«پاسخ تمرین اول الگوریتم»

(۱) متنی به طول n و الگویی به طول m در نظر بگیرید. برای یافتن الگو به روش brute_force مثالی از بدترین حالت بزنید؟ در این حالت دقیقاً چند مقایسه بین کاراکترها رخ خواهد داد؟

الگوریتم brute force برای string matching:

دو اشاره‌گر در نظر می‌گیریم، یکی برای متن و دیگری برای الگو. مقدار اولیه هر دو اشاره‌گر، کاراکتر اول متن و کاراکتر اول الگو است. حال شروع به مقایسه می‌کنیم. اگر کاراکتر متن با کاراکتر الگو یکسان بود، هر دو اشاره‌گر را یک واحد جلو می‌بریم. اگر کاراکتر متن با کاراکتر الگو یکسان نبود، اشاره‌گر متن یک واحد جلو رفته و اشاره‌گر الگو reset می‌شود (به ابتدای الگو باز می‌گردد) و از ابتدای رشته تا زمانی که کاراکترهای یکسان پیدا کند، یکی یکی جلو می‌رود.

پیچیدگی زمانی برای الگو به طول m و رشته به طول n ← $O(m(n-m+1))$

```
m ← length(pattern)
n ← length(text)

for i=0 to i=n-m          → (n-m-0+1) بار
    j=0
    while (j<m)           → m بار
        if text[i+j]!=pattern[i]:     (در بدترین حالت)
            break;
    if j==m:
        print("Pattern found")
```

$\Downarrow$

$\boxed{m(n-m+1)}$ تا مقایسه در بدترین حالت

Subject .... Year .... Month .... Day ....

مثال بدترین حالت ← یک رشته به فرم $\underbrace{aaa\dots a}_{\text{تا } n}$

← یک الگو به فرم $\underbrace{aaa\dots ab}_{\text{تا } m-1}$

---

(2) مسئله شمردن زیر رشته‌هایی که با A آغاز و به B ختم می‌شوند در یک متن دلخواه را در نظر بگیرید.

a) برای این مسئله راه حل brute force طراحی کرده و پیچیدگی زمانی آن را به دست آورید.

b) یک راه حل بهتر از brute force هم برای این مسئله ارائه دهید.

a) راه حل brute force، دنبال اولین A می‌گردیم. بعد از آن، هر چه B دیدیم، به تعداد B ها به تعداد زیر رشته‌ها اضافه می‌کنیم.
(تا انتهای رشته)

سپس دنبال A بعدی می‌گردیم و باز هم هر چه B تا انتهای رشته دیدیم، به تعداد B ها به تعداد زیر رشته‌ها اضافه می‌کنیم. این کار را (یافتن A بعدی) تا یک کاراکتر قبل از آخرین کاراکتر رشته تکرار می‌کنیم:

```
n ← length(text), counter ← 0
for i=0 to i = n-2 :
    j = i+1
    if text[i] == 'A' :
        while (j<n) :
            if text[j] == 'B' :
                counter++
            j++
print(counter)
```

تحلیل پیچیدگی زمانی: حلقه while در بدترین حالت $n-(i+1)+1$ بار تکرار می‌شود و i هم از 0 تا $n-2$ است:

$$\Rightarrow O\left(\sum_{i=0}^{n-2} (n-i)\right)$$

$$= n + n-1 + \dots + 2$$

$$= n(n+1)/2 - 1 = \boxed{O(n^2)}$$

مثال بدترین حالت ← رشته ورودی تماماً A باشد. (به فرم $\underbrace{A\dots A}_{\text{تا } n}$ باشد).

ط) از دو شمارنده یکی برای تعداد A ها و دیگری برای تعداد زیر رشته‌هایی که با A شروع و به B ختم می‌شوند در نظر می‌گیریم و به هر دو مقدار 0 می‌دهیم. در ادامه شروع به پیمایش رشته متن می‌کنیم. اگر A دیدیم، شمارندهٔ A ها را یکی زیاد می‌کنیم و اگر B دیدیم، به شمارندهٔ تعداد زیر رشته‌ها، مقدار کنونی شمارندهٔ A ها را اضافه می‌کنیم.

```
counter ← 0
A_counter ← 0
for character in text:
    if character == 'A':
        A_counter++
    if character == 'B':
        counter += A_counter
print counter
```

پیچیدگی زمانی: فقط یک حلقه داریم که روی تمام متن پیمایش می‌کند. پس $O(n)$ است.

