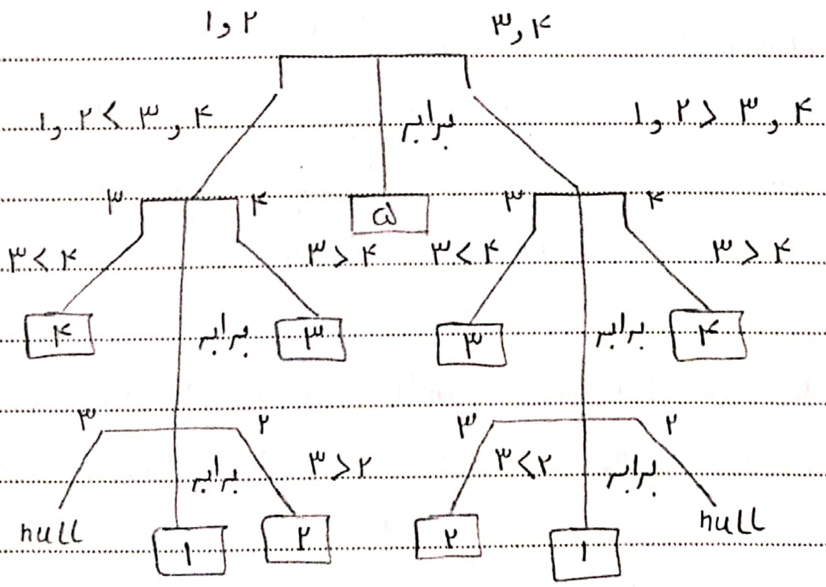
Subject:
Year: Month: Day: page: ( )

حوری دهش ۹۸۲۱۴۱۳

س ۱ ← ابتدا سکه ها را به ۳ گروه تقسیم بندی می کنیم سپس وزن گروه ها را دو به دو با هم مقایسه می کنیم و گروهی که وزن متفاوتی دارد شامل سکه متفاوت است.

برای درک بهتر این موضوع را با یک مثال بیان می کنیم ← n = ۴ در نظر می گیریم برای راحتی کار n را به توان ۳ نمی رسانیم ولی در آخر می توانیم برای هر n به توان ۳ این موضوع را تعمیم دهیم ← ۴ تا سکه داریم و آن ها را از ۱ تا ۴ نامگذاری می کنیم :



از یکی از ۲ مسیر بالا می توانیم بفهمیم سکه تقلبی کدام است

حالا اگر n فرد بود یکی از سکه ها تک می افتاد که در همان مرحله ابتدایی اگر ۱ و ۲ با ۳ و ۴ وزن برابر داشت آن گاه سکه شماره ۵ سکه تقلبی می شد (در شکل بالا نشان داده شده است)

Arman

page: ( )

Subject:
Year: Month: Day:

n log n

س ۲ ← ابتدا آرایه را مرتب می کنیم و این کار را با الگوریتم merge sort انجام می دهیم.

فرض می کنیم index از صفر شروع می شود و از آنجایی که آرایه فرد است پس index عنصر وسط آرایه $\frac{n-1}{2}+1$ می شود سپس عنصر میانی را با ۲ عنصر سمت راست و چپ خودش مقایسه می کنیم و ۳ حالت رخ می دهد:

1 ← اگر عنصر میانی با ۲ عنصر سمت راست و چپ خودش یکی نبود در این حالت عنصر میانی جواب مسئله می شود.

2 ← اگر عنصر میانی با عنصر سمت چپ خودش برابر بود ۲ حالت رخ می دهد:

2_1 ← اگر مقدار $\frac{n-1}{2}$ زوج شود آن موقع جواب مسئله در نیمه سمت چپ آرایه است و باید در آن جستجو را انجام دهیم.

2_2 ← اگر مقدار $\frac{n-1}{2}$ فرد شود آن موقع جواب مسئله در نیمه سمت راست آرایه است و باید در آن جستجو را انجام دهیم.

3 ← اگر عنصر میانی با عنصر سمت راست خودش برابر بود ۲ حالت رخ می دهد:

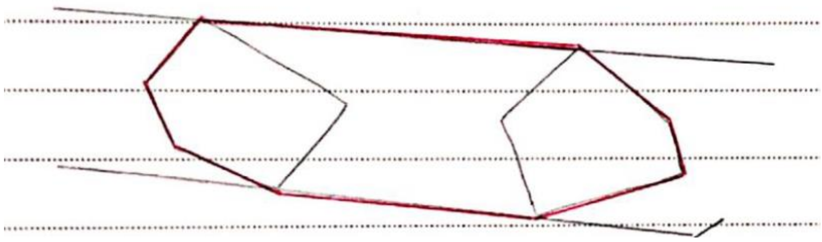
3_1 ← اگر مقدار $\frac{n-1}{2}$ زوج شود آن موقع جواب مسئله در نیمه سمت راست آرایه است و باید در آن جستجو را انجام دهیم.

3_2 ← اگر مقدار $\frac{n-1}{2}$ فرد شود آن موقع جواب مسئله در نیمه سمت چپ آرایه است و باید در آن جستجو را انجام دهیم.

Arman

سوال 3:

فرض میکنیم بدنه محدب نقاط سمت چپ و بدنه محدب نقاط سمت راست را می دانیم در نهایت می اییم این دو بدنه محدب را با هم merge میکنیم به این صورت که مماس بالایی و پایینی دو بدنه را پیدا میکنیم و یک بدنه محدب کلی می سازیم به صورت زیر:



برای اینکه بدنه محدب سمت راست و چپ را پیدا کنیم نقاطی که داریم را انقدر تقسیم میکنیم تا تعداد نقاط هر مجموعه کم شود (تقریبا 5 تا) در نهایت با استفاده از الگوریتم Jarvis بدنه محدب این چند نقطه را پیدا میکنیم و در اخر تکه هایی که داریم را با هم merge میکنیم تا بدنه محدب کلی را به ما دهد.

توضیح الگوریتم Jarvis : در ابتدا چپ ترین نقطه را انتخاب میکنیم و نقاط را در خلاف جهت ساعت بهم وصل میکنیم مثلا X_1 را چپ ترین نقطه در نظر میگیریم و الگوریتم زیر را انقدر تکرار میکنیم که به نقطه اول برسیم:

نقطه بعدی را X_2 در نظر میگیریم به طوری که سه تایی (X_1, X_2, X_3) برای هر نقطه X_3 خلاف جهت ساعت باشد.

در نقطه ها جلو می رویم و برای هر نقطه دیگر i که (X_1, i, X_2) خلاف جهت ساعت باشد X_2 را به i تغییر می دهیم.

X_2 به عنوان نقطه بعدی X_1 در نظر گرفته میشود و بعد $X_1 = X_2$ میشود.

سوال 4:

این مسئله را به صورت بازگشتی حل میکنیم:

برای این مسئله ابتدا میانه هر دو آرایه را محاسبه میکنیم و نیمی از هر آرایه را کنار می گذاریم. این رویکرد به این صورت است که اندازه آرایه ها را در نظر می گیرد. آرایه با اندازه کوچکتر اولین آرایه در پارامتر در نظر گرفته می شود.

- اگر اندازه ارایه کوچکتر 0 باشد میانه ارایه بزرگتر را برمی گردانیم.
- اگر اندازه ارایه کوچکتر 1 باشد:
  - اندازه ارایه بزرگتر نیز 1 است و میانه دو عنصر را برمی گردانیم
  - اگر اندازه ارایه بزرگتر فرد باشد سپس پس از اضافه کردن عنصر از دومین ارایه زوج خواهد شد بنابراین میانه به طور متوسط دو عنصر میانی خواهد شد:
    - بنابراین عنصر ارایه کوچکتر بر میانه تاثیر می گذارد اگر و فقط اگر بین عنصر (M/2-1) و عنصر (M/2+1) ارایه بزرگتر باشد.

- بنابراین میانه بین چهار عنصر: عنصر آرایه کوچکتر، عنصر (M/2)، عنصر (M/2-1) و عنصر (M/2+1) ارایه بزرگتر را می یابیم.
  - به طور مشابه اگر اندازه زوج باشد میانه سه عنصر: عنصر ارایه کوچکتر و عنصر (M/2) و عنصر (M/2-1) ارایه بزرگتر را بررسی میکنیم.
- اگر اندازه ارایه کوچکتر 2 باشد:
  - اگر ارایه بزرگتر دو عنصر داشته باشد میانه چهار عنصر را پیدا میکنیم.
  - اگر ارایه بزرگتر دارای تعداد فرد باشد میانه یکی از 3 عنصر زیر خواهد شد:
    - عنصر میانی ارایه بزرگتر
    - حداکثر عنصر دوم ارایه کوچکتر و عنصر قبل از میانه یعنی عنصر M/2-1 در ارایه بزرگتر
    - حداقل عنصر اول ارایه کوچکتر و عنصر بعد از میانه در ارایه بزرگتر یعنی M/2 + عنصر اول در ارایه بزرگتر
  - اگر ارایه بزرگتر دارای تعداد زوج باشد میانه یکی از 4 عنصر زیر خواهد شد:
    - دو عنصر میانی ارایه بزرگتر
    - حداکثر عنصر اول ارایه کوچکتر و عنصر قبل از اولین عنصر میانی در ارایه بزرگتر یعنی M/2 – عنصر دوم
    - حداقل عنصر دوم ارایه کوچکتر و عنصر بعد از دومین میانه در ارایه بزرگتر یعنی M/2 + عنصر اول

مثال:

Input :arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10},
brr[] = { 11, 12, 13, 14, 15, 16, 17, 18, 19 }

Recursive call 1:
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
Discard first half of the first array and second half of the second array
Recursive call 2:
smaller array[] = 11 12 13 14 15, mid = 13
larger array[] = 5 6 7 8 9 10, mid = 7
7 < 13
Discard first half of the second array and second half of the first array

Recursive call 3:
smaller array[] = 11 12 13 , mid = 12
larger array[] = 7 8 9 10 , mid = 8
8 < 12
Discard first half of the second array and second half of the first array

Recursive call 4:
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.

← س ۵

الف) $T(1)=1 \quad , \quad T(2)=2 \quad , \quad T(n) = 8\,T(n-1) - 15\,T(n-2)$

$T(n) = K^n$

$\longrightarrow K^n = 8K^{n-1} - 15K^{n-2} \xrightarrow{\div K^{n-2}} K^2 = 8K - 15 \longrightarrow$

$K^2 - 8K + 15 = 0 \longrightarrow K = 3 \quad , \quad K = 5$

$T(n) = c_1 3^n + c_2 5^n$

$$\left.\begin{array}{l} T(1)=1 \longrightarrow 1 = 3c_1 + 5c_2 \\ T(2)=2 \longrightarrow 2 = 9c_1 + 25c_2 \end{array}\right\} \Longrightarrow c_1 = \frac{1}{2} \quad , \quad c_2 = \frac{-1}{10}$$

$$T(n) \in \Theta\Big(\underbrace{\frac{3^n}{2} - \frac{5^n}{10}}_{c^n}\Big) \longrightarrow T(n) \in \Theta(c^n)$$

ب) $T(1) = 1 \ , \ T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + n^2$

$$cn^2 \longrightarrow cn^2$$

$$c\left(\frac{n}{4}\right)^2 \quad c\left(\frac{3n}{4}\right)^2 \longrightarrow \frac{10}{16}cn^2$$

$$c\left(\frac{n}{16}\right)^2 \quad c\left(\frac{3n}{16}\right)^2 \quad c\left(\frac{3n}{16}\right)^2 \quad c\left(\frac{9n}{16}\right)^2 \longrightarrow \left(\frac{10}{16}\right)^2 cn^2$$

$1 \ 1 \ 1 \ 1 \ \ldots \ 1 \ 1 \ 1 \ 1 \longrightarrow$ تعداد برگ‌ها = ?

ارتفاع $\longrightarrow \frac{n}{\left(\frac{4}{3}\right)^i} = 1 \longrightarrow \left(\frac{4}{3}\right)^i = n \longrightarrow \log\left(\frac{4}{3}\right)^i = \log n \longrightarrow$

$i = \log_{\frac{4}{3}} n$

تعداد برگ‌ها $\longrightarrow 2^i = 2^{\log_{\frac{4}{3}} n} = n^{\log_{\frac{4}{3}} 2} \longrightarrow \Theta\left(n^{\log_{\frac{4}{3}} 2}\right)$

$$T(n) = cn^2 + \frac{10}{16}cn^2 + \left(\frac{10}{16}\right)^2 cn^2 + \cdots + \left(\frac{10}{16}\right)^{\log_{\frac{4}{3}} n-1} cn^2 + \Theta\left(n^{\log_{\frac{4}{3}} 2}\right) =$$

$$\sum_{i=0}^{\log_{\frac{4}{3}} n-1} \left(\frac{10}{16}\right)^i cn^2 + \Theta\left(n^{\log_{\frac{4}{3}} 2}\right) <$$

$$\sum_{i=0}^{\infty} \left(\frac{10}{16}\right)^i cn^2 + \Theta\left(n^{\log_{\frac{4}{3}} 2}\right) = \frac{1}{1-\frac{10}{16}} cn^2 + \Theta\left(n^{\log_{\frac{4}{3}} 2}\right)$$

$$= \frac{16}{6} cn^2 + \Theta\left(n^{\log_{\frac{4}{3}} 2}\right) = O(n^2)$$

$\star \longrightarrow \sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$

Arman

ج) $T(۲)=۱ \;,\; T(n)=۴T(\lceil\sqrt{n}\rceil)+۱$

$$n=۲^m \longrightarrow T(۲^m)=۴T(۲^{\frac{m}{۲}})+۱ \xrightarrow{T(۲^m)=S(m)}$$

$$S(m)=۴S\left(\frac{m}{۲}\right)+۱ \xrightarrow[\text{method}]{\text{master}} \begin{array}{l} a=۴ \\ b=۲ \\ f(m)=۱ \end{array} \Rightarrow m^{\log_۲^۴}=m^۲$$

$$\longrightarrow f(m)=O(m^{\log_۲^۴-\varepsilon}) \longrightarrow S(m)=\Theta(m^{\log_۲^۴})=\Theta(m^۲)$$

$$\longrightarrow T(۲^m)=\Theta(m^۲) \xrightarrow[n=۲^m]{m=\log n} T(n)=\Theta(\log^n)^۲$$

د) $T(۱)=۱ \;,\; T(۲^n)=۷T(۲^{n-۱})+۲^n$

$$۲^n=m \;,\; ۲^{n-۱}=\frac{۲^n}{۲}$$

$$T(m)=۷T\left(\frac{m}{۲}\right)+m \xrightarrow[\text{method}]{\text{master}} \begin{array}{l} a=۷ \\ b=۲ \\ f(m)=m \end{array} \Rightarrow m^{\log_۲^۷}$$

$$\longrightarrow f(m)=O(m^{\log_۲^۷-\varepsilon}) \longrightarrow T(m)=\Theta(m^{\log_۲^۷}) \xrightarrow{m=۲^n}$$

$$T(۲^n)=\Theta\left((۲^n)^{\log_۲^۷}\right)=\Theta\left(۲^{\log_۲^{۷^n}}\right)$$

سوال 7:

در ابتدا مقادیر تابع g(i,j) را محاسبه می کنیم برای این کار می توانیم از روش prefix sum برویم به این صورت که یک ارایه جدیدی به نام prefix_sum می سازیم که در ان عناصر از ابتدای ارایه اصلی تا هر مکانی در ان ارایه جمع میشوند:

```
prefix_sum[0] = 0

for i = 1 to n:

prefix_sum[i] = prefix_sum[i-1] + a[i]
```

در نهایت برای محاسبه مقدار تابع g(i,j) با استفاده از ارایه prefix_sum می توانیم به صورت زیر عمل کنیم:

```
int g(int i, int j, int *prefix_sum)

{

int min = (i < j) ? i : j;

int max = (i > j) ? i : j;

int sum = prefix_sum[max] - prefix_sum[min];

return sum;

}
```

برای اینکه کمینه f(i,j) را حساب کنیم می توانیم از الگوریتم Closest Pair استفاده کنیم به این صورت که اول باید تمام جفت های ممکن i,j را برای مقادیر f(i,j) حساب کنیم بعد از ان با استفاده از الگوریتم Closest Pair جفتی از اندیس های i,j را پیدا می کنیم که مقدار کمینه f(i,j) را دارا باشند.

پس الگوریتم کلی برای محاسبه کمینه f(i,j) به صورت زیر است:

1- محاسبه آرایه prefix_sum با استفاده از روش prefix sum

2- محاسبه مقادیر تابع f(i,j) برای تمام جفت های ممکن i,j

3- پیدا کردن جفتی از اندیس های i,j که مقدار کمینه f(i,j) را دارند با استفاده از الگوریتم Closest Pair

در نهایت پیچیدگی الگوریتم ما از O(n log n) میشود.