

1. تجزیه اعداد صحیح به عامل‌های اول آنها

برای این سوال من برنامه رو جوری نوشتم که داخل حلقه اصلی for برای هر عدد اولی که می‌گیره هم چک کنه ایا عدد ورودی به عدد اولمون بخش پذیره یا نه و هم عدد اول بعدی رو می‌سازه

کلیت تقسیمم اینطوره که میاد می‌گه تا وقتی عدد اولمون بر عددمون بخش پذیر هست عدد اولمون رو بر اون تقسیم کنه و خارج قسمت رو برابر عددمون بزاره

با این کار وقتی دوباره می‌رسیم به اول حلقمون و می‌دونیم عددی که داریم بر اعداد اول قبل از عدد اولی که تازه گرفتیم دیگه بخش پذیر نیست. پس عددمون در صورتی می‌تونه تجزیه بشه یا اینطور بگیریم در صورتی می‌تونه عددی اول نباشه که ضربی از اعداد اول بزرگ تر یا برابر عدد اول جدیدمون باشه

برای همین گفتم اگه عددمون کوچیک تر توان 2 عدد اولمون بود نتیجه بگیره عددمون یه عدد اوله و اون رو به شرط بزرگ تر از 2 بودن توی لیست عامل ها بزاره و بعد از اونم از حلقمون کاملاً خارج بشه

قسمت بعدشم که عدد اول بعدی رو پیدا می‌کنه هم چیز خاصی نداره

فقط اون قسمت که گفتم اگه عددمون برابر 2 نبود بیا بعلاوه 2 کن اگه برابر 2 بود بعلاوه یک کن

هدفم صرفاً این بود تعداد محاسبات کمتر بشه چون می‌دونیم اعداد اول بجز 2 همشون فرد هستن و عدد فرد به علاوه یک میشه عدد زوج و دیگه عدد اول نیست

2. بسط فاکتوریلی اعداد طبیعی

برای سوال 2 هم من اینجوری برنامه رو نوشتم که بیاد بصورت جستجوی باینری ای که سر کلاس بهمون یاد دادید نزدیک ترین فاکتوریل رو به عددمون پیدا کنه

برا همین یه تابع مقایسه باینری نوشتم که یه تابع می گیره و یه عدد و بزرگ ترین عدد صحیحی که تابع می تونه بگیره و خروجیش کمتر از عددمون باشه رو پیدا می کنه

من فکر می کردم این راه بهینه تریه نسبت به اینکه بیایم از اول یکی یکی فاکتوریل اعداد رو محاسبه کنیم تا برسیم به بزرگ ترین فاکتوریلی که از عددمون کوچک تره

ولی اینجوری هم روش بهینه ای نیست چون محاسبه فاکتوریل های کوچیک خیلی ساده تر از محاسبه فاکتوریل های بزرگه

3. محاسبه کوچک ترین مضرب مشترک چند عدد

سوال 3 رو هم اقا مسعود راهنمایمون کرد که گفت ضرب 2 تا عدد تقسیم بر بزرگ ترین مقسوم الیهشون برابر با کوچک ترین
مخرج مشترکشونه

البته توی دبیرستانم اینو زیاد داشتیم

و خب پیدا کردن بزرگ ترین مقسوم الیه رو هم که توی کلاس یاد گرفته بودیم

4. مرتب سازی تیم ها

برای این سوال هم اومدم همه جایگشت های تیم ها رو پیدا کردم و یکی یکی چک کردم که کدومشون در حالت درستی هستن قبلش می خواستم خودم یکی یکی تیم ها رو بچینم ولی نتونستم. برای همین تصمیم گرفتم با این روش برم و تابعم اینجوری نوشتم که اگه یه درخت فرض کنیم رئوس قبلیشو بهش دادیم و همینطور ادامه میده تا می رسه آخرین راس و وقتی به آخرین راس رسید کل رئوس قبلی رو بعلاوه آخرین راس داخل لیست ذخیره می کنه یعنی تمام آخرین راس ها مسیری رو که تا آخرین راس رسیدن رو توی لیست ذخیره می کنن

5. انتگرال گیری عددی

برای سوال 5 هم چون من قبلا المپیاد فیزیک خونده بودم و یکم با تقریب و بسط ها آشنا بودم تا حدودی می دونستم چطوری توابع رو می تونیم تخمین بزنیم

برا همین من برای تابع سینوس از تناوبی بودن تابع استفاده کردم

و توضیحش رو هم توی خود کد دادم

و برای تابع رادیکالی هم کاری که کردم این بود عدد زیر رادیکال رو بصورت جمع 2 عدد نوشتم که یکیشون بزرگ ترین توان 2 یک عدد صحیح که برای پیدا کردنش هم دوباره از مقایسه باینری ای که توی سوال 2 استفاده کرده بودم استفاده کردم و ما وقتی این عدد رو از داخل رادیکال فاکتور می گیریم و میاریمش بیرون، زیر رادیکال یه دونه یک بعلاوه یه عدد تقسیم بر اون عددی که فاکتور گرفتیم میشه

و این خیلی خوبه چون به شکل یک بعلاوه ایکس هست و ایکس هم برای اعداد بزرگ نزدیک به صفر میشه و توی کتاب های حساب دیفرانسیل مثل توماس و یا ویکی پدیا می تونیم تقریب این عبارت رو پیدا کنیم که من توی ویکی پدیای رادیکال ایکس پیدا کردم

https://en.wikipedia.org/wiki/Square_root

https://wikimedia.org/api/rest_v1/media/math/render/svg/5e4f533de7fa291ba3d5837e3db4b7cb61ca9cf4

6. تجزیه شهر

مسئله 6 هم مسئله خوبی بود

من اول اومدم شهر رو به مربع های یک در یک تقسیم کردم و بعد گفتم هر کدوم از این مربع ها به کدوم فروشگاه نزدیکه

بعد ریختمشون توی 3 تا ست و اشتراکشون رو 2 به 2 پیدا کردم

اینجوری مرز بین فروشگاه ها پیدا شد

و خب خوبم بود

کدش رو هم گذاشتم توی فایل

ولی سوال گفته بود راس ها رو پیدا کنیم

و اینجوری نمی شد

اومدم گفتم راس ها رو میشه گفت اشتراک 2 تا مرز هستن به این صورت که یا اشتراک مرز های 3 فروشگاه

و یا اشتراک مرز 2 تا فروشگاه با مرزش شهر

ولی وقتی برنامه رو با یه سری مقادیر امتحان کردم کار می کرد و با یه سری دیگه کار نمی کرد و خیلی نقاط رو پیدا نمی کرد

و اینجا یه نکته خیلی خیلی جالب و مهمی داشت

اینم این بود چون مرز هامون از مساحت های یک در یک تشکیل شدن و گسسته هستن و نه پیوسته

ممکنه 2 تا مرز از روی هم دیگه رد بشن ولی اشتراکی نداشته باشن

مثل یه صفحه شطرنج می مونه

خط ها ممکنه بهم برخورد کنن ولی باهم روی یک نقطه اشتراکی نداشته باشن چون خطی نیست

نقاط گسسته هستن

و بعد اومدم از طریق عمود منصف رفتم

گفتم نقاطی راس هستن که یا عمون منصف ها هم دیگه رو قطع کرده باشن و یا عمود منصف ها و مرز شهر

و معادلاتش رو نوشتم

ولی یه چالشی داشت اونم وقتی بود که شیب عمود منصف بی نهایت می شد

و اونجا تقسیم بر صفر داشتیم و به مشکل می خوردیم

روی این زیاد فکر کردم و به این نتیجه رسیدم که بیایم از سوال 10 استفاده کنیم و ضرایب معلوم ها و مجهول ها رو بدیم و نقطه برخورد رو پیدا کنیم

و برای اینکه مشکل تقسیم بر صفر رو حل کنم اومدم گفتم بجای اینکه فقط یه دونه شیب داشته باشیم که ضریب دلتا ایکسمون باشه بیایم یه ضریب برای دلتا ایکس تعریف کنیم و یه ضریب هم برای دلتا وای و شیبمون اینجوری میشه ضریب ایکس تقسیم بر ضریب وای که مساویه با دلتا وای تقسیم بر دلتا ایکس