---

3) فرض کنید $x_1 < x_2 < \dots < x_n$ اعدادی حقیقی باشند که نشان دهندهٔ مختصات n روستا در یک جاده‌اند. قرار است یک ادارهٔ پست در یکی از این روستاها ساخته شود.

a) الگوریتمی بهینه برای آن که میانگین فاصلهٔ بین روستاها و ادارهٔ پست کمینه باشد طراحی کنید.

b) الگوریتمی بهینه برای آن که بیشینهٔ فاصلهٔ یک روستا تا ادارهٔ پست کمینه باشد طراحی کنید.

a) فرض کنید ، اداره پست در نقطه x_i واقع شود. حال 2 حالت را بررسی می‌کنیم:

حالت (I): n زوج باشد. ابتدا میانگین فاصله بین روستاها و اداره پست را محاسبه می‌کنیم

x_i ← مکان اداره پست

⇐ میانگین فاصله روستاها تا اداره پست $= \frac{1}{n} \sum_{j=1}^{n} |x_j - x_i|$

از آن جایی که n ثابت است، برای کمینه شدن میانگین فاصله روستاها تا اداره پست کافیست $\sum_{j=1}^{n} |x_i - x_j|$ کمینه شود.

حالت $n=2$ را در نظر بگیرید، در این حالت حاصل $\sum$ برابر $|x_1 - x_2|$ خواهد شد. (اگر x_1 را انتخاب کنیم، $|x_1 - x_1|$، صفر می‌شود و اگر x_2 را انتخاب کنیم $|x_2 - x_2|$ برابر صفر است ⇐ هم فقط $|x_1 - x_2|$ می‌ماند.)

حال اگر مکان اداره پست در بازه $[x_1, x_2]$ باشد، حاصل $\sum$ که کمتر یا مساوی $|x_1 - x_2|$ خواهد بود. (در این حالت دقیقاً برابر $|x_1 - x_2|$ است) ولی اگر مکان اداره پست خارج از بازه $[x_1, x_2]$ باشد، حاصل $\sum$، بزرگ‌تر از $|x_1 - x_2|$ می‌شود. ⇐ برای آن که $\sum$ کمینه شود، باید مکان اداره پست در بازه $[x_1, x_2]$ باشد. حال حالت کلی را در نظر بگیرید:

$$\sum_{j=1}^{n} |x_j - x_i| = \left(|x_1 - x_i| + |x_n - x_i|\right) + \left(|x_2 - x_i| + |x_{n-1} - x_i|\right) + \cdots + \left(|x_{n/2} - x_i| + |x_{n/2+1} - x_i|\right)$$

برای کمینه شدن حاصل $\sum$، حاصل هر یک از پرانتزها باید کمینه باشد. پس اگر اشتراک

بگیریم، خواهیم داشت :

مکان اداره پست $\in [x_1, x_n], [x_2, x_{n-1}], [x_3, x_{n-2}], \dots, [x_{n/2}, x_{n/2+1}]$

که حاصل این اشتراک این است که مکان اداره پست $\in [x_{n/2}, x_{n/2+1}]$

$\Leftarrow$ در حالتی که n زوج باشد، هر یک از $x_{n/2}$ و $x_{n/2+1}$ می‌توانند مکان اداره پست باشند.

حالت II : n فرد باشد. کاملاً شبیه با استدلال حالت قبلی این بار داریم :

$$\sum_{j=1}^{n} |x_j - x_i| = \left(|x_1 - x_i| + |x_n - x_i|\right) + \left(|x_2 - x_i| + |x_{n-1} - x_i|\right) + \dots + \left(|x_{\lceil n/2 \rceil} - x_i|\right)$$

که برای کمینه شدن $\sum$ ، اشتراک به این صورت است :

مکان اداره پست $\in [x_1, x_n], [x_2, x_{n-1}], [x_3, x_{n-2}], \dots [x_{\lceil n/2 \rceil}, x_{\lceil n/2 \rceil}]$

که حاصل اشتراک یعنی مکان اداره پست باید $x_{\lceil \frac{n}{2} \rceil}$ باشد.

