

طراحی و تحلیل الگوریتم

استاد:

دکتر زاهد رحمتی

تدریس‌یاران:

داریوش کاظمی

اشکان ودادی

ترم دوم ۱۴۰۰



جلسه سوم

درخت بازگشتی – تقسیم و حل
(بخش ۳-۴ کتاب CLRS)

سوال ۱: پیچیدگی زمانی را با کمک قضیه اساسی بدست آورید.

1. $T(n) = 2^n T(n/2) + n^n$ $\rightarrow a$ is not constant!
2. $T(n) = \sqrt{2} T\left(\frac{n}{2}\right) + \log n$ $\rightarrow T(n) = \Theta(\sqrt{n})$ (Case 1)
3. $T(n) = 0.5 T\left(\frac{n}{2}\right) + \frac{1}{n}$ $\rightarrow a < 1$!
4. $T(n) = 64 T(n/8) - n^2 \log n$ $\rightarrow f(x)$ is not positive!
5. $T(n) = 3 T\left(\frac{n}{3}\right) + \frac{n}{2}$ $\rightarrow T(n) = \Theta(n \log n)$ (Case 2)
6. $T(n) = T(n/2) + n(2 - \cos n)$ \rightarrow We are in Case 3, but the regularity condition is violated. (Consider $n = 2\pi k$, where k is odd and arbitrarily large. For any such choice of n , you can show that $c \geq 3/2$, thereby violating the regularity condition.)
7. $T(n) = 2 T\left(\frac{n}{4}\right) + n^{0.51}$ $\rightarrow T(n) = \Theta(n^{0.51})$ (Case 3)


$$T(n) = 2^n T(n/2) + n^n \quad \rightarrow a \text{ is not constant!}$$

$$T(n) = T(n/2) + n(2 - \cos n) \quad \rightarrow \text{We are in Case 3, but the regularity condition is violated. (Consider } n = 2\pi k, \text{ where } k \text{ is odd and arbitrarily large. For any such choice of } n, \text{ you can show that } c \geq 3/2, \text{ thereby violating the regularity condition.)}$$

