# طراحی و تحلیل الگوریتم


استاد:

**دکتر زاهد رحمتی**

تدریسیاران:

**داریوش کاظمی**

**اشکان ودادی**


ترم دوم ۱۴۰۰

# سوالات میانترم درس طراحی و تحلیل الگوریتم

ترم اول ۱۴۰۰

## سوال ۱: اولویت‌های مردها و زن‌ها در جدول زیر نشان داده شده است.

| مرد | اولویت ۱ | اولویت ۲ | اولویت ۳ |
|---|---|---|---|
| A | D | E | F |
| B | D | F | E |
| C | F | D | E |

| زن | اولویت ۱ | اولویت ۲ | اولویت ۳ |
|---|---|---|---|
| D | C | A | B |
| E | C | B | A |
| F | A | B | C |

| مچینگ |
|---|
| A-E |
| B-F |
| C-D |

آیا مچینگ زیر پایدار است؟چرا؟

## سوال ۱: اولویت‌های مردها و زن‌ها در جدول زیر نشان داده شده است.

| مرد | اولویت ۱ | اولویت ۲ | اولویت ۳ |
|---|---|---|---|
| A | D | E | F |
| B | D | F | E |
| C | F | D | E |

| زن | اولویت ۱ | اولویت ۲ | اولویت ۳ |
|---|---|---|---|
| D | C | A | B |
| E | C | B | A |
| F | A | B | C |

| مچینگ |
|---|
| A-E |
| B-F |
| C-D |

## آیا مچینگ زیر پایدار است؟چرا؟

پایدار است.
به A، E داده شده است که با اینکه D اولویت بالاتر آن است، اما D اولویت اول خود را گرفته است. به E هم با این که A داده شده اما B و C اولویت‌های بالاتری را گرفته اند.
به B، F داده شده است، اولویت D بالاتر است ولی D اولویت اول خود را گرفته است. F نیز اولویت اولش A است ولی A اولویت بالاتری را گرفته و نمیخواهد جابه‌جا شود.
C اولویت اول خود را گرفته و D نیز با این که F اولویت بالاتری برایش دارد، اما اولویت F برای آن پایین‌تر است، پس تعویض انجام نمی‌شود.
در نتیجه این matching، یک مچینگ پایدار است. زیرا که به هرکس حداقل یکی assign شده و زوج ها نیز نمی‌توانند در وضعیت بهتری باشند.

**سوال۲: الف)** رابطه بازگشتی $T(n) = 4T(\sqrt{n}) + (logn)^2$ را حل کنید؟

$T(n) = 4T(\sqrt{n}) + (\log n)^2$

## **سوال۲: الف)** رابطه بازگشتی $T(n) = 4T(\sqrt{n}) + (logn)^2$ را حل کنید؟

$T(n) = 4T(\sqrt{n}) + (\log n)^2$

برای حل این سوال از تغییر متغیر باید استفاده کنیم.  کافی است $n = 2^m$.

**راهنمایی:** در سوالاتی که مقدار  درون تابع T به صورت رادیکالی یا توانی بود، تغییر متغیر را آزمایش کنید.

## **سوال۲: الف)** رابطه بازگشتی $T(n) = 4T(\sqrt{n}) + (logn)^2$ را حل کنید؟

$T(n) = 4T(\sqrt{n}) + (\mathrm{logn})^2$

برای حل این سوال از تغییر متغیر باید استفاده کنیم.  کافی است $n = 2^m$.

**راهنمایی:** در سوالاتی که مقدار  درون تابع T به صورت رادیکالی یا توانی بود، تغییر متغیر را آزمایش کنید.

$T(2^m) = 4T\left(2^{\frac{m}{2}}\right) + m^2$

## **سوال۲: الف)** رابطه بازگشتی $T(n) = 4T(\sqrt{n}) + (logn)^2$ را حل کنید؟

$T(n) = 4T(\sqrt{n}) + (\log n)^2$

برای حل این سوال از تغییر متغیر باید استفاده کنیم. کافی است $n = 2^m$.

