گاهنامه علمی خبری عصر رایانه

سال یازدهم، شماره ۲۱ - مهر ۱۳۹۸

behdad

#خبر_های_خوبی_در_راه_است

انجمن علمی کامپیوتر دانشگاه صنعتی خواجه نصیرالدین طوسی برگزار میکند:

Capture The Flag

PROGRAMMER

NOUN-[PRO-GRAM-MER]

1. A person who solves problems you can't.

2. One who does precision guess work based on unreliable data provided by those of questionable knowledge.

See also: WIZARD, MAGICIAN

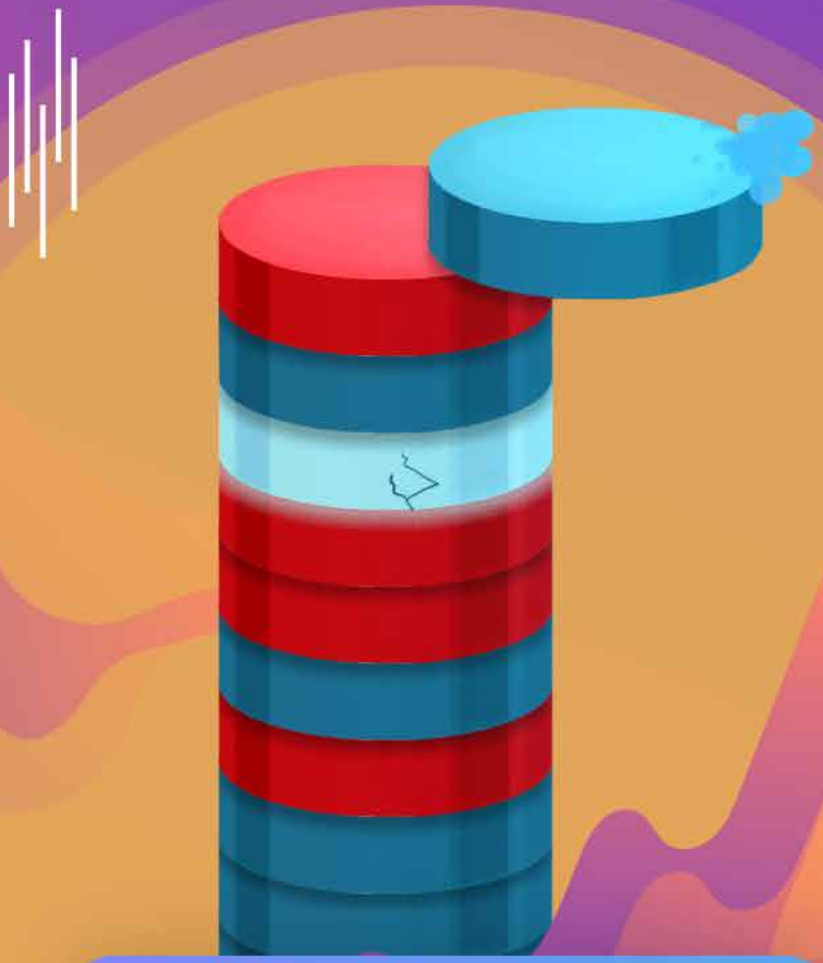
تیم اول: ۵ میلیون ریال
تیم دوم: ۳ میلیون ریال
تیم سوم: ۲ میلیون ریال

راه های ارتباط با ما و کسب اطلاعات بیشتر:

@KntuCTF

KNTU acm Chapter

MADNESS
GAME STUDIO



Orbis

@MadnessStudio

@MadnessGameStudio

@MadnessGameStudio

برای نصب بازی اسکن کنید

بازار

گاهنامه علمی خبری
عصر رایانه

سال یازدهم، شماره ۲۱
مهر ۱۳۹۸ - ۲۴ صفحه

صاحب امتیاز: انجمن علمی کامپیوتر

حامیان مالی :

behdad

فهرست مطالب

مقدمه

به لطف خدا شماره‌ای دیگر از نشریه آماده شد. در این شماره چند مقاله آماده شده است که تلاش کرده‌ایم موضوعات مختلف را پوشش دهیم. در کنار مقالات علمی این بار مصاحبه‌ای با یکی از دانشجویان موفق کامپیوتر خواجه نصیر تدارک دیده‌ایم تا تجربه ارزشمندش را با شما شریک شویم. اما از این ها هم که بگذریم بد نیست روی جلد را نگاهی بکنیم...