$\Leftarrow$ مکان اداره پست در هر دو حالت همان میانه است و اگر مکان‌ها مرتب شده باشند در $O(1)$ میانه پیدا می‌شود و در صورت مرتب نبودن، در $O(n\log n)$.

b) فرض کنید اندازه لیست در مکان x_i باشد. در این صورت چون x_i ها به شکل صعودی در صورت سؤال آمده اند، پس بیشترین فاصله تا اندازه لیست یا $x_n - x_i$ است یا $x_i - x_1$. پس بالیستی x_i ای انتخاب کنیم که به یک اندازه تا هر دوی x_1 و x_n فاصله داشته باشد. که در این صورت بالیستی میانگین x_1, x_n را در نظر بگیریم حال کافیست روی لیست مکان ها پیمایش کرده و نزدیک ترین مقدار به $\frac{x_1+x_n}{2}$ را پیدا کنیم. برای این کار کافیست تا اولین x_i بزرگ تر از $\frac{x_1+x_n}{2}$ جلو رفته و آن را پیدا کنیم. در این صورت جواب $\min(x_n - x_{i-1}, x_i - x_1)$ خواهد بود:

```
m ← (x_1 + x_n) / 2
i ← 1
while x_i < m:
    i++
return min(x_n - x_{i-1}, x_i - x_1)
```

پیچیدگی زمانی این الگوریتم در بدترین حالت $O(n)$ است.

یک مثال از بدترین حالت ← $\underbrace{1,1,1,\ldots,1}_{n-1 \text{ تا}},2$

* اگر مکان های داده شده مرتب نباشند، ابتدا باید آن ها را مرتب کنیم، در این صورت پیچیدگی زمانی برابر است با:

$$O(\underbrace{n\log n}_{\text{مرتب سازی}} + \underbrace{n}_{\text{یافتن } x_i}) = O(n\log n)$$

Subject .... Year .... Month .... Day ....

4) a) فاصله منهتنی بین دو نقطه $P_1(x_1,y_1)$ و $P_2(x_2,y_2)$ در صفحه مختصات کارتزین به شکل زیر تعریف می‌شود:

$$d_M(P_1,P_2)=|x_1-x_2|+|y_1-y_2|$$

هر یک از اصل‌های زیر را برای d_M که تابع محاسبه فاصله منهتنی است، اثبات نمایید.

i) به ازای هر دو نقطه P_1 و P_2، $d_M(P_1,P_2)\geq 0$ و $d_M(P_1,P_2)=0$ اگر و تنها اگر $P_1=P_2$.

ii) $d_M(P_1,P_2)=d_M(P_2,P_1)$

iii) به ازای هر 3 نقطه P_1، P_2 و P_3: $d_M(P_1,P_2)\leq d_M(P_1,P_3)+d_M(P_3,P_2)$

b) تمام نقاطی که فاصله منهتنی شان تا مبدأ $(0,0)$ برابر 1 است را رسم کنید.
همچنین تمام نقاطی که فاصله اقلیدسی شان تا مبدأ، برابر 1 است را رسم کنید.

c) درست یا غلط: برای مسأله نزدیک‌ترین جفت، استفاده از فاصله منهتنی با استفاده از فاصله اقلیدسی فرقی ندارد.

a) i) $d_M(P_1,P_2) = \underbrace{|x_1-x_2|}_{\geq 0} + \underbrace{|y_1-y_2|}_{\geq 0} \Rightarrow d_M(P_1,P_2)\geq 0$

if $P_1=P_2 \Rightarrow x_1=x_2\,,\ y_1=y_2 \Rightarrow d_M(P_1,P_2)=0$

if $d_M(P_1,P_2)=0 \Rightarrow |x_1-x_2|+|y_1-y_2|=0$

جمع دو قدر مطلق زمانی 0 است که هر دو صفر باشند $\Leftarrow$ $x_1=x_2$ ، $y_1=y_2$

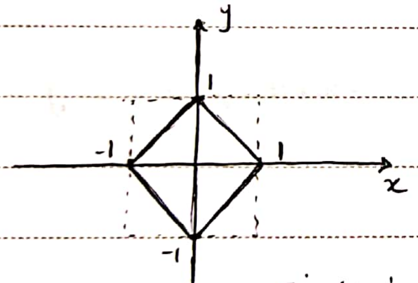
Subject .... Year .... Month .... Day ....

