takes

$$\frac{50 \cdot 10^6 \lg 10^6 \text{ instructions}}{10^7 \text{ instructions/second}} \approx 100 \text{ seconds .}$$

تحلیل الگوریتم ها

➤ الگوریتم مرتب سازی درجی: (مناسب برای تعداد کمی از عناصر)

```
Insertion – Sort (A)
for  $j \leftarrow 2$  to  $\text{length}[A]$ 
    { $\text{Key} \leftarrow A[j]; i \leftarrow j - 1;$ 
    while  $((i > 0) \text{ and } (A[i] > \text{key}))$ 
        { $A[i + 1] \leftarrow A[i];$ 
         $i \leftarrow i - 1;$ }
     $A[i + 1] \leftarrow \text{key};$ 
}
```

- تعداد عملیات مرتب سازی درجی در بهترین حالت (best case): $n - 1$

تحليل الگوریتم ها

j	i
2	$1 \rightarrow 1 \quad (1)$
3	$2 \rightarrow 1 \quad (2)$
4	$3 \rightarrow 1 \quad (3)$
\vdots	
n	$n - 1 \rightarrow 1 \quad (n - 1)$

- تعداد عملیات مرتب سازی درجی در بدترین حالت (worst case):

$$1 + 2 + \cdots n - 1 = \frac{(n - 1)n}{2}$$

تحلیل الگوریتم ها

منظور از تحلیل الگوریتم ها پیش بینی منابع مورد نیاز توسط برنامه می باشد این منابع چند دسته اند.

۱- زمان اجرا

۲- حافظه

۳- سخت افزار

۴- پهنای باند ارتباطی

در بحث تحلیل الگوریتم ها به تحلیل زمان اجرای آنها می پردازیم.

توجه: زمان اجرا running time تابعی از اندازه ورودی می باشد.

در مسأله مرتب سازی، اندازه ورودی تعداد بیت ها برای نمایش اعداد می باشد.

زمان اجرای یک الگوریتم بر روی یک ورودی خاص برابر با تعداد عملیات اساسی یا تعداد گام های اولیه که باید اجرا شود.

تحلیل الگوریتم ها

چرا همواره به دنبال مرتبه زمانی در بدترین حالت هستیم؟

- ۱- بدست آوردن یک کران بالا یا بدست آوردن یک تضمین که الگوریتم بدتر از این کران نمی شود.
- ۲- احتمال رخداد فراوان بدترین حالت
- ۳- در بیشتر مواقع زمان اجرا در حالت میانگین (average case) به اندازه بدترین حالت (worst case) می باشد.