۲۱ امین شماره نشریه عصر رایانه... به راستی که مسیر پر پیچ و خمی بوده است. ۲۰ شماره در طول سال‌ها و در دوره‌های متفاوت آماده شد. هرکدام با یک نگرش و نگارش نوشته شد، اما همگی با یک هدف مشترک: ایجاد علاقه، انگیزه و پویایی در دانشجوها.

در طول دوره های مختلف دانشجویانی که به دنبال دانشکده‌ای بهتر بودند، جمع شدند و در کنار یکدیگر تلاش کردند. کلاس، کارگاه، رویداد و مسابقه برگزار کردند تا دانشکده‌ای پویاتر و قوی‌تر داشته باشند. تمام این فعالیت‌ها به کمک همراهان انجمن امکان‌پذیر شد. دوستانی که بدون چشم‌داشت پا به پای اعضای انجمن از وقت خود گذشتند تا اثری بگذارند که بماند. اگر انجمن علمی کامپیوتر امروز موفق است و سابقه‌ای درخشان دارد به لطف اعضا و دوستداران دوره‌های پیشین است. اما این روزها انجمن در برابر انبوه فعالیت‌ها دست‌تنها مانده است. انجمن این روزها به کمک دانشجویانی نیاز دارد که آرزوهای بزرگ داشته باشند تا بتواند بیشتر از گذشته بدرخشد و این درخشیدن بدون کمک شما همراهان انجمن امکان پذیر نیست. مسابقات، رویدادها، همایش‌ها و کلاس‌هایی که برای ماه‌های آینده برنامه‌ریزی شده است، نیازی مبرم به کمک دارد. کمکی که تنها از دست دانشجویان مشتاق رشد و پیشرفت برمی‌آید.

و سخن آخر... نام خواجه نصیر چه خوب چه بد تا آخر عمر همراه ما در مدارک تحصیلیمان خواهد بود. اگر این نام بدرخشد باعث اعتبار ماست و اگر این نام معتبر نباشد باز این ما خواهیم بود که ضرر می کنیم. پس به دنبال اثر مثبت بوده‌ایم. اگر شما نیز دغدغه‌ای بزرگ از جنس تغییر و اثر گذاشتن دارید، اکنون وقت آن است که اثر بگذارید.

منتظرتان هستیم.

ترجمه و گردآوری: زهرا عظیمی پور / The Story Of This Big Company

داستان این کمپانی بزرگ، در سال ۱۹۹۵ در دانشگاه استنفورد آغاز می‌شود. زمانی که Sergey Brin و Larry Page در دانشگاه استنفورد با هم دیدار می‌کنند. سِرگی دو سال قبل وارد دانشگاه شده و قرار است به عنوان راهنما، آنجا را به دانشجویان تازه ورود که لری هم جزء آن‌ها است نشان دهد.

پیج می‌گوید:« سرگی ابتدا آدم مزخرفی به نظرم آمد.»، برین نیز تایید می‌کند و در حقیقت هر دوی آن‌ها از یک‌دیگر متنفر بودند.

در ظاهر، آنها هیچ نقطه اشتراکی نداشتند:

پیج شخصی ساکت و ملاحظه‌کار بود و برین شخصی برونگرا، اجتماعی و پر سر و صدا؛ پیج شخصی بسیار متفکر و تحلیل‌گر و برین حلال مسئله بود.

اما آن دو در آن روز بیشترین نقاط اشتراک را در میان افراد نیز داشتند: خانواده‌ی هر دوی آنها تحصیلات آکادمیک داشتند؛ پدر پیج پروفسور پیشگام در زمینه علوم کامپیوتر در دانشگاه میکان استیت و مادر او مدرس برنامه نویسی کامپیوتر بود؛ پدر برین پروفسور ریاضیات در مری‌لند و مادر او محققی در اولین مرکز پروازهای فضایی در ناسا بود؛ بنابراین هردوی آنها به قدرت دانش برای غلبه بر هر چالش تئوری و عملی باور داشتند.