سوال: مقدار رابطه بازگشتی زیر را با استفاده از درخت بازگشت بدست آورید؟

$$T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + n$$

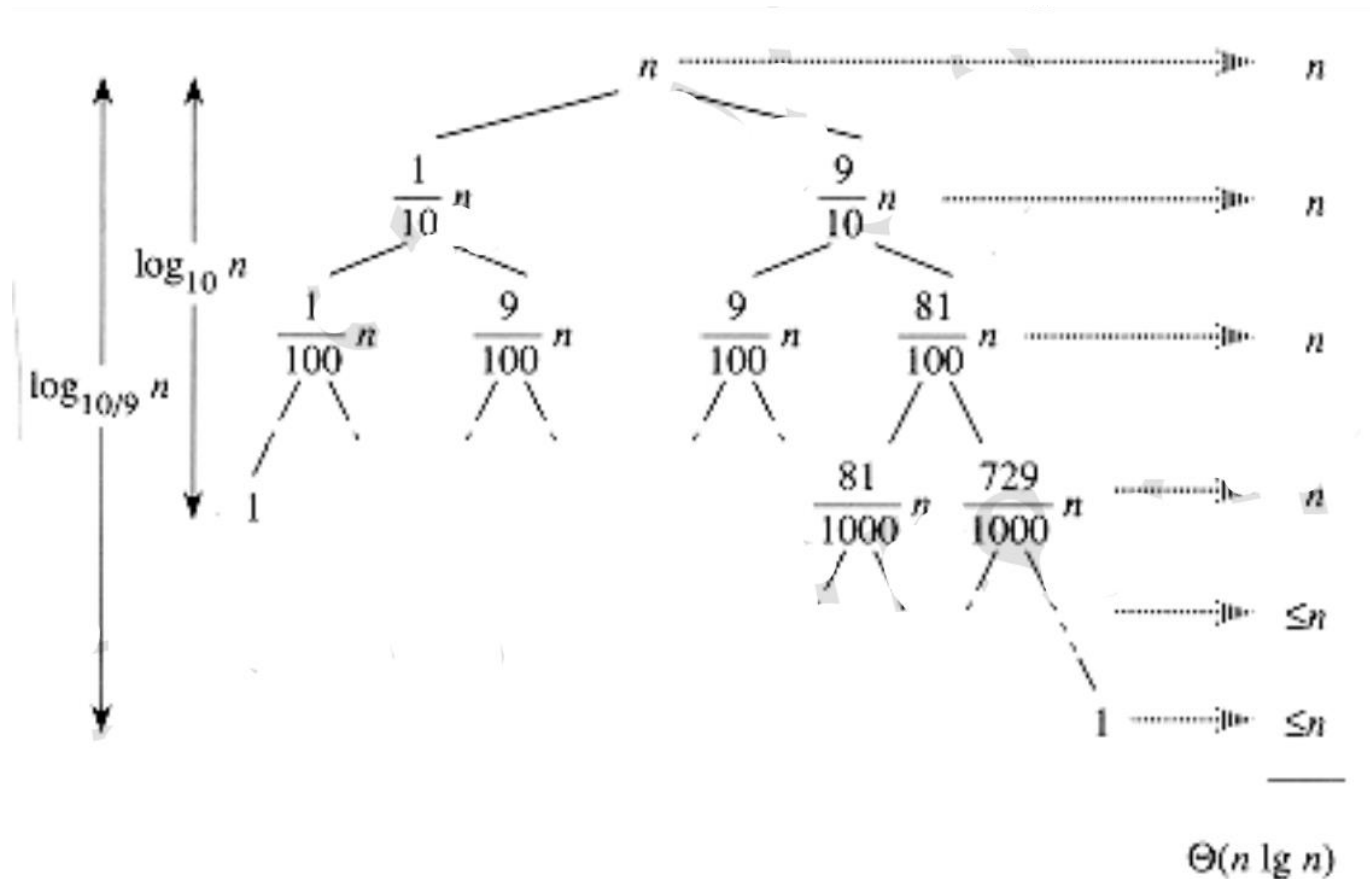
درخت بازگشت:

در این روش نحوه جای گذاری یک عبارت بازگشتی و نیز مقدار ثابتی را که در هر سطح از آن عبارت به دست می آید نشان می دهد. با جمع کردن مقادیر ثابت تمام سطح، جواب بدست می آید.

درخت های بازگشت زمانی که رابطه بازگشتی، زمان اجرای یک الگوریتم تقسیم و حل را توصیف می کند بسیار مفید است.

سوال: مقدار رابطه بازگشتی زیر را با استفاده از درخت بازگشت بدست آورید؟

$$T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + n$$



تقسیم و حل

1. تقسیم: نمونه‌ای از مسئله را به مسائل کوچکتر تبدیل می‌کنیم.
2. حل: مسائل کوچک‌تر حل می‌کنیم.
3. ترکیب: حل نمونه‌های کوچکتر برای حل مسئله اصلی (در صورت نیاز)

تقسیم و حل

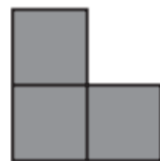
1. تقسیم: نمونه‌ای از مسئله را به مسائل کوچکتر تبدیل می‌کنیم.
2. حل: مسائل کوچک‌تر حل می‌کنیم.
3. ترکیب: حل نمونه‌های کوچکتر برای حل مسئله اصلی (در صورت نیاز)

معمولا به صورت روال بازگشتی!