(ii
$$d_M(P_1,P_2) = |x_1-x_2| + |y_1-y_2|$$
$$d_M(P_2,P_1) = |x_2-x_1| + |y_2-y_1| = |x_1-x_2| + |y_1-y_2|$$
$$= d_M(P_1,P_2)$$

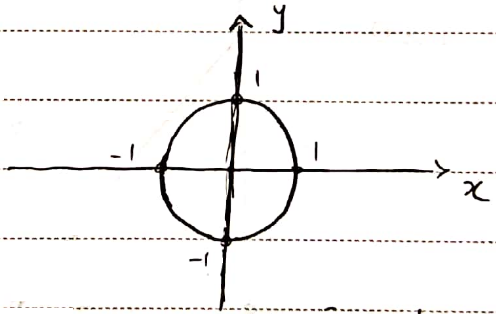
(iii
$$d_M(P_1,P_2) = |x_1-x_2| + |y_1-y_2|$$
$$= |(x_1-x_3)+(x_3-x_2)| + |(y_1-y_3)+(y_3-y_2)|$$
$$\leq |x_1-x_3| + |y_1-y_3| + |x_3-x_2| + |y_3-y_2|$$
$$\Rightarrow d_M(P_1,P_2) \leq d_M(P_1,P_3) + d_M(P_3,P_2)$$

(b
فاصلهٔ منهتنی $\longrightarrow$ $P_1=(0,0)$, $P_2=(x,y) \Rightarrow |x|+|y|=1$

فاصلهٔ اقلیدسی $\longrightarrow$ $\sqrt{(x-0)^2+(y-0)^2}=1 \Rightarrow \sqrt{x^2+y^2}=1$



فاصلهٔ منهتنی



فاصلهٔ اقلیدسی

c) غلط است. اثبات به وسیلهٔ مثال نقض: نقاط را به صورت $P_1(0,0)$ ، $P_2(1,0)$ و $P_3(\frac{1}{2},\frac{3}{4})$ در نظر بگیرید. در این صورت برای فاصلهٔ اقلیدسی داریم:

$$d_E(P_1,P_2)=1 \ , \ d_E(P_1,P_3)=d_E(P_2,P_3)=\sqrt{\left(\frac{1}{2}\right)^2+\left(\frac{3}{4}\right)^2}=\sqrt{\frac{13}{16}}<1$$

$\Rightarrow$ closest pair $= (P_2,P_3)$ یا (P_1,P_3)

برای فاصلهٔ منهتنی داریم:

$$d_M(P_1,P_2)=1 \ , \ d_M(P_1,P_3)=d_M(P_2,P_3)=\frac{1}{2}+\frac{3}{4}=\frac{5}{4}>1$$

$\Rightarrow$ closest pair $= (P_1,P_2) \Rightarrow$ جواب با فاصلهٔ اقلیدسی فرق کرد.

Subject .... Year .... Month .... Day ....

5) مسأله زیر یک نمونه خوب از مسئله برنامه ریزی خطی است را در نظر بگیرید:

- مقدار $3x+5y$ را با شرایط
$$\begin{cases} x+y \leq 4 \\ x+3y \leq 6 \\ x \geq 0 \,,\; y \geq 0 \end{cases}$$
بیشینه کنید.

a) در صفحه مختصات کارتزین تمام نقاط‌های محدودیت‌های مسأله را ارضا می‌کنند، رسم نمایید.