فعالیت اصلی آن‌ها از خوابگاه آغاز می‌شود. در ژانویه ۱۹۹۶ آنها تصمیم می‌گیرند با هم پایان‌نامه دکترای خود را بنویسند. موضوع این پایان‌نامه یک پروژه تحقیقاتی برای روش‌های جمع‌آوری اطلاعات از اینترنت بود. آن‌ها فرضیه‌ای را مطرح می‌کنند: "استفاده از لینک‌های خارجی برای تعیین اهمیت صفحات شخصی بسیار کارآمدتر از استفاده از لغات کلیدی است."؛ سپس یک موتور جست و جو بر اساس همین الگوریتم ساخته و نام آن را backrub می‌گذارند. البته این موتور جست و جو در آن زمان فقط برای دانشجویان استنفورد قابل استفاده بود. در پاییز ۱۹۹۷ لری و سرگی از روش طوفان مغزی برای پیدا کردن نام مناسب برای موتور جست و جوی خود استفاده کردند. آنها به سراغ دوستشان اندرسون رفتند و او نام googolplex را به آن‌ها پیشنهاد کرد که به معنای عدد یکی است که جلوی آن ۱۰۰ تا صفر قرار دارد. آنها از این نام خوششان می‌آید. اندرسون می‌گوید:« من کلمه googol را با غلط املایی به شکل google نوشتم و لری عصر آن روز همان نام را ثبت کرد.» مدت زیادی نمی‌گذرد که یکی از همکاران آن‌ها متوجه این موضوع می شود؛ اما تعمداً همین نام را تثبیت می‌کنند.

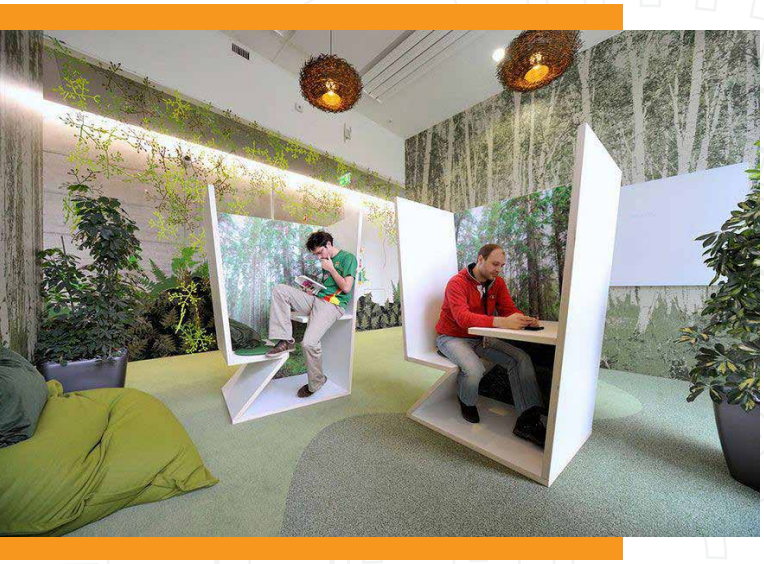
ابتدا هاست آن با دامنه‌ی google.stanford.edu ایجاد شده و در سپتامبر ۱۹۹۷ google.com ثبت می‌شود.

در نیمه‌ی ۱۹۹۸ آنها مشغول توسعه‌ی تکنولوژی جدید و نویدبخش این موتور جست و جو می‌شوند.

اتاق لری در خوابگاه دانشگاه، مرکز داده و اتاق برین به عنوان اداره کسب و کار، به کار برده می‌شود.

در این زمان آن‌ها متوجه می‌شوند گوگل در استنفورد بسیار شهرت یافته است؛ بنابراین تصمیم به فروش آن به altavista با مبلغ پیشنهادی یک میلیون دلار می‌کنند که پیشنهاد آنها رد می‌شود. آنها این پیشنهاد را با یاهو و Excite نیز در میان می‌گذارند که آنها نیز این پیشنهاد را رد می‌کنند.
پس آنها تصمیم به عملی کردن ایده‌ی خود می‌گیرند و به این منظور شروع به طراحی بیزینس پلن مناسب و یافتن سرمایه گذار می‌کنند.

