

سوال ۷ برای سازه‌ای از ۲ تا heap استفاده می‌کنیم. برای max و min.

کل عناصر در این سازه به ۲ نسبت تقریباً مساوی با اختلاف حداقل ۱ تقسیم می‌کنیم.

نیمه برتر در min heap و نیمه کوچکتر در max heap درج می‌کنیم.

حالا برای درج — عنصر را در حجم با اندازه‌ی کوچکتر درج می‌کنیم. اگر برابر بود هم فرق نمی‌کند.

اگر در حجم مشکلی نداشته و چون $O(\log n)$ باقی می‌ماند.

برای یافتن و حذف میانه — یا راس حجم بزرگتر است یا راس حجم کوچکتر.

(یعنی با کوچکترین عضو یا بزرگترین عضو یا کوچکترین عضو) تا اینجا در $O(1)$.

کار راه می‌افتد. اما اگر اندازه‌ها برابر بود، از ۲ تا راس میانه‌ها را می‌گیریم.

برای حذف میانه در حالت اول (در راس بزرگتر یا کوچکتر) و در حالت دوم (در راس بزرگتر یا کوچکتر) می‌کنیم.

که در $O(\log n)$ اولی می‌شود. در حالت دوم هم نیاز به حذف نیست چون عنصر میانه

وجود ندارد.

سوال ۱، هم نریخته (H_{max}) ، هم لخته (H_{min}) [از لحاظ اندازه] را در

تأثيره. في حدود قيم مختلفه حول بصورت درختي بياض سنك. (بداً r_{max} و r_{min})

• If $r_{\min} > r_{\max}$:

remove r_{\min} from H_{\min} .

insert k_{\max} into H_{\min}

$$H_{max} = \frac{0}{1}$$
$$r_{\max} = r_{\min}$$

then merge H_{\min} and remaining of H_{\max}

در نهایت ، H_{max} همان هم (مجموعه) است .

$$O(\text{height}(H_{\min})) + O(\sqrt{n}) \rightarrow O(\log n)$$

سوال ۹) از رشتہ شروع می کنیم تا رشتہ به نایافته که null شویم و به انتهای برسیم، در درخت پائین می رویم

حیرائی والے ہیچ ایسی ہی کہیں، بعد از فرزنداش راندت می اور کم وبہ بعد از کے از ابتدائیں،

امانی می کنیم - هزینه خارج عمل $O(\log n)$ با ارتفاع درخت است ، چون به عمق $\log n$ می رسد ، نه در هر از برای می کنیم

if $x.\text{value} < k$

ب. باب اللزیم بابر کشتی سوال روحانی فی لیم

BST less than (left(x), k)

else if $x.value == k$ && $left(x) \neq null$:

$$\text{count} = \text{Left}(x) + \text{keys}$$

else ::

$$\text{Count} = \text{left}(x) \cdot \text{keys} + 1 + \text{BSTlessThan}(\text{Right}(x), k)$$

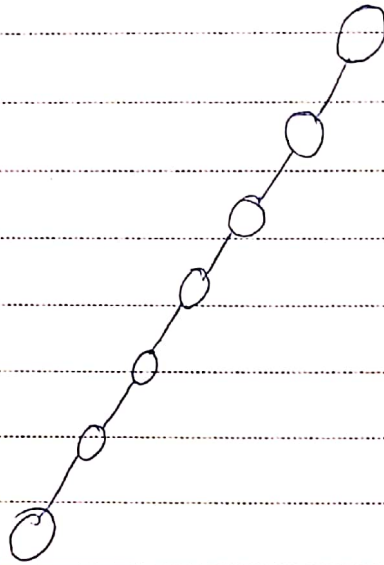
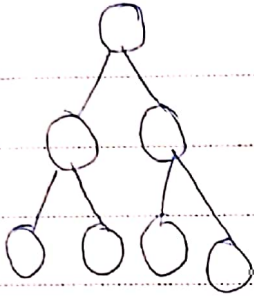
P4PCO

$$T(n) = T\left(\frac{n}{4}\right) + O(1) \rightarrow T(n) = O(\log n)$$
~~$$T(n) = T(n-1) + O(1) \rightarrow T(n) = O(n)$$~~

ما تليق

150

سوال ۱۰، حد اعلیٰ ارتفاع ۲ الب۔ عناصر خوش سیرب باشند



حدائق ارتفاع ۴ - حالت زنجیری

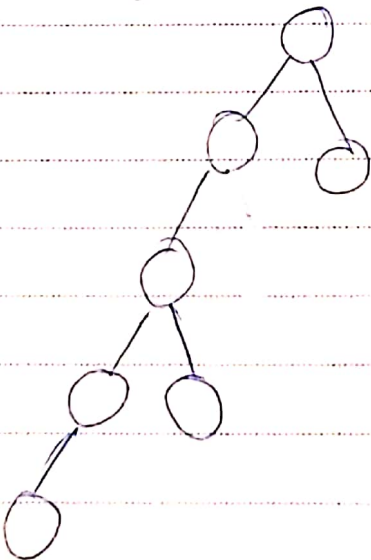
پ) درخت AVL با توجه به روشی که داده (اختلاف در الگ) در ارتفاعها ، ارتفاعش فقط ۲ یا ۳ باشد. اگر ۴ باشد، یکی از حالتها به شکل زیر باشد که

مِنْ مَعَالِمِ قَبْلِ قَدَمِ

رسیده با آما عصر، ارتفاعی شهر از ۳ دانستم با هم.

و باعث می‌شود که چرخش لازم داشته باشیم ماد و باور استناد دارد

2



۱۰- پ، اگر وی BST، یک پیمایش در ترتیب (inorder) انجام دهم، آن به ترتیب

شده اعداد رو به من میبندد. حالاً بدوینم که درج در AVL با چرخش ها، $O(n \log n)$ هزینه دارد.

پس اگر رادر AVL با این هزینه درج می کنیم و سوال حل می شود.

۱۰- ت، یک رابطه بازگشتی به صورت زیر وجود دارد:

برای حالت اول: $P(h=0) \rightarrow 1$

$P(h=1) \rightarrow 2$

$$O.W. \quad 1 + P(h-2) + P(h-1)$$

کمترین تعداد نود را دارد
ماجرای کنیم

حداکثر: سطح نام می تواند حداکثر ۲ تارو داشته باشد. جمع زدن روی سطوح

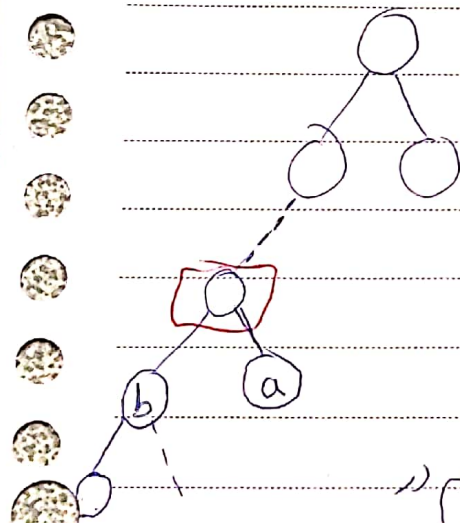
$$\begin{matrix} h+1 \\ 2-1 \end{matrix} \quad \begin{matrix} 2^0 + 2^1 + 2^2 + \dots + 2^h \end{matrix}$$

۱۲- یک حله اولیه نیاز داریم که بسیم آیا اصلاً a و b یزدخت هستن یا نه؟

این حله $O(\log n)$ هزینه می‌بره. حالا باز روی BST مان، یک نمایش بین ترتیب انجام می‌دهیم تا اریک مرتب شده را برداریم. [فقط لازم است این کار را روی a انجام دهیم.

یعنی از راس a برس ط پیمایش را انجام می‌دهیم $O(\log n + k)$

۱۳- از راس انتقد باین می‌رویم تا جایی که حقوق ۲ عنصر a و b در یک طرف راس باشند.



در پیمایش مان تا جایی که ۲ عنصر فرزیرسیم، فقط از یزدخت‌ها استراتژی عبور می‌کنیم، اصلاً به چپ‌ها کار نداریم. [چون فرزیرین

را می‌خواهیم] بعد از اینکه عنصر فرزیریک را یافتیم، یک بار یزدخت

راستش و یک بار یزدخت چپش را پیمایش می‌کنیم و ۲ عنصر باقیمانده در

میرهارا می‌یابیم. در نهایت بین این ۲ عدد، باز Max می‌گیریم و خروجی ماسخ می‌گردد.

بجای زمانی هم از $O(n)$ است در بدترین حالت.

۱۴- از سمت راست آرایه مان شروع می کنیم و دوباره عناصر و درخت AVL (درج می کنیم).

rotation های لازم هم انجام می دهیم تا درخت balanced باقی بماند.

حالا برای هر عضو، ام آرایه، حله می کنیم. آیا Node ما زیر درخت چپ دارد یا نه. اگر دارد.

تعداد Node های حاضر در زیر درخت چپ را به اسم "Gunt" اضافه می کنیم.

بعد این بودا باید درش مقایسه می کنیم، اگر بزرگتر از درش بود، اندازه بود درش و با یک زیر درخت چپ

آن را به Gunt اضافه می کنیم. چنانچه طور بالا تر می رویم تا به root برسیم. عدد آخر Gunt، همان

جواب ما است.

پسیدنی فانی: $O(n)$ برای insert کردن آرایه - $O(\log n)$ برای ساخت AVL و میزان کردن آن

چون که از Node بالای آرایه تا ریشه، $O(\log n)$ زمان می برد پس در مجموع $O(n \log n)$.

بزرگترین عدد بین اعداد، همان $O(n \log n)$ است پس پاسخ نهایی ما هم به همین صورت

می شود.

۱۵- می دانیم که اگر مرتب سازی (درجی) برای آرایه با اندازه k از $O(k^2)$ است.

$\frac{n}{k}$ هم زرا را به داریم که باید با این نوع صورت ، مرتب کنیم $\rightarrow O(k \times \frac{n}{k}) = O(nk)$

$O(n \log(\frac{n}{k}))$

برای ثبت مرتب سازی سریع:

حاصل از
مرتب سازی
سریع

$\log n - \log k$
↓
حالت
عادی
ارزایی مرتب سازی
(درجی) استفاده نشده

این زرا را به عنوان Pivot
در مرتب سازی سریع به کار می آورند.

$O(n \log(\frac{n}{k}) + nk)$

در کل:

به ازای $k = \begin{cases} 1 \\ \log n \end{cases}$ ، به ازای خواسته شده یعنی $O(n \log n)$ می آید