**راهنمایی:** در سوالاتی که مقدار درون تابع T به صورت رادیکالی یا توانی بود، تغییر متغیر را آزمایش کنید.

$T(2^m) = 4T\left(2^{\frac{m}{2}}\right) + m^2$

$S(m) = 4S\left(\frac{m}{2}\right) + m^2$

## **سوال۲: الف)** رابطه بازگشتی $T(n) = 4T(\sqrt{n}) + (logn)^2$ را حل کنید؟

$T(n) = 4T(\sqrt{n}) + (\log n)^2$

برای حل این سوال از تغییر متغیر باید استفاده کنیم.  کافی است $n = 2^m$.

**راهنمایی:** در سوالاتی که مقدار  درون تابع T به صورت رادیکالی یا توانی بود، تغییر متغیر را آزمایش کنید.

$T(2^m) = 4T\left(2^{\frac{m}{2}}\right) + m^2$

$S(m) = 4S\left(\frac{m}{2}\right) + m^2 \text{ (case 2 } \rightarrow \log_b a = 2)$

$S(m) = \theta(m^2 logm)$

## سوال۲: الف) رابطه بازگشتی $T(n) = 4T(\sqrt{n}) + (logn)^2$ را حل کنید؟

$T(n) = 4T(\sqrt{n}) + (\log n)^2$

برای حل این سوال از تغییر متغیر باید استفاده کنیم. کافی است $n = 2^m$.

**راهنمایی:** در سوالاتی که مقدار درون تابع T به صورت رادیکالی یا توانی بود، تغییر متغیر را آزمایش کنید.

$T(2^m) = 4T\left(2^{\frac{m}{2}}\right) + m^2$

$S(m) = 4S\left(\frac{m}{2}\right) + m^2 \qquad (\text{case } 2 \ \rightarrow \log_b a = 2)$

$S(m) = \theta(m^2 logm)$

$T(n) = \theta(\log^2 n \ \times \log(logn))$

## **سوال۲: الف)** رابطه بازگشتی $T(n) = 4T(\sqrt{n}) + (logn)^2$ را حل کنید؟

$T(n) = 4T(\sqrt{n}) + (\text{logn})^2$

برای حل این سوال از تغییر متغیر باید استفاده کنیم. کافی است $n = 2^m$.

**راهنمایی:** در سوالاتی که مقدار درون تابع T به صورت رادیکالی یا توانی بود، تغییر متغیر را آزمایش کنید.

$T(2^m) = 4T\left(2^{\frac{m}{2}}\right) + m^2$

$S(m) = 4S\left(\frac{m}{2}\right) + m^2 \qquad (\text{case } 2 \ \rightarrow \log_b a = 2)$

$S(m) = \theta(m^2 logm)$

$T(n) = \theta(\log^2 n \ \times \log(logn))$

$T(n) \in \theta(g(x)) \rightarrow T(n) \in O(g(x))$

$T(n) = O(\log^2 n \ \times \log(logn))$

**سوال۲: ب)** رابطه بازگشتی $T(n) = T(n-2) + \frac{1}{logn}$ را حل کنید؟

اول سعی کنیم T را در هر مرحله باز کنیم.

$$T(n) = T(n-2) + \frac{1}{logn} = T(n-4) + \frac{1}{\log(n-2)} + \frac{1}{logn} = \cdots$$

## سوال۲: ب) رابطه بازگشتی $T(n) = T(n-2) + \frac{1}{logn}$ را حل کنید؟

اول سعی کنیم T را در هر مرحله باز کنیم.

$$T(n) = T(n-2) + \frac{1}{logn} = T(n-4) + \frac{1}{\log(n-2)} + \frac{1}{logn} = \cdots$$