دیوید کریتون، پروفسور علوم رایانه آنها، Andy Bechtol-sheim را به آنها معرفی می‌کند. او از این ایده به خوبی استقبال کرده و یک چک ۱۰۰ هزار دلاری به آنها می‌دهد. اواسط ۱۹۹۸ برین و پیج، با وجود مخالفت پدر و مادرشان، درس خود را در استنفورد متوقف می‌کنند.
اولین اداره‌ی گوگل یک گاراژ در حومه شهر Menlo park در کالیفرنیا بود که در سال ۱۹۹۸ توسط یکی از دوستانشان در اختیار آن‌ها قرار می‌گیرد. این اداره شامل کامپیوترهای رومیزی، یک میز پینگ پونگ و یک موکت به رنگ آبی روشن بوده است که صحنه‌ی آسمان اول صبح و اواخر شب را تداعی می‌کرد؛ این سنت به کارگیری اشیای رنگارنگ تا به امروز ادامه یافته است.
در اواخر سال ۱۹۹۸ گوگل حدود ۶۰ میلیون صفحه را ساپورت و روزانه حدود ۱۰۰ هزار درخواست را پردازش می‌کند و به همین دلیل در آن سال (۱۹۹۸) در مجله‌ای به عنوان یکی از ۱۰۰ وب سایت و موتور جست و جوی برتر سال انتخاب می‌گردد. اندکی بعد، "Craig Silverstein" به عنوان اولین کارمند در آن‌جا شروع به کار می‌کند.



سال بعد کمپانی به شهر پالو آلتو منتقل می‌شود. رفته رفته تعداد کاربرهای راضی از گوگل افزایش می‌یابد و نام آن درهمه جا شنیده می‌شود.
بنابراین، این کمپانی برای توسعه خود به سرمایه گذار احتیاج پیدا می‌کند.
در ژوئن ۱۹۹۹ آنها تصمیم می‌گیرند دو شرکت سرمایه گذاری رقیب یعنی Sequoia Capital وKleiner Perkins را برای سرمایه گذاری بر روی پروژه خود متقاعد کنند.

سرگی و لری مخالف استفاده از بنرهای تبلیغاتی بودند. آنها به طراحی تمیز و ساده خود افتخار می‌کردند چرا که هدف اصلی آنهاsearch بوده است نه تجارت. در حالی که در همان زمان شرکت‌های رقیب در حال تلاش برای فروش بنرهای تبلیغاتی هستند، سرگی و لری با استفاده از اسپانسرها مخالفند.
در اواخر ۱۹۹۹ گوگل حدود ۷ میلیون سرچ را در هر روز پردازش می‌کند و در سال ۲۰۰۰ این رقم به ۱۸میلیون جست و جو افزایش می‌یابد.
هزینه‌های عملیاتی تا ۵۰۰ هزار دلار در ماه بالا می‌رود و این آن‌ها را ناچار می‌کند که از تصمیم خود برای عدم استفاده از اسپانسرها تجدید نظر کنند.
آنها ایده‌ای مطرح می‌کنند که به جای نمایش تبلیغات در صفحه اصلی، لینک‌های هایلایت شده را در صفحاتی جداگانه ارائه کنند.

برای مثال اگر شخصی در جست و جوی خود اطلاعات بیشتری را درباره خودرو می‌خواهد، می‌تواند روی دکمه اطلاعات بیشتر کلیک کند که به اسپانسر مربوطه لینک داده شود.
بنابراین در اکتبر ۲۰۰۰، AdWords به عنوان سرویس تبلیغات آنلاین ارائه می‌شود.

در این زمان در حالی که کمپانی‌های رقیب، برای مثال یاهو، اکسایت، آلتا ویستا و لیکس، میلیون‌ها دلار برای تبلیغات و برندسازی هزینه می‌کنند، گوگل بی‌صدا، به‌سختی برای یافتن توقعات کاربران تلاش می‌کند.

سرگی معتقد بود در اصول مارکتینگ، گوگل باید به کاربران خود اتکا کند تا کاربرانی که از سرویس‌های آن استفاده کرده‌اند، آن را به دیگران پیشنهاد کنند.
در دسامبر ۱۹۹۹ وقتی "مارس میر" به عنوان کارمند دوم به تیم آنها اضافه می‌شود، با تغییر فونت متن صفحه‌ی نتیجه، برای ظاهر بهتر، با اعتراض شدید کاربران مواجه می‌شود و سرگی و لری ترجیح می‌دهند که صفحه‌ی اصلی و صفحه‌ی نتیجه آن‌ها تغییر نکرده باقی بماند.
اداره‌ی گوگل چندین بار تا سال ۲۰۰۳ به شهرهای مختلف نقل مکان می‌کند؛ تا این که در سال ۲۰۰۶ در Mountain view که محل فعلی اداره گوگل است مستقر می‌شود.
در سال ۲۰۰۴ گوگل با سهام NASDAQ (GOOG) با سهام‌هایی با ارزش ۸۵ دلار وارد IPO مارکت می‌شود؛ در اواخر روز ۱۹ میلیون سهام فروخته می‌شود و ارزش سهام به هم‌پیوسته تا نزدیک ۱۰۰ دلار افزایش می‌یابد.