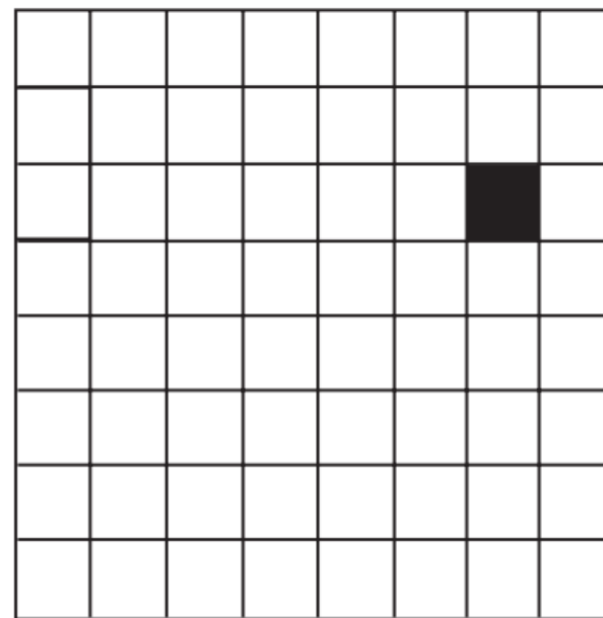
تقسيم و حل

1. MERGE SORT
2. BINARY SEARCH
3. POWERING A NUMBER
4. INVERSION COUNTING
5. MATRIX MULTIPLICATION + STRASSEN'S METHOD
6. CLOSEST PAIR
7. LONG INTEGER MULTIPLICATION
8. SELECTION + SIEVE TECHNIQUE
9. CHOOSING THE PIVOT
10. QUICK SORT
11. TROMINO PUZZLE (TILING PROBLEM)
12. MEDIAN OF TWO SORTED ARRAYS

سوال: Tromino Puzzle - چه طور می‌شود یک صفحه شطرنج $2^n \times 2^n$ با یک خانه پر شده را به طور کامل با تعدادی Right Tromino پوشش داد. توجه کنید آزادانه می‌توان Tromino را جهت دهی کرد اما نباید هیچ Tromino ای روی خانه پر شده قرار گیرد و سایر خانه‌ها را پوشش دهند. همچنین هیچ کدام از Tromino ها نباید بر هم منطبق شوند. یک الگوریتم تقسیم و حل برای این مسئله طراحی کنید و پیچیدگی زمانی آن را بررسی کنید.



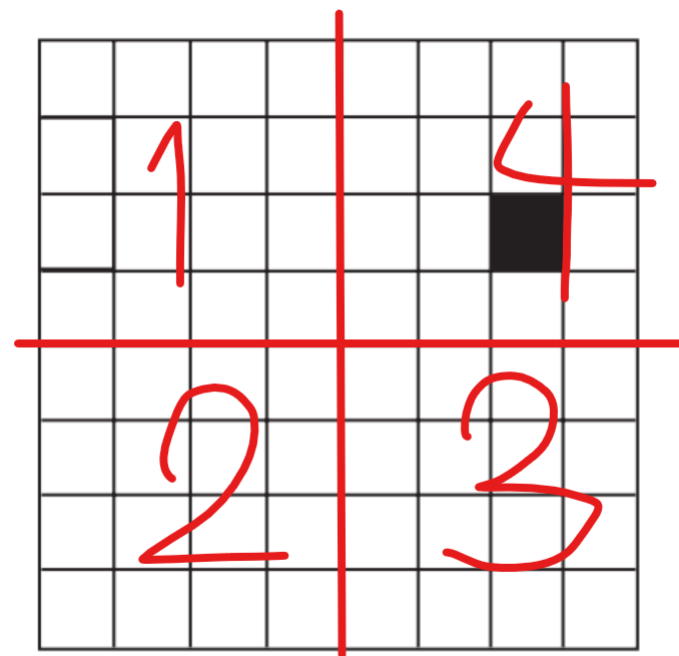
Right Tromino



سوال: Tromino Puzzle - چه طور می‌شود یک صفحه شطرنج $2^n \times 2^n$ با یک خانه پر شده را به طور کامل با تعدادی Right Tromino پوشش داد. توجه کنید آزادانه می‌توان Tromino را جهت دهی کرد اما نباید هیچ Tromino ای روی خانه پر شده قرار گیرد و سایر خانه‌ها را پوشش دهند. همچنین هیچ کدام از Tromino ها نباید بر هم منطبق شوند. یک الگوریتم تقسیم و حل برای حل این مسئله طراحی کنید و پیچیدگی زمانی آن را بررسی کنید.