اما چون دوتا دوتا در حال کاهش است زوج و فرد بودن در مسئله تاثیر گذار است:

$$T(n) = \begin{cases} T(0) + \sum_{k=1}^{n/2} \frac{1}{\log(2k)} & \text{if } n \text{ is even} \\ T(1) + \sum_{k=1}^{(n-1)/2} \frac{1}{\log(2k+1)} & \text{if } n \text{ is odd} \end{cases}$$

سوال۲: ب) رابطه بازگشتی $T(n) = T(n-2) + \frac{1}{logn}$ را حل کنید؟

$$T(n) = \begin{cases} T(0) + \sum_{k=1}^{n/2} \frac{1}{\log(2k)} & \text{if } n \text{ is even} \\ T(1) + \sum_{k=1}^{(n-1)/2} \frac{1}{\log(2k+1)} & \text{if } n \text{ is odd} \end{cases}$$

و میدانیم (؟)

$$\sum_{k=1}^{(n-1)/2} \frac{1}{\log(2k+1)} \leq \frac{1}{\log(3)} + \int_1^{(n-1)/2} \frac{\mathrm{d}x}{\log(2x+1)}$$
$$= \frac{1}{\log(3)} + \frac{1}{2}\int_3^8 \frac{\mathrm{d}x}{\log(x)} + \frac{1}{2}\int_8^n \frac{\mathrm{d}x}{\log(x)}$$

$$\sum_{k=1}^{n/2} \frac{1}{\log(2k)} \leq \frac{1}{\log(2)} + \int_1^{n/2} \frac{\mathrm{d}x}{\log(2x)}$$
$$= \frac{1}{\log(2)} + \frac{1}{2}\int_2^8 \frac{\mathrm{d}x}{\log(x)} + \frac{1}{2}\int_8^n \frac{\mathrm{d}x}{\log(x)}$$

## سوال۲: ب) رابطه بازگشتی $T(n) = T(n-2) + \frac{1}{logn}$ را حل کنید؟

$$T(n) = \begin{cases} T(0) + \sum_{k=1}^{n/2} \frac{1}{\log(2k)} & \text{if } n \text{ is even} \\ T(1) + \sum_{k=1}^{(n-1)/2} \frac{1}{\log(2k+1)} & \text{if } n \text{ is odd} \end{cases}$$

و میدانیم (؟)

$$\sum_{k=1}^{(n-1)/2} \frac{1}{\log(2k+1)} \le \frac{1}{\log(3)} + \int_1^{(n-1)/2} \frac{\mathrm{d}x}{\log(2x+1)}$$
$$= \frac{1}{\log(3)} + \frac{1}{2}\int_3^8 \frac{\mathrm{d}x}{\log(x)} + \frac{1}{2}\int_8^n \frac{\mathrm{d}x}{\log(x)}$$

$$\sum_{k=1}^{n/2} \frac{1}{\log(2k)} \le \frac{1}{\log(2)} + \int_1^{n/2} \frac{\mathrm{d}x}{\log(2x)}$$
$$= \frac{1}{\log(2)} + \frac{1}{2}\int_2^8 \frac{\mathrm{d}x}{\log(x)} + \frac{1}{2}\int_8^n \frac{\mathrm{d}x}{\log(x)}$$

سوال۲: ب) رابطه بازگشتی $T(n) = T(n-2) + \frac{1}{logn}$ را حل کنید؟

$$\sum_{k=1}^{(n-1)/2} \frac{1}{\log(2k+1)} \leq \frac{1}{\log(3)} + \int_1^{(n-1)/2} \frac{\mathrm{d}x}{\log(2x+1)}$$
$$= \frac{1}{\log(3)} + \frac{1}{2}\int_3^8 \frac{\mathrm{d}x}{\log(x)} + \frac{1}{2}\int_8^n \frac{\mathrm{d}x}{\log(x)}$$

$$\sum_{k=1}^{n/2} \frac{1}{\log(2k)} \leq \frac{1}{\log(2)} + \int_1^{n/2} \frac{\mathrm{d}x}{\log(2x)}$$
$$= \frac{1}{\log(2)} + \frac{1}{2}\int_2^8 \frac{\mathrm{d}x}{\log(x)} + \frac{1}{2}\int_8^n \frac{\mathrm{d}x}{\log(x)}$$

