پاسخ امتحان سوم - حریصانه

طراحی الگوریتم - بهار ۱۴۰۰

● بخش اول

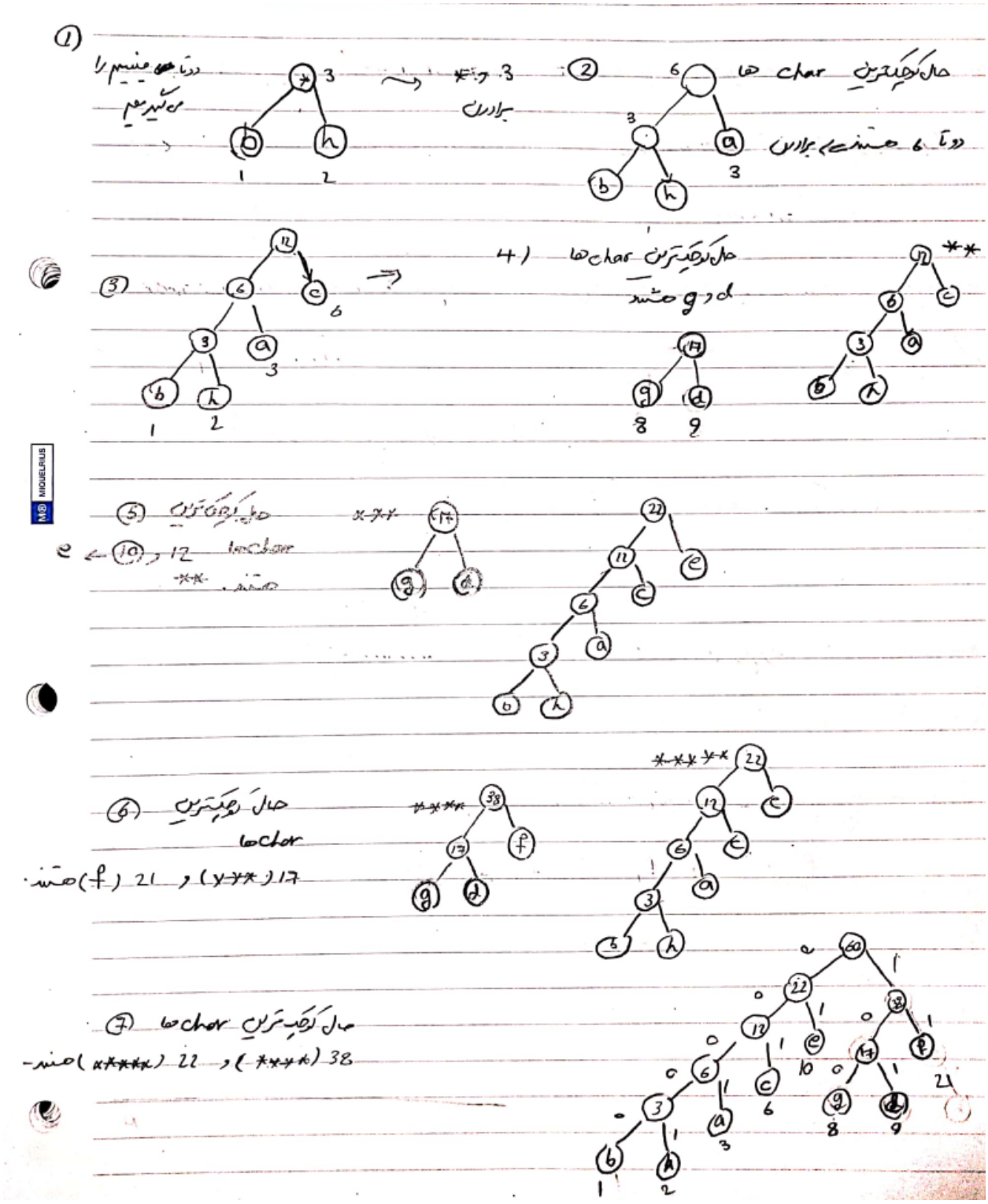
سوال اول:

جواب را برای شماره دانشجویی نمونه **810198543** بررسی می کنیم.

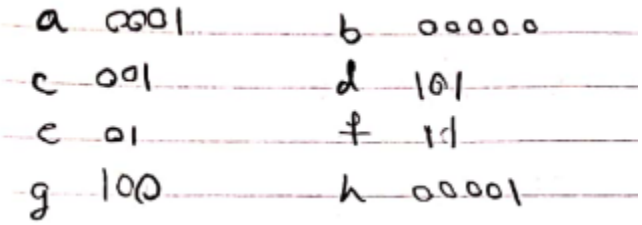
جدول فراوانی:

f_a	f_b	f_c	f_d	f_e	f_f	f_g	f_h
3	1	6	9	10	21	8	2

مراحل اضافه شدن رئوس به درخت:



در نهایت کد اختصاص یافته به هر حرف برابر است با:



● بخش دوم

سوال دوم:

(الف) n عدد گرفته شده را مرتب می‌کنیم. (هزینه‌ی زمانی O(nlogn) است.) فواصل مرتب شده‌ی گل‌ها را به ترتیب x1, x2, …, xn نام‌گذاری می‌کنیم. متغیر ind که اندیس اولین گل آبیاری نشده است و در ابتدا ۱ است را داریم. در هر مرحله در مختصات $x_{ind} + k$ یک فواره گذاشته و آنقدر ind را یکی یکی زیاد می‌کنیم تا x_{ind} از محدوده‌ی فواره‌ی قبلی خارج شود یا گل‌ها تمام شده باشند. ind از n بزرگ‌تر شده باشد الگوریتم به پایان رسیده و در غیر این صورت دوباره فرآیند را اجرا می‌کنیم. از آن‌جایی که ind از ۱ تا n می‌رود و هر زیاد شدن آن از O(1) است، پس الگوریتم ما از O(nlogn) خواهد بود.

(ب) از بین تمامی جواب‌هایی که کم‌ترین تعداد فواره‌ی ممکن را دارند، نزدیک‌ترین جواب را به جواب ما در نظر می‌گیریم. منظور از نزدیک‌ترین جواب، جوابی است که اگر مختصات فواره‌های آن را مرتب کنیم و به ترتیب آن‌ها را با جواب خود مقایسه کنیم، دیرتر تفاوت را پیدا کنیم.

فرض کنید مختصات فواره‌های ما a1, a2, …, ah باشد و مختصات فواره‌های جواب b1, b2, …, bm باشد. اولین جایی که این دو تفاوت دارند نیز در اندیس i ام باشد. طبق الگوریتم گفته شده، در مکان $a_i - k$ حتما گلی وجود دارد که آبیاری نشده است. بنابراین چون جواب مسئله باید این گل را نیز آبیاری کند، پس $b_i < a_i$

است. از آن جایی که تمامی a و b های قبل از i تماما با هم یکی هستند، پس تمامی گل‌های قبل از گل $a_i - k$آبیاری شده‌اند. پس با جلو بردن b_iمشکلی ایجاد نخواهد شد. b_iرا روی a_i گذاشته و جوابی نزدیک‌تر به جواب ما پیدا می‌کنیم که با انتخاب نزدیک‌ترین جواب به جواب ما تناقض دارد.

سوال سوم:

(الف) فواره‌ها را بر اساس کم‌ترین مختصات آبیاری‌شان مرتب می‌کنیم. (مرتبه زمانی (O(nlogn) متغیر k داریم که در ابتدا ۰ و پس از آن، نقطه‌ای است که خودش آبیاری شده ولی به ازای هر $\epsilon > 0$، $k + \epsilon$ آبیاری نشده است را نشان می‌دهد. حال فواره‌ها را تا جایی که k را پوشش دهند مشاهده می‌کنیم و از بین این فواره‌ها فواره‌ای را روشن می‌کنیم که مختصات انتهایی پوشش دهنده توسط آن بیش‌تر از همه باشد. این فواره را روشن کرده و بقیه‌ی فواره‌ها را دیگر کاری نداریم. k نیز برابر با نقطه‌ی پایانی فواره‌ی روشن شده می‌شود. حال دوباره همین کار را انجام می‌دهیم تا k بیش‌تر مساوی m شود. از آن جایی که هر فواره را یک بار بررسی کردیم، مرتبه‌ی زمانی کلی الگوریتم O(nlog(n)) خواهد شد.

(ب) از بین جواب‌ها نزدیک‌ترین را به جواب ما در نظر بگیرید. منظور از نزدیک‌ترین جوابی است که اگر فواره‌های روشن شده‌اش را بر حسب مختصات شروع آبیاری‌شان مرتب کنیم، اولین اختلاف بین جواب ما و جواب اصلی دورتر از بقیه جواب‌ها باشد.

فرض کنید فواره‌هایی که روشن شده‌اند در روش ما به ترتیب a1, a2, …, ap باشند و در جواب بهینه b1, b2, …, bp باشند و اولین اختلاف این دو در اندیس i ام باشد. طبق الگوریتم، k ای که قبل از روشن کردن i امین فواره داریم، اولین جاییست که بعد از آن آبیاری نشده است. پس bi حتما باید k را پوشش دهد. اگر k را پوشش ندهد پس چون مرتب شده‌اند، بازه‌ای قبل‌تر از k که آبیاری شده بود را دارد پوشش می‌دهد که نیازی به این کار نیست و با حذف کردنش می‌توان جواب بهتری پیدا کرد و این تناقض است. پس حتما k را پوشش می‌دهد. طبق الگوریتم ما،

انتخاب ما بین تمامی بازه‌هایی که k را پوشش می‌دهند، بازه‌ای بود که دورترین نقطه را آبیاری می‌کند. حال از آن‌جایی که طبق الگوریتم، هم ما و هم جواب بهینه همه‌ی بازه‌ی صفر تا k را آبیاری کرده‌اند، اگر به جای bi از ai استفاده کنیم مشکلی نخواهیم داشت و این با انتخاب نزدیک‌ترین جواب بهینه تناقض دارد.

● بخش سوم

سوال چهارم:

(الف) سکه ها را به صورت صعودی مرتب می کنیم. روی اعداد حرکت میکنیم. در هر لحظه عدد m را نگه می داریم که نشان دهنده آن است که با سکه های موجود میتوان هزینه 1 تا m را پرداخت کرد. فرض کنید عدد بعدی k باشد:

(1) اگر $k < m$ آنگاه تنها $m = m + k$ می کنیم

(2) اگر $m < k$ آنگاه m+1 را به مجموعه جواب ها اضافه کرده و $m = 2m + 1$

با توجه به به توجه به میزان تغییر m در حالت دوم (دو برابر شدن) و همچنین حرکت بر روی مجموعه اعداد هزینه این الگوریتم برابر با $O(n + log(k))$خواهد بود.

(ب) بدین منظور فرض کنید راه ما با راه بهینه تفاوت داشته باشد. راه بهینه را شامل مجموعه سکه $\{y_1, y_2, ..., y_i, ..., y_n\}$باشد و فرض کنید پاسخ حریصانه مجموعه $\{x_1, x_2, ..., x_i, ..., x_n\}$ باشد. فرض کنید اولین اختلاف بین دو پاسخ در ایندکس i ام باشد. حال دو حالت در اینجا داریم: یا اینکه $x_i > y_i$ باشد یعنی $x_i > m$ پس خود m را نمی توان ساخت که این تناقض است.

در غیر اینصورت در راه بهینه می دانیم میتوان تا مقدار $m + x_i - 1$ ساخت و در راه حریصانه تا مقدار $m + y_i - 1$ را ساخت. می دانیم $m + y_i - 1 > m + x_i - 1$ پس با تعویض x_i و y_i

این راه همچنان درست باقی می ماند اما اختلاف آن با راه حریصانه یک واحد کمتر می شود که این اشتباه است زیرا بدان معنی است که راه ارائه داده شده بهینه نبوده است.

سوال پنجم:

(الف) فرض کنید گراف بازه را ایجاد می‌کنیم. (دو بازه به یکدیگر یال دارند به شرطی که با یکدیگر اشتراک داشته باشند). حال بر اساس ابتدای هر بازه، بازه ها را مرتب می کنیم. از ابتدا شروع می کنیم. برای مشخص کردن رنگ هر بازه اولین رنگی که بین همسایه های رنگی اش نیست را به آن اختصاص می دهیم(این الگوریتم به الگوریتم حریصانه رنگ آمیزی گراف معروف است). برای انتخاب رنگ از بین رنگ های باقی مانده در هیپ رنگ های متفاوتی که داریم را می گذاریم. همچنین بازه ها را بر اساس پایانشان نیز مرتب می کنیم و وقتی از یک بازه رد شدیم (اشاره گر روی آرایه اول از آن ها بیشتر شد) رنگ را به هیپ اضافه می کنیم و رنگ جدید را از سر هیپ برداشته و استفاده می کنیم(در واقع همیشه رنگ با شماره کمتر انتخاب می شود).

اما در بررسی اردر زمانی این الگوریتم می توان گفت که دو بار مرتب سازی داریم هر کدام $O(nlog(n))$. روی بازه حرکت می کنیم و در هر مرحله اضافه کردن و حذف از هیپ را داریم که باز هم در کل $O(nlog(n))$می شود.

(ب) شبیه ترین پاسخ بهینه به جواب حریصانه را در نظر میگیریم. فرض کنید اولین اختلاف دو پاسخ در ایندکس i ام باشد پس متوجه می شویم که امتحان i ام در دو جواب در دو کلاس متفاوت برگزار شده اند. فرض کنید در پاسخ بهینه در کلاس x و در پاسخ حریصانه در کلاس y برگزار شده باشند. حال چون بازه ها بر اساس شروع مرتب شده بودند پس در حالت زیر:

کلاس x $[s_p, f_p]$ $[s_i f_i]$ $[s_j, f_j]$

کلاس y $[s_t, f_t]$ $[s_i f_i]$ $[s_k, f_k]$

داریم $s_i \leq s_j$ و $s_i > f_t$ پس $f_t < s_j$ و طبق حریصانه می دانیم $f_t < s_i$پس می توان کلاس y را از s_i به بعد با کلاس x از s_j به بعد عوض کرد در این صورت جواب ثابت و شبیه تر شده که تناقض دارد و حکم ثابت می شود.