و بعد هم طبق توضیحاتی که توی کد ها نوشتم ضرایب رو به ماتریس دادم و نقطه ها رو پیدا کردم

7. ساخت رشته‌ها

سوال هفت رو هم بصورت یه تابع بازگشتی نوشتم

اینطور که بصورت یه درخت مسئله رو در نظر گرفتم که توی مرحله اول n تا شاخه میشه

و هر شاخه ای هم دوباره n شاخه

و تابع رو هم طوری نوشتم که رشته قبلیش رو هم بگیره

پس تا وقتی طول رشته کمتر از طول ورودیمون باشه تابع باز هم خودش رو صدا می زنه

در غیر این صورت متوقف میشه

و هر بار هم چک می کنه اگه رشتمون کاراکتر تکراری ای رو نداشت داخل لیست رشته ها ذخیره کنه

اینو شاید بهترم می شد بنویسیم

یه تابع برای ترکیب حروف و یک تابع هم برای جایگشت حروف بنویسیم

و بعد با استفاده از این 2 تا تابع همه رشته ها رو به دست بیاریم

8. ارزیابی عبارت‌های پسوندی

سوال هشت رو هم به این صورت نوشتم که ورودی رو با فاصله از هم جدا کردم و تبدیل به یه لیست کردم و سپس یه لیست اعداد تعریف کردم و بعد گفتم از اول لیست ورودی‌مون شروع کن اگه عدد بود به این لیست اعداد اضافه کن

و وقتی رسیدی به یکی از عملگر ها بیا اون عملگر رو روی 2 تا عدد آخری که گرفتی اعمال کن و بعد اون 2 تا عدد رو حذف کن و بجاش بیا نتیجه رو به آخر لیست اضافه کن

9. دوره زمانی طلایی

سوال 9 هم اول میاد دوره ها رو 2 به 2 باهم دیگه مقایسه می کنه و اگه سال تولد یکی با مرگ دیگری برابر بود یکی از مرگ اون یکی کم می کنه

بعدم میاد تمام دوره ها رو به سال تبدیل می کنه و میریزه توی یک لیست و همزمان کوچیک ترین سال و بزرگ ترین سال رو هم مشخص می کنه که بعدا برا نوشتن حلقه به کارمون میاد

بعدش میاد سال هایی که بیشترین تکرار رو توی لیست داشتن پیدا می کنه و توی یه لیست با تاپل های 2 تایی ذخیره می کنه که این تاپل ها عنصر اولشون تعداد تکراره یا همون تعداد دانشمند های زنده توی اون سال

که خب اون سال ها میشه سال های دوره های طلائیمون چون هرچقدر تکرار بیشتر بوده باشه یعنی توی اون سال تعداد بیشتری دانشمند زنده بودن

می تونستم تعداد دانشمند های زنده رو حذف کنم چون نیازی نیست ولی چون اخر کار میخواستم ببینم برنامه درست کار می کنه یا نه گذاشتم باشه

بعدش میاد این سال ها رو به دوره تبدیل می کنه به این صورت که تا وقتی سال ها پشت سر هم باشن اونا رو توی یه دوره قرار میده ولی وقتی فاصله بینشون بیفته قبلیا رو توی یه دوره قرار میده و ادامهش رو یه دوره جدید حساب می کنه

و بعدم دوره ها رو با دوره ی هر دانشمند چک می کنه که اگه سال تولد دانشمند بین دوره مورد نظر بود و مرگش هم بعد از دوره مورد نظر بود این دانشمند جزو دانشمندی های دوره طلائیمون حساب میشه

10. حل دستگاه‌های معادلات خطی

سوال 10 هم چیز خاصی نداشت

صرفاً طبق همون توضیحاتی که توی پی دی اف داده بودید جلو رفتم فقط همین حذف سطر و ستونش مهم بود که اون رو هم
یه آموزش دیده بودم بودم با این شیوه نوشتن آشنا شدم و اینجوری لیست جدید رو تشکیل دادم