و برای n>=8 داریم که log(x)<=2(log(x)-1) پس:

$$\frac{1}{2}\int_8^n \frac{\mathrm{d}x}{\log(x)} \leq \int_8^n \frac{(\log(x)-1)\mathrm{d}x}{\log(x)^2}$$
$$= \frac{n}{\log(n)} - \frac{8}{\log(8)}$$

## سوال۲: ب) رابطه بازگشتی $T(n) = T(n-2) + \frac{1}{logn}$ را حل کنید؟

$$\frac{1}{2}\int_8^n \frac{\mathrm{d}x}{\log(x)} \leq \int_8^n \frac{(\log(x)-1)\mathrm{d}x}{\log(x)^2}$$
$$= \frac{n}{\log(n)} - \frac{8}{\log(8)}$$

پس در کل داریم:

$$T(n) \leq C + \frac{n}{\log(n)} = O\left(\frac{n}{\log(n)}\right)$$

سوال۲: ب) رابطه بازگشتی $T(n) = T(n-2) + \frac{1}{logn}$ را حل کنید؟

مسئله ۴-۳ فصل ۴ام کتاب CLRS

جواب: Time Complexity of $T(n)=T(n-2)+\frac{1}{\log(n)}$ - Mathematics Stack Exchange

سوالات مشابه:

g) T(n)=T(n-1)+1/n

h) T(n)=T(n-1)+logn

## سوال مشابه: رابطه بازگشتی $T(n) = T(n-1) + \frac{1}{n}$ را حل کنید؟

g. Recall that $\chi_A$ denotes the indicator function of $A$, then, we see that the sum is

$$T(0) + \sum_{j=1}^{n} \frac{1}{j} = T(0) + \int_1^{n+1} \sum_{j=1}^{n+1} \frac{\chi_{(j,j+1)}(x)}{j} dx$$

However, since $\frac{1}{x}$ is monatonically decreasing, we have that for every $i \in \mathbb{Z}^+$,

$$\sup_{x \in (i,i+1)} \sum_{j=1}^{n+1} \frac{\chi_{(j,j+1)}(x)}{j} - \frac{1}{x} = \frac{1}{i} - \frac{1}{i+1} = \frac{1}{i(i+1)}$$

So, our expression for $T(n)$ becomes

$$T(N) = T(0) + \int_1^{n+1} \left( \frac{1}{x} + O(\frac{1}{\lfloor x \rfloor (\lfloor x \rfloor + 1)} \right) dx$$

We deal with the error term by first chopping out the constant amount between 1 and 2 and then bound the error term by $O(\frac{1}{x(x-1)})$ which has an anti-derivative (by method of partial fractions) that is $O(\frac{1}{n})$. so,

$$T(N) = \int_1^{n+1} \frac{dx}{x} + O(\frac{1}{n} = \lg(n) + T(0) + \frac{1}{2} + O(\frac{1}{n})$$

This gets us our final answer of $T(n) \in \Theta(\lg(n))$

## سوال مشابه: رابطه بازگشت $T(n) = T(n-1) + logn$ را حل کنید؟

h. we see that we explicity have

$$T(n) = T(0) + \sum_{j=1}^{n} \lg(j) = T(0) + \int_{1}^{n+1} \sum_{j=1}^{n+1} \chi_{(j,j+1)}(x) \lg(j) dx$$

Similarly to above, we will relate this sum to the integral of $\lg(x)$.

$$\sup_{x \in (i,i+1)} \left| \sum_{j=1}^{n+1} \chi_{(j,j+1)}(x) \lg(j) - \lg(x) \right| = \lg(j+1) - \lg(j) = \lg\left(\frac{j+1}{j}\right)$$

So,

$$T(n) \leq \int_{i}^{n} \lg(x+2) + \lg(x) - \lg(x+1) dx = (1 + O(\frac{1}{\lg(n)}))\Theta(n\lg(n))$$

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

الف) اگر $m$ برابر $n$ باشد، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(logn)$ پیدا کند.

ب) برای هر $m$ و $n$ دلخواه، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(\log(\min\{m,n\}))$ پیدا کند.

Median of two sorted arrays of same size - GeeksforGeeks

Median of two sorted arrays of different sizes - GeeksforGeeks

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**الف) اگر $m$ برابر $n$ باشد، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(logn)$ پیدا کند.**

**روش ۱** – همزمان با Merge کردن (ادغام کردن) بشماریم تا به n امی برسیم! -> $O(n)$

**روش ۲** – مقایسه میانه‌های دوتا لیست -> $O(logn)$

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**الف) اگر $m$ برابر $n$ باشد، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(logn)$ پیدا کند.**

~~**روش ۱** – همزمان با Merge کردن (ادغام کردن) بشماریم تا به n امی برسیم! -> $O(n)$~~

**روش ۲ – مقایسه میانه‌های دوتا لیست -> $O(logn)$**

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**الف) اگر $m$ برابر $n$ باشد، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(logn)$ پیدا کند.**

روش ۲ – **مقایسه میانه‌های دوتا لیست -> O(logn)**

مثال:

A[] = {1, 12, 15, 26, 38}, B[] = {2, 13, 17, 30, 45}, $m_1 = 15 < m_2 = 17$

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**الف) اگر $m$ برابر $n$ باشد، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(logn)$ پیدا کند.**

**روش ۲ – مقایسه میانه‌های دوتا لیست -> O(logn)**

مثال:

```
A[] = {1, 12, 15, 26, 38},   B[] = {2, 13, 17, 30, 45},
```

$m_1 = 15 < m_2 = 17$

```
[15, 26, 38],          [2, 13, 17]
```

$m_1 = 26 > m_2 = 13$

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**الف) اگر $m$ برابر $n$ باشد، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(logn)$ پیدا کند.**

**روش ۲ – مقایسه میانه‌های دوتا لیست -> O(logn)**

مثال:

```
A[] = {1, 12, 15, 26, 38},   B[] = {2, 13, 17, 30, 45},   m1 = 15 < m2 = 17

 [15, 26, 38],             [2, 13, 17],                 m1 = 26 > m2 = 13

[15, 26] ,              [13, 17],                 size = 2

median = (max(ar1[0], ar2[0]) + min(ar1[1], ar2[1]))/2
                     = (max(15, 13) + min(26, 17))/2
                     = (15 + 17)/2
                     = 16
```

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**الف) اگر $m$ برابر $n$ باشد، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(logn)$ پیدا کند.**

**روش ۲ – مقایسه میانه‌های دوتا لیست -> $O(logn)$**

۱- میانه‌های دوتا لیست را بدست آوریم و آن‌ها را $m_1$ و $m_2$ بنامیم.

۲- اگر $m_1$ و $m_2$ باهم برابر باشد، که جواب مسئله است.

۳- اگر $m_2 > m_1$، آنگاه میانه در یکی از دو حالت زیر است:

الف) آرایه A، لیست جدید از اولین خانه تا $m_1$

ب) آرایه B، لیست جدید از خانه $m_2$ تا آخرین خانه

۴- اگر $m_1 > m_2$، آنگاه میانه در یکی از دو حالت زیر است:

الف) آرایه B، لیست جدید از اولین خانه تا $m_2$

ب) آرایه A، لیست جدید از خانه $m_1$ تا آخرین خانه

۵- مراحل را از اول تکرار میکنیم برای لیست های جالت ۳ یا ۴ تا زمانی که به لیستی با طول ۲ برسیم.

۶- طول ۲ چه کنیم؟

میانه: $\frac{(\max(ar1[0],ar2[0])+\min(ar1[1],ar2[1]))}{2}$

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**ب) برای هر m و n دلخواه، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(\log(\min\{m,n\}))$ پیدا کند.**

**روش – مقایسه میانه‌های دوتا لیست**

مثال:

```
a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10},  b[] = { 11, 12, 13, 14, 15, 16, 17, 18, 19 },

    smaller array[] = 1 2 3 4 5 6 7 8 9 10, mid = 5
    larger array[] = 11 12 13 14 15 16 17 18 19 , mid = 15
    5<15
```

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**ب) برای هر m و n دلخواه، یک الگوریتم پیشنهاد دهید که میانه را در زمان $\boldsymbol{O}(\log(\min\{\boldsymbol{m},\boldsymbol{n}\}))$ پیدا کند.**

**روش – مقایسه میانه‌های دوتا لیست**

مثال:

```
a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10},   b[] = { 11, 12, 13, 14, 15, 16, 17, 18, 19 },
```

smaller array[] = 1 2 3 4 5 6 7 8 9 10, mid=5 < larger array[] = 11 12 13 14 15 16 17 18 19 , mid = 15

smaller array[] = 11 12 13 14 15, mid = 13 > larger array[] = 5 6 7 8 9 10, mid = 7

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**ب) برای هر m و n دلخواه، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(\log(\min\{m,n\}))$ پیدا کند.**

**روش – مقایسه میانه‌های دوتا لیست**

مثال:

```
a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10},  b[] = { 11, 12, 13, 14, 15, 16, 17, 18, 19 },
```

smaller array[] = ~~1 2 3 4~~ 5 6 7 8 9 10, mid=5 < larger array[] = 11 12 13 14 15 ~~16 17 18 19~~ , mid = 15

smaller array[] = 11 12 13 ~~14 15~~, mid = 13 > larger array[] = ~~5 6~~ 7 8 9 10, mid = 7

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**ب) برای هر m و n دلخواه، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(\log(\min\{m,n\}))$ پیدا کند.**

**روش – مقایسه میانه‌های دوتا لیست**

مثال:

```
a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10},  b[] = { 11, 12, 13, 14, 15, 16, 17, 18, 19 },
```

smaller array[] = 1 2 3 4 5 6 7 8 9 10, mid = 5 < larger array[] = 11 12 13 14 15 16 17 18 19 , mid = 15

smaller array[] = 11 12 13 14 15, mid = 13 > larger array[] = 5 6 7 8 9 10, mid = 7

smaller array[] = 11 12 13 , mid = 12 > larger array[] = 7 8 9 10 , mid = 8

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**ب) برای هر m و n دلخواه، یک الگوریتم پیشنهاد دهید که میانه را در زمان $O(\log(\min\{m,n\}))$ پیدا کند.**

**روش – مقایسه میانه‌های دوتا لیست**

مثال:

```
a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10},  b[] = { 11, 12, 13, 14, 15, 16, 17, 18, 19
},
```

smaller array[] = 1 2 3 4 5 6 7 8 9 10, mid = 5 < larger array[] = 11 12 13 14 15 16 17 18 19 , mid = 15

smaller array[] = 11 12 13 14 15, mid = 13 > larger array[] = 5 6 7 8 9 10, mid = 7

smaller array[] = 11 12 13 , mid = 12 > larger array[] = 7 8 9 10 , mid = 8

smaller array[] = 11 12 ? larger array[] = 8 9 10

*Size of the smaller array is 2 and the size of the larger array is odd so, the median will be the median of* $\boldsymbol{max}(\mathbf{11},\mathbf{8}),\mathbf{9},\boldsymbol{min}(\mathbf{10},\mathbf{12})$ *that is* 9, 10, 11, *so the median is* ***10***.

**سوال ۳:** فرض کنید آرایه مرتب شده A شامل $n$ عنصر و آرایه مرتب شده B شامل $m$ عنصر باشند و همچنین فرض کنید که تمام عناصر هردو آرایه متمایز از هم هستند. می‌خواهیم تمام عناصر در A و B را پیدا کنیم.

**ب) برای هر m و n دلخواه، یک الگوریتم پیشنهاد دهید که میانه را در زمان $\boldsymbol{O}(\log(\min\{\boldsymbol{m},\boldsymbol{n}\}))$ پیدا کند.**

**روش – مقایسه میانه‌های دوتا لیست**

۱- میانه‌های دوتا لیست را بدست آوریم و آن‌ها را $m_1$ و $m_2$ بنامیم. (اگر سایز آرایه ای ۰ بود میانه آرایه دومی را خروجی بدهید).

۲- اگر اندازه آرایه کوچکتر، ۱ بود،

الف) همچنین سایز ارایه دیگر نیز ۱ بود، میانه میشود، میانه عضوها.

ب) اگر اندازه آرایه بزرگتر فرد بود، میانه ۴ تا عضو زیر را حساب کنید:

**عضو آرایه ۱ عضوی، عضو $\frac{m}{2}$، $\frac{m}{2}-1$** و **$\frac{m}{2}+1$** از آرایه بزرگتر

پ) اگر اندازه آرایه بزرگتر زوج بود، میانه ۳ تا عضور زیر را حساب کنید:

**عضو آرایه ۱ عضوی، عضو $\frac{m}{2}$** و **$\frac{m}{2}-1$** از آرایه بزرگتر

۳- اگر اندازه آرایه کوچکتر، ۲ بود،

الف) همچنین سایز ارایه دیگر نیز ۲ بود، میانه میشود، میانه ۴ عضو.

ب) اگر اندازه آرایه بزرگتر فرد بود، میانه ۳ عضو زیر را حساب میکنیم.

**- عضو وسط آرایه بزرگتر، ماکس عضو اول آرایه کوچکتر و $\frac{m}{2}-1$ آرایه بزرگتر ،مین عضو آخر آرایه کوچکتر و $\frac{m}{2}+1$ آرایه بزرگتر**

پ) اگر اندازه آرایه بزرگتر زوج بود؛ میانه ۳ عضو زیر را حساب میکنیم.

**- میانه دو عضور وسط آرایه بزرگتر، ماکس عضو اول آرایه کوچکتر و m/2 – 2 ، مین عضو آخر آرایه کوچکتر و $\frac{m}{2}+1$ آرایه بزرگتر**

1. **If the size of smaller array is 0. Return the median of a larger array.**
2. ***if the size of smaller array is 1.***
    1. The size of the larger array is also 1. Return the median of two elements.
    2. If the size of the larger array is odd. Then after adding the element from 2nd array, it will be even so the median will be an average of two mid elements. So the element from the smaller array will affect the median if and only if it lies between (m/2 – 1)th and (m/2 + 1)th element of the larger array. So, find the median in between the four elements, the element of the smaller array and (m/2)th, (m/2 – 1)th and (m/2 + 1)th element of a larger array
    3. Similarly, if the size is even, then check for the median of three elements, the element of the smaller array and (m/2)th, (m/2 – 1)th element of a larger array

3. ***If the size of smaller array is 2***
    1. If the larger array also has two elements, find the median of four elements.
    2. *If the larger array has an odd number of elements, then the median will be one of the following 3 elements*
        1. Middle element of larger array
        2. Max of the second element of smaller array and element just before the middle, i.e M/2-1th element in a bigger array
        3. Min of the first element of smaller array and element
           just after the middle in the bigger array, i.e M/2 + 1th element in the bigger array
    3. *If the larger array has even number of elements, then the median will be one of the following 4 elements*
        1. The middle two elements of the larger array
        2. Max of the first element of smaller array and element just before the first middle element in the bigger array, i.e M/2 – 2nd element
        3. Min of the second element of smaller array and element just after the second middle in the bigger array, M/2 + 1th element