جواب:

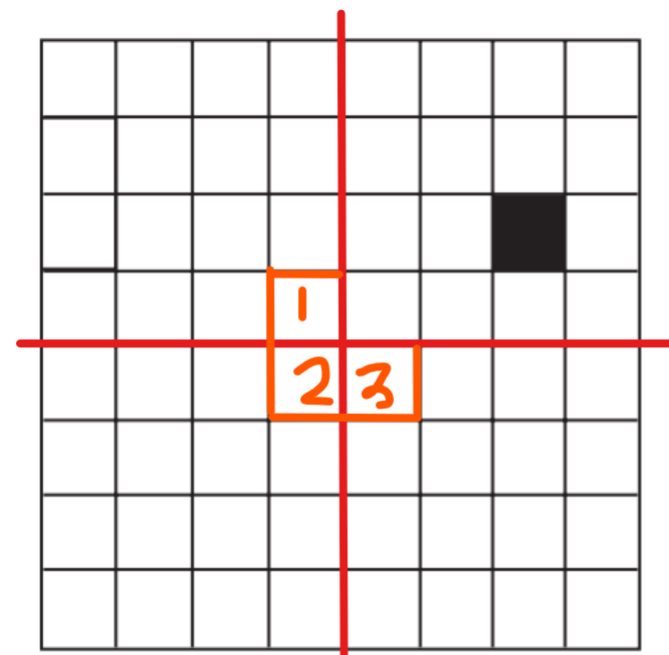
برای حل این سوال فرض کنید می‌خواهیم مسئله $T(2^n)$ را حل کنیم.
و مسئله را به مجموع ۴ شکل هم اندازه تر (تقسیم به مسئله کوچک تر) تبدیل می‌کنیم.



سوال: Tromino Puzzle - چه طور می‌شود یک صفحه شطرنج $2^n \times 2^n$ با یک خانه پر شده را به طور کامل با تعدادی Right Tromino پوشش داد. توجه کنید آزادانه می‌توان Tromino را جهت دهی کرد اما نباید هیچ Tromino ای روی خانه پر شده قرار گیرد و سایر خانه‌ها را پوشش دهند. همچنین هیچ کدام از Tromino ها نباید بر هم منطبق شوند. یک الگوریتم تقسیم و حل برای این مسئله طراحی کنید و پیچیدگی زمانی آن را بررسی کنید.

جواب:

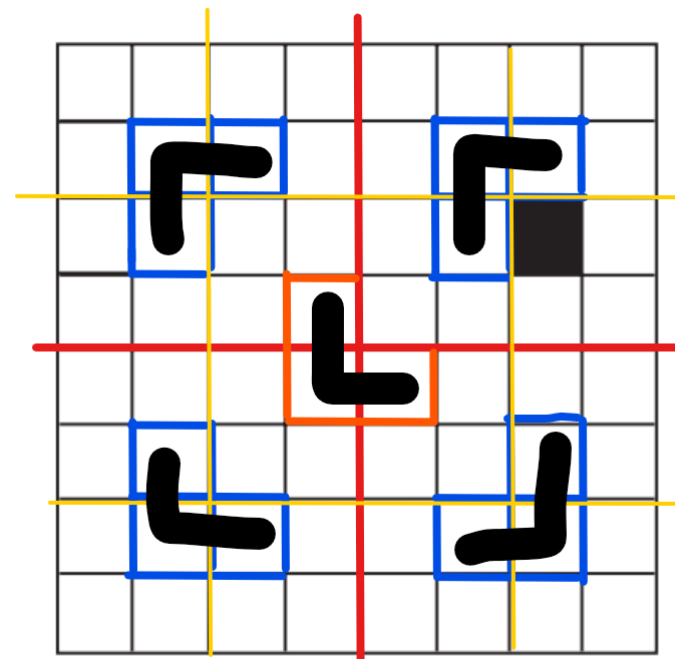
برای حل این سوال فرض کنید می‌خواهیم مسئله $T(2^n)$ را حل کنیم.
و مسئله را به مجموع ۴ شکل هم اندازه تر (تقسیم به مسئله کوچک تر) تبدیل میکنیم.
در هر مرحله شکل Tromino را وسط ۴ شکل حاصل قرار میدهیم. به طوری که اگر یک مربع ۲ در ۲ فرض کنیم قسمت خالی آن به سمت شکلی که خانه پر شده دارد قرار گیرد.
و این عمل را تا زمانی که به $n = 1$ می‌رسیم.



سوال: Tromino Puzzle - چه طور می‌شود یک صفحه شطرنج $2^n \times 2^n$ با یک خانه پر شده را به طور کامل با تعدادی Right Tromino پوشش داد. توجه کنید آزادانه می‌توان Tromino را جهت دهی کرد اما نباید هیچ Tromino ای روی خانه پر شده قرار گیرد و سایر خانه‌ها را پوشش دهند. همچنین هیچ کدام از Tromino ها نباید بر هم منطبق شوند. یک الگوریتم تقسیم و حل برای این مسئله طراحی کنید و پیچیدگی زمانی آن را بررسی کنید.

جواب:

برای حل این سوال فرض کنید می‌خواهیم مسئله $T(2^n)$ را حل کنیم.
و مسئله را به مجموع ۴ شکل هم اندازه تر (تقسیم به مسئله کوچک تر) تبدیل میکنیم.
در هر مرحله شکل Tromino را وسط ۴ شکل حاصل قرار میدهیم. به طوری که اگر یک مربع ۲ در ۲ فرض کنیم قسمت خالی آن به سمت شکلی که خانه پر شده دارد قرار گیرد.
و این عمل را تا زمانی که به $n = 1$ می‌رسیم.
در این مرحله تمام خانه‌های ۲ در ۲ خالی مانده یک خانه پر شده دارد و با $O(1)$ شکل Tromino را در آن قرار میدهیم.



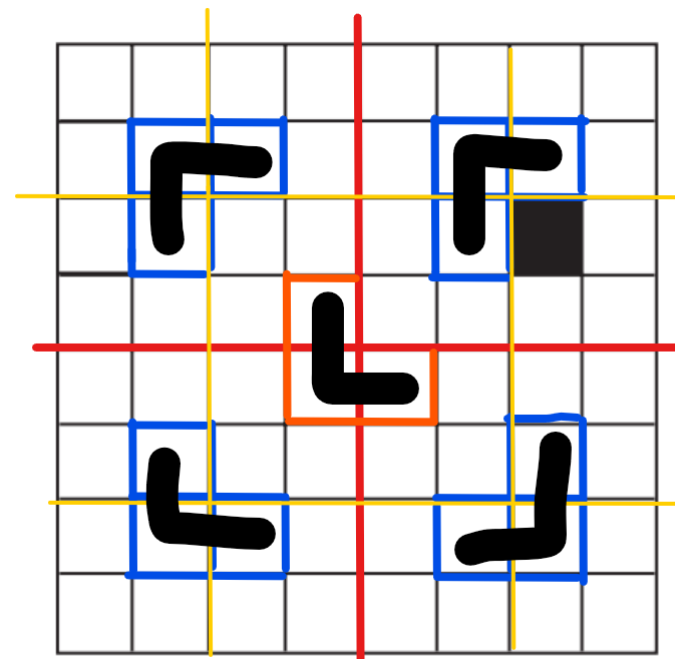
سوال: Tromino Puzzle - چه طور می‌شود یک صفحه شطرنج $2^n \times 2^n$ با یک خانه پر شده را به طور کامل با تعدادی Right Tromino پوشش داد. توجه کنید آزادانه می‌توان Tromino را جهت دهی کرد اما نباید هیچ Tromino ای روی خانه پر شده قرار گیرد و سایر خانه‌ها را پوشش دهند. همچنین هیچ کدام از Tromino ها نباید بر هم منطبق شوند. یک الگوریتم تقسیم و حل برای این مسئله طراحی کنید و پیچیدگی زمانی آن را بررسی کنید.

جواب:

برای حل این سوال فرض کنید می‌خواهیم مسئله $T(2^n)$ را حل کنیم.
و مسئله را به مجموع ۴ شکل هم اندازه تر (تقسیم به مسئله کوچک تر) تبدیل میکنیم.
در هر مرحله شکل Tromino را وسط ۴ شکل حاصل قرار میدهیم. به طوری که اگر یک مربع ۲ در ۲ فرض کنیم قسمت خالی آن به سمت شکلی که خانه پر شده دارد قرار گیرد.
و این عمل را تا زمانی که به $n = 1$ می‌رسیم.
در این مرحله تمام خانه‌های ۲ در ۲ خالی مانده یک خانه پر شده دارد و با $O(1)$ شکل Tromino را در آن قرار میدهیم.

درکل به رابطه بازگشتی زیر میرسیم.

$$T(2^n) = 4T(2^{n-1}) + O(1) \rightarrow O(n^2)$$



سوال: 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 هر دو 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 $O(\log n)$ 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: 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 $O(\log n)$ پیدا کند.

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

روش ۲ – مقایسه میانه‌های دوتا لیست $O(\log n)$

سوال: 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 هر دو 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 $O(\log n)$ پیدا کند.

~~روش ۱ - همزمان با Merge کردن (ادغام کردن) بشماریم تا به n امی برسیم! - $\Theta(n)$~~

روش ۲ - مقایسه میانه‌های دو تا لیست - $O(\log n)$

سوال: 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 هر دو 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 $O(\log n)$ پیدا کند.

روش ۲ – مقایسه میانه‌های دوتا لیست - $O(\log n)$
مثال:

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

سوال: 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 $O(\log n)$ پیدا کند.

روش ۲ – مقایسه میانه‌های دوتا لیست - $O(\log n)$
مثال:

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

$$[15, 26, 38], \quad [2, 13, 17] \quad m_1 = 26 > m_2 = 13$$

سوال: 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 هر دو 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 $O(\log n)$ پیدا کند.

روش ۲ - مقایسه میانه‌های دوتا لیست - $O(\log n)$
مثال:

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

$$[15, 26, 38], \quad [2, 13, 17], \quad m_1 = 26 > m_2 = 13$$

$$[15, 26], \quad [13, 17], \quad \text{size} = 2$$

$$\begin{aligned} \text{median} &= (\max(\text{ar1}[0], \text{ar2}[0]) + \min(\text{ar1}[1], \text{ar2}[1]))/2 \\ &= (\max(15, 13) + \min(26, 17))/2 \\ &= (15 + 17)/2 \\ &= 16 \end{aligned}$$

سوال: 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 هر دو 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 $O(\log n)$ پیدا کند.

روش ۲ - مقایسه میانه‌های دوتا لیست - $O(\log n)$

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 $m_1 > m_2$ ، 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 $m_2 > m_1$ ، 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 لیست‌های حالت 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: 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 هر دو 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: 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 هر دو 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: 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 هر دو 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: 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: 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 هر دو 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 $\max(11, 8), 9, \min(10, 12)$ that is 9, 10, 11, so the median is 10.

سوال: 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 هر دو 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

۱- میانه‌های دوتا لیست را بدست آوریم و آن‌ها را m_1 و m_2 بنامیم. (اگر سائز آرایه ای \cdot بود میانه آرایه دومی را خروجی بدهید).
۲- اگر اندازه آرایه کوچکتر، ۱ بود،

الف) همچنین سائز آرایه 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 عضو، 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 عضو زیر را حساب کنید:

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

۳- اگر اندازه آرایه کوچکتر، ۲ بود،

الف) همچنین سائز آرایه 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 عضو وسط آرایه بزرگتر، ماکس عضو اول آرایه کوچکتر و $\frac{m}{2} - 2$ ، 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 $M/2 - 1$ th 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 $M/2 + 1$ th 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 $M/2 - 2$ nd element
 3. Min of the second element of smaller array and element just after the second middle in the bigger array, $M/2 + 1$ th element

خسته نباشید!

داریوش کاظمی – اشکان ودادی