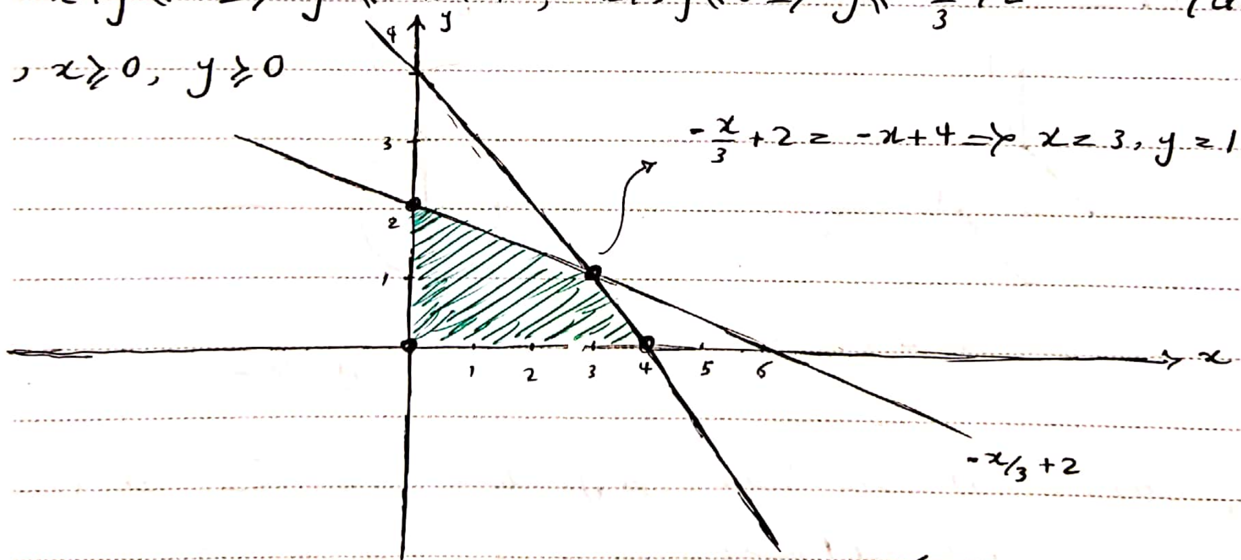
b) نقاط اکسترمم را مشخص کنید.

c) این مسأله بهینه سازی را با استفاده از قضیه زیر حل کنید:

- قضیه: یک مسأله برنامه ریزی خطی که نمودار تمام نقاطی که محدودیت‌های آن را ارضا می‌کنند، تهی نیست، حتماً جواب دارد و جوابش را می‌توان در یکی از نقاط اکسترمم آن یافت.

a) $x+y \leq 4 \Rightarrow y \leq -x+4 \,,\quad x+3y \leq 6 \Rightarrow y \leq -\frac{x}{3}+2$

و $x \geq 0 \,,\; y \geq 0$



b) با توجه به نمودار نقاط اکسترمم عبارتند از: $(3,1),(0,0),(0,2),(4,0)$

c)

نقاط	3x+5y
(3,1)	14
(0,0)	0
(0,2)	10
(4,0)	12

⇐ بیشترین مقدار $3x+5y$ با رعایت شرایط مسأله برابر 14 می‌شود.

6. برای گراف همبندی که توسط ماتریس مجاورت نمایش داده می‌شود، الگوریتمی تعیین کنید که تشخیص دهد گراف دور اویلری دارد یا خیر؟ پیچیدگی زمانی الگوریتم را نیز بدست آورید.

می‌دانیم یک گراف دور اویلری دارد اگر و تنها اگر درجه تمام رئوس آن زوج باشد. ⇐ کافیست در ماتریس مجاورت، تعداد 1 های هر سطر را بشماریم، اگر این تعداد فرد بود ⇐ درجه آن رأس فرد است ⇐ دور اویلری نداریم

```
for i=1 to i=n:
    ones_count ← 0
    for j=1 to j=n:
        if adj_mat[i,j] == 1 :
            ones_count++
    if ones_count % 2 == 1 :
        print ('No Eulerian circuit exists')
        return
print ('Eulerian circuit exists')
```

حلقه دوم، n بار اجرا می‌شود. حلقه اول در بدترین حالت، n بار اجرا می‌شود ⇐ پیچیدگی زمانی، $O(n^2)$ است.

* بدترین حالت ← فقط سطر آخر، تعداد فردی 1 داشته باشد یا تمام رئوس درجه زوج داشته باشند.