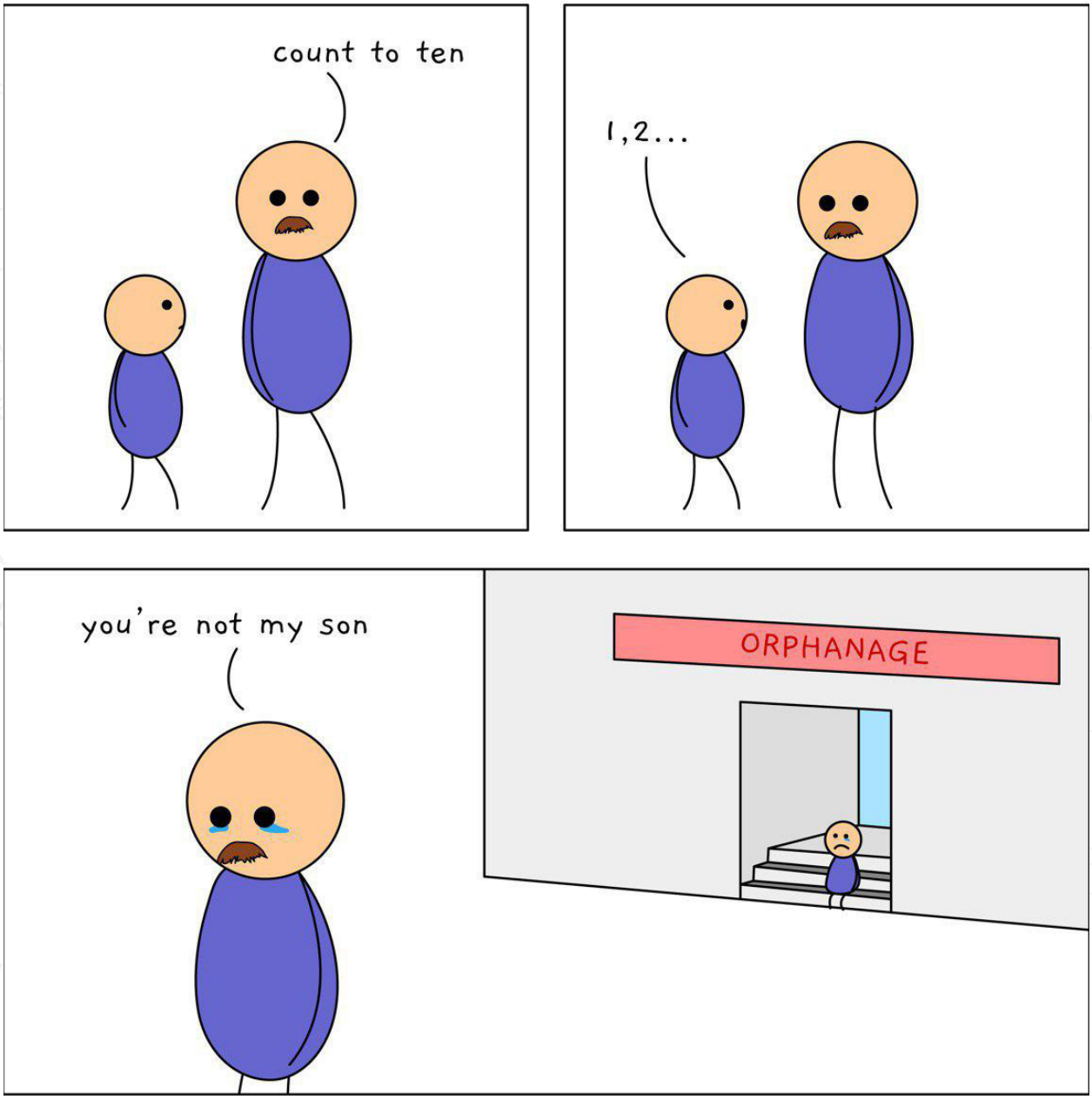
در آگوست ۲۰۰۵ ارزش هر سهام به نزدیکی ۳۰۰ دلار رسیده و در ