



نیمسال اول ۱۴۰۰-۱۴۰۱

مدرس: دکتر مجتبی رفیعی

## ساختمان داده‌ها و الگوریتم‌ها

### جلسه ۳

نگارنده: سعید شریعتی

۹ مهر ۱۴۰۰

## فهرست مطالب

۱ مثال‌هایی از رده‌های مختلف مسائل محاسباتی

## ۱ مثال‌هایی از رده‌های مختلف مسائل محاسباتی

همانطور که در جلسه قبل مطرح شد، رده‌های مسائل محاسباتی را می‌توان در دو رده کلی تقسیم بندی کرد. شکل ۱، این رده‌های محاسباتی را نشان می‌دهد. در این گزارش سعی بر آن است تا برای هر یک از رده‌های مسائل محاسباتی، مثال‌هایی تشریح شود.

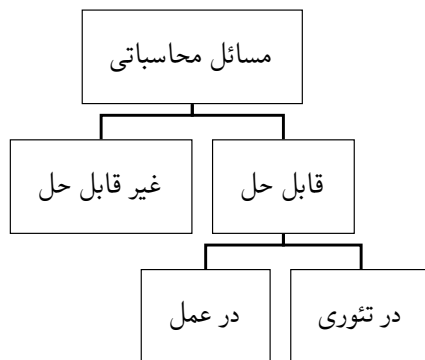
در ادامه مثال‌های منتخب برای هر یک از رده‌های مسائل محاسباتی آورده شده است.

- **غیر قابل حل:** مسائلی که برای حل آنها الگوریتمی وجود ندارد. مثل مسئله توقف (Halting problem) که در جلسه قبل آن را تشریح کردیم.

- **قابل حل:** مسائلی که الگوریتمی برای حلشان موجود است که خود به دو دسته تقسیم می‌شوند:

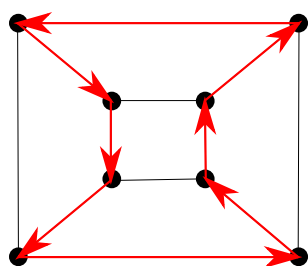
- **در ثنوری:** مسائلی که حل می‌شوند ولی پیچیدگی خاصی دارند که باعث می‌شود برای نمونه‌های بزرگ با کامپیوترهای امروزی، کارایی لازم برای اجرای عملیاتی را نداشته باشند. مثال‌های زیر نمونه‌هایی از این رده مسائل محاسباتی هستند:

- **Boolean Satisfiability Problem (SAT/B-SAT):** آیا ارزش‌دهی (True, False) برای لیتراهای یک فرمول بولی می‌توان یافت به نحوی که آن فرمول بولی به مقدار True ارزشیابی شود.

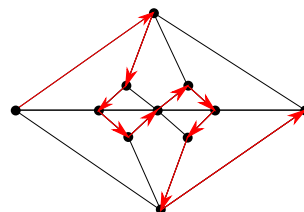


شکل ۱: رده‌بندی مسائل محاسباتی

- مسئله مسیر همیلتونی (Hamiltonian Path Problem): پیدا کردن یک مسیر در گراف (جهت‌دار یا غیرجهت‌دار) به طوریکه هر نود گراف دقیقاً یکبار ملاقات شود. شکل ۲ را مشاهده کنید.
- مسئله دور همیلتونی (Hamiltonian Cycle Problem): پیدا کردن یک دور در گراف (جهت‌دار یا غیرجهت‌دار) به طوریکه هر نود گراف دقیقاً یکبار ملاقات شود. شکل ۲ را مشاهده کنید.



(ب) مسیر همیلتونی دارد و دور همیلتونی هم دارد



(آ) مسیر همیلتونی دارد ولی دور همیلتونی ندارد

شکل ۲: مسائل قابل حل در تئوری - مسیر همیلتونی و دور همیلتونی

- در عمل: مسائلی که الگوریتمی برای حل کردن دارند و هم برای نمونه‌های بزرگ و هم کوچک می‌شود آن‌ها را توسط کامپیوترهای امروزی حل کرد. مثال:
- مرتب کردن یک لیست: یک لیست از اعداد داده شده و قرار است یک لیست مرتب شده از آنها بدست آوریم. شبکه‌کد مربوط به مرتب سازی درجی به عنوان یک نمونه منتخب از مرتب سازی در الگوریتم ۱ نمایش داده شده است.

---

**Algorithm 1** Insertion-Sort( $A[1 \dots n]$ )

---

```

1: for  $j = 2$  to  $n$  do
2:    $key \leftarrow A[j]$ 
3:    $i \leftarrow j - 1$ 
4:   while  $i > 0$  and  $A[i] > key$  do
5:      $A[i + 1] \leftarrow A[i]$ 
6:      $i \leftarrow i - 1$ 
7:    $A[i + 1] \leftarrow key$ 
  
```

---

- جست و جو کردن یک مقدار در یک لیست: یک لیست اعداد داده شده (می تواند مرتب نباشد)، قرار است یک عدد را در آن جست و جو کنیم. شبکه‌کد مربوط به جستجوی خطی به عنوان یک نمونه منتخب برای جستجو در الگوریتم ۲ نمایش داده شده است.

---

**Algorithm 2** Linear-Search( $A[1 \dots n], x$ )

---

```

1: for  $i = 1$  to  $n$  do
2:   if ( $A[i] == x$ ) then
3:     return True
4: return False

```

---

#### نکته

تمرکز ما در درس ساختمان داده‌ها و الگوریتم‌ها، روی رده مسائل قابل حل در عمل می‌باشد.

## تجربیات لاتک‌نویسی

- نوشتن متن انگلیسی به صورت صحیح. متن انگلیسی This is some text را اگر در لاتک بنویسید، متن به صورت زیر چاپ خواهد شد:

text some is This

برای جلوگیری از این مشکل باید از دستور `\lr` استفاده کنید. مثال:

```
\lr{This is some text}
```

خروجی این کد می‌شود:

This is some text

یعنی دیگر متن وارون نخواهد شد.

- توپر کردن متن. دستور `\textbf` متن This is some text را توپر نمایش می‌دهد.

```
\lr{\textbf{This is some text}}
```

خروجی دستور فوق به صورت زیر است:

**This is some text**

- ساختن لیست. با استفاده از کد زیر می‌شود یک لیست با آیتم‌های `item۱`، `item۲`، `item۳`، `item۴` ساخت:

```

\begin{itemize}
  \item item1
  \item item2
  \item item3
  \item item4
\end{itemize}

```

کد بالا، خروجی زیر را تولید می‌کند:

- item1
- item2
- item3
- item4

همچنین می‌توان چند لیست را به صورت زیر داخل هم قرار داد:

```
\begin{itemize}
  \item item1
  \item item2
  \item item3
  \begin{itemize}
    \item item3-1
    \item item2-2
    \item item3-3
    \item item4-4
  \end{itemize}
\end{itemize}
```

خروجی حاصل از اجرای کد فوق به صورت زیر می‌باشد:

- item1
- item2
- item3
  - \* item3-1
  - \* item2-2
  - \* item3-3
  - \* item4-4

اگر می‌خواهید که لیست شماره‌بندی یا حروف‌بندی شود می‌توانید به جای `itemize` از `enumerate` استفاده کنید. مثال:

```
\begin{enumerate}
  \item item1
  \item item2
  \item item3
  \begin{enumerate}
    \item item3-1
    \item item2-2
    \item item3-3
    \item item4-4
  \end{enumerate}
\end{enumerate}
```

خروجی حاصل از اجرای کد فوق به صورت زیر است:

1. item1
2. item2

### 3. item3

- (a) item3-1
- (b) item2-2
- (c) item3-3
- (d) item4-4

- اضافه کردن تصاویر به سند. برای اضافه کردن یک تصویر به متن می‌توانید از دستور زیر استفاده کنید:

```
\begin{figure}[!h]
\centering
\includegraphics[width=0.7\linewidth]{cat.jpg}
\caption{cat}
\end{figure}
```

به جای 0.7 در width=0.7 باید طول تصویر خود را مشخص کنید، عدد 0.7 در این کد به این معنی است که طول تصویر حداکثر هفتاد درصد طول یک خط در این فایل خواهد بود.

به جای cat.jpg باید اسم فایل تصویر خود را مشخص کنید. تصویر باید داخل پروژه شما باشد تا لاتک بتواند آن را پیدا کند.

متن داخل پرانتزهای caption هم به همراه شماره شکل، زیر تصویر نمایش داده خواهد شد.

دستور centering در خط دوم، تصویر را در وسط صفحه قرار می‌دهد.

کاراکتر !h در خط اول برای این است که تصویر در همان جای متن نمایش داده شود که تعریف شده است. می‌توانید از کاراکتر t برای نمایش تصویر در بالای صفحه یا از کاراکتر b برای نمایش تصویر در پایین صفحه استفاده کنید.

نتیجه کد بالا در پروژه من به صورت زیر است:



شکل ۳: cat