مهرزاد ممیز

تکلیف چهارم طراحی الگوریتم _ استاد اسدپور 810100272

سوال ۱) طبق روش brute force ، ۲ تا nested Loop به طول n که یکی یکی عناصر را choose می کنند و با بررسی تعداد تکرار هر عنصر در کل Array آرایه توانایی داریم از $O(n^2)$ سوال را حل کنیم

در ابتدا کل Array آرایه را از $O(n \log n)$ order sort می کنیم و در این صورت با یک سری حرکت بر روی اعضای Array و مقایسه ی آن در ۲ عضو پشت سرهم [متوالی] توانایی داریم که سوال را حل کنیم

و بنابراین در حالتی که عدد بعدی با عدد فعلی ما مساوی شود و تفاوت بود، count را یکی افزایش می دهیم و در حالتی که تفاوت داشت، یک متغیر جدید count برای آن عدد جدید می نویسیم و مقدار عنصر جدید را در آن save می کنیم

طبق این روش، $O(n \log n)$ با استفاده از حافظه خیلی کمتری است

سوال ۲) در ابتدا algorithm را بررسی می کنیم و معلوم می کنیم در وضعیتی در آخر آن قطعاً یک عنصر می ماند و در چه وضعیتی یک عنصر نمی ماند. algorithm به این صورت است که در ابتدا برای آسوده تر شدن راحت تر شدن نحوه ی حل، Array را منظم می کنیم، و بعد از آن هر عنصر را با عنصر بعدی مقایسه می کنیم اگر تفاضل آنها ≥ ۱ کمتر مساوی ۱ بوده آن مورد را حذف می کنیم. در صورتی که اینگونه نبود، عنصر بعدی را انتخاب می کنیم. پیچیدگی algorithm چون حرکت روی آرایه Array از order n است، بنابراین به order منظم سازی بستگی دارد. در این حالت استفاده از merge sort، پیچیدگی Argorithm هم $O(n \log n)$ می شود. در حالتی که برای هر عنصر ثابت شود که حداقل ۲ عنصر با تفاضل کوچکتر مساوی ۱ با آن وجود دارد، توانایی داریم که در آخر، به یک عنصر در Array رسید، و اما اگر اینگونه نشد امکان پذیر نیست که بتوانیم به یک عنصر در Array آرایه برسیم

A = [۳, ۴, ۵, ۶, ۷, ۷, ۸, ۹]

[۹]

تقاضایی دارم فقط به یک
عنصر برسیم

B = [۳, ۴, ۵, ۷, ۷, ۸, ۹, ۹]

سوال ۳

$$7x + 5y - 3z = 14$$
$$3x - 5y + 2z = -8$$
$$5x + 3y - 7x = 0$$

$$\left[\begin{array}{ccc|c} 7 & 5 & -3 & 14 \\ 3 & -5 & 2 & -8 \\ 5 & 3 & -7 & 0 \end{array}\right]$$

$$\left[\begin{array}{ccc|c} 7 & 5 & -3 & 14 \\ 3 - \frac{3}{7}\cdot 7 & -5 - \frac{3}{7}\cdot 5 & 2 - \frac{3}{7}\cdot(-3) & -8 - \frac{3}{7}\cdot 14 \\ 5 & 3 & -7 & 0 \end{array}\right] = \left[\begin{array}{ccc|c} 7 & 5 & -3 & 14 \\ 0 & -\frac{50}{7} & \frac{23}{7} & -\frac{104}{7} \\ 5 & 3 & -7 & 0 \end{array}\right]$$

$$\left[\begin{array}{ccc|c} 7 & 5 & -3 & 14 \\ 0 & -\frac{50}{7} & \frac{23}{7} & -\frac{104}{7} \\ 5 - \frac{5}{7}\cdot 7 & 3 - \frac{5}{7}\cdot 5 & -7 - \frac{5}{7}\cdot(-3) & 0 - \frac{5}{7}(14) \end{array}\right] =$$

$$\left[\begin{array}{ccc|c} 7 & 5 & -3 & 14 \\ 0 & -\frac{50}{7} & \frac{23}{7} & -\frac{104}{7} \\ 0 & -\frac{4}{7} & -\frac{34}{7} & -\frac{80}{7} \end{array}\right]$$

$$\left[\begin{array}{cccc} 7 & 5 & -3 & 14 \\ 0 & -\frac{50}{7} & \frac{23}{7} & -\frac{104}{7} \\ 0 & 0 & -\frac{128}{25} & -\frac{256}{25} \end{array}\right]$$

$$-\frac{128}{25} z = \frac{-256}{25} \qquad z = 2$$

$$-\frac{50}{7} y + \frac{23}{7} \times 2 = \frac{-104}{7} \qquad y = 3$$

$$7x + 15 - 6 = 14 \qquad x = 1$$

سوال ۴ـ از نقطه‌ی x یک نیم خط به مبدا x می‌کشیم. و مقدار نقاط برخورد نیم خط رسم شده با اولین شکل را می شماریم. در حالتی که مقدار برخوردها فرد باشد، نقطه x درون شکل بوده است و در غیر این حالت x خارج شکل است. اشتباه این سوال این است که در حالتی که x بیرون از شکل باشد، با وارد شدن هر نیم خط به شکل، قطعاً باید از شکل خارج شود، به این دلیل که شکل نامتناهی نیست. بنابراین اگر x خارج از شکل باشد، به ازای هر برخورد وارد شدن به شکل یک برخورد خروج از شکل هم داریم. که سبب می‌شود مقدار برخورد آن، زوج باشد.

سوال ۵) در ابتدا، در هر قسمت x^2 را در ضریب جمله ضرب می‌کنیم. و ضریب در ضریب جمله را از آن کم می‌کنیم.

$$f(x) = 2x^7 - x^5 + 3x^3 + x$$

$$x^2\left(x^2\left(2x^3 - x\right) + 3x\right) + x = x^2\left(x^2\left(x^2(2x) - x\right) + 3x\right) + x$$

$4x$

$x=2$

$$\frac{2}{2\times 2} \quad \frac{-1}{2^2\times 4 + (-1)(2)} \quad \frac{3}{2^2\times 14 + (3\times 2)} \quad \frac{1}{2^2 \times 62 + (1\times 2)} = 250$$

الگوریتم sody_Horner(P[0...n], x)

```
p ← P[n] x
for (i = n-1 ; i > 0 ; i--)
    p ← x^2 p + P[i] x
return p;
```

جمع الگوریتم = $A(n)=n$

ضرب Algorithm = $M(n)=2n$

پیچیدگی Algorithm از order n می‌باشد.

سوال ۶: از اولین عنصر Array آغاز می‌کنیم و عنصر را ریشه Heap است. و بعد آن را با عنصر دوم و سوم که فرزندانش اند، مقایسه می‌کنیم. در صورتی که عنصر اول بزرگ‌تر بود، به این معنی است که max-heap موجود است. پس در این صورت سایر عناصر را بررسی می‌کنیم. در صورتی که اندیس هر عنصر را می‌پیماییم تا اینکه به عنصر وسط Array برسیم. در صورتی همه‌ی این عناصر از عنصر $2i$ و $2i+1$ آن بزرگ‌تر بودند، به این معنی است که max heap موجود است. و البته در صورتی که عنصر اول از عنصر دوم و سوم کمتر بود، به این معنی است که باید min heap بودن یا نبودن آن را بررسی کنیم. که باید بار حرکت کردن روی آرایه Array کم بپیماییم

و بررسی کردن هر عنصر با اندیس i با عنصر $2i$ و $2i+1$ به عنوان فرزندانش، در صورتی که اگر آنها از عنصر i بزرگ‌تر بودند و عنصر پدر کمتر بود، این برای همه عناصر قبل وسط Array صدق کند، یا min heap موجود است، و اگر اینگونه نبود، با پیمایش همه‌ی Array

heap موجود نیست

سوال ۷: پیراهن = C دامن = F مقدار = Z $6C+4F+8Z$ سود

مقصود این سوال max کردن $6C+4F+8Z$، برای max کردن سود است. باید هر ۱۲ ساعت و هر طراحی به مقداری که توانایی دارم ساعت را به هر کدام از آنها مشخص می‌دهیم و این نقطه‌های ۳ معادله بالا، برقرار می‌باشد. و می‌تواند محصول‌ها ⊖ منفی باشد

طراحی: $\frac{C}{100}\times 3 + \frac{F}{100}\times 2 \leq 12$

برش: $\frac{C}{100}\times 1 + \frac{F}{100}\times 2 + \frac{Z}{100}\times 3 \leq 14$

دوخت: $\frac{C}{100}\times 2 + \frac{F}{100}\times 3 + \frac{Z}{100}\times 4 \leq 16$

سوال ۸: (در حالتی که v_i و v_j همسایه باشند، و به هم وصل باشند، یعنی یک مسیر به طول ۱ بین آن ۲ است) $\{v_i, v_j\} \in E$ اگر → $A_{ij}=1$

(در حالتی که v_i و v_j همسایه نباشند، و بین v_i به v_j وصل نباشد، بنابراین مسیری بین آن ۲ به طول ۱ موجود نیست) $\{v_i, v_j\} \in E$ اگر → $A_{ij}=0$

فرض استقرا: A_{ij}^{k-1} مقدار مسیرها را به طول $k-1$ را در میان v_i و v_j را تعیین می‌کند

حکم استقرا: مشخص می‌کنیم که A_{ij}^{k} مقدار و تعداد مسیرهای بین v_i و v_j و به طول k را نشان می‌دهیم

$$A^k_{ij} = (A^{k-1} \times A)_{ij} = \sum_{c=1}^{n} (A^{k-1})_{ic} \cdot A_{cj}$$

تعداد مسیرهای به طول k بین c و j را i می‌یابیم و بعد تعداد مسیرهای به طول k از این c و j را در آن ضرب می‌کنیم تا پاسخ آن به دست آید. و پاسخ تعداد مسیرهای به طول k بین i و j را به ماتریس