7) مثالی از مسأله‌ی تخصیص بزنید که راه حل بهینه‌ی آن شامل کوچکترین عنصر ماتریس هزینه‌اش نشود.

مسأله‌ی تخصیص: n مأمور داریم و n وظیفه. هر مأمور می‌تواند هر وظیفه‌ای را با هزینه‌ای مشخص انجام دهد. حال هدف این است وظایف را به هر مأمور یک وظیفه بدهیم که: - تمام وظایف انجام شوند. - انجام وظایف کمترین هزینه را داشته باشد.

مثال ← فرض کنید علی، حسن و حسین، 3 وظیفه‌ی شستن ماشین، جارو زدن خانه و درست کردن شام را طبق ماتریس هزینه‌ی زیر انجام می‌دهند:

	جارو	ماشین	شام
علی	1500	4000	4500
حسن	2000	6000	3500
حسین	2000	4000	2500

برای یافتن تخصیص با کمترین هزینه از الگوریتم مانکرز استفاده می‌کنیم:

1) کمترین مقدار در هر سطر را از تمام عناصر آن سطر کم می‌کنیم:

0	2500	3000
0	4000	1500
0	2000	500

2) کمترین مقدار در هر ستون را از تمام عناصر آن ستون کم می‌کنیم:

0	500	2500
0	2000	1000
0	0	0

Subject .... Year .... Month .... Day ....

③ حال تمام 0ها را با خطوط عمودی و افقی پوشش می‌دهیم و این کار با 2 خط ممکن است

0	500	2500
0	2000	1000
0	0	0

④ چون تعداد خطوط کمتر از تعداد وظایف است هنوز به جواب بهینه نرسیده‌ایم. سپس کمترین مقدار، cover نشده را از سطرهای cover نشده کم می‌کنیم، سپس کمترین مقدار cover نشده را به تمام ستون‌های cover شده اضافه می‌کنیم:

I →

-500	0	2000
-500	1500	500
0	0	0

II →

0	0	2000
0	1500	500
500	0	0

حال دوباره تمام 0 ها را با خطوط افقی و عمودی cover می‌کنیم. این بار برای cover کردن 0 ها، به 3 خط نیاز داریم ⟸ به جواب رسیده‌ایم:

0	0✓	2000
0✓	1500	500
500	0	0✓

برای مشخص کردن تخصیص بهینه باید از بین 0 ها، آن‌ها را انتخاب کنیم که با یکدیگر conflict نداشته باشند ⟸ با 0ای که کمترین محدودیت را دارد شروع می‌کنیم و پس از انتخاب، سطر نظیرش را در انتخاب‌های بعدی در نظر نمی‌گیریم. حال به جدول اصلی نگاه کرده و مقادیر اصلی را جایگذین می‌کنیم ⟸ کمترین هزینه در این مسئله برابر $4000+2000+2500=8500$ است. در صورتی که علی ماشین بشوید، حسن جارو کند و حسین شام درست کند.

کمترین مقدار ماتریس هزینه، 1500 بود. همان طور که در جواب بهینه مشخص است، این مقدار در جواب بهینه وجود ندارد.

۴) فرض کنید در یک صفحه شطرنج 8×8 ، بخواهیم 8 وزیر قرار دهیم. در این صورت مطلوبست:

a) تعداد حالات مختلف قرارگیری وزیرها به شکلی که هیچ دو وزیری در یک خانه نباشند؟

b) تعداد حالات مختلف قرارگیری وزیرها به طوری که هیچ دو وزیری در سطر یکسانی نباشند؟

c) تعداد حالات مختلف قرارگیری وزیرها به طوری که هیچ دو وزیری در یک سطر یا در یک ستون نباشند؟

زمان تقریبی جستجو در بین حالات مختلف را برای هر سه پرسش a ، b و c بیابید. با این فرض، ثانیه 10 میلیارد حالت قابل بررسی باشد.

a) $\binom{64}{8} = 4\,426\,165\,368 \simeq 0.4\,s$

b) $8^8 = 16\,777\,216 \simeq 1.6\,ms$

c) $8! = 40\,320 \simeq 40\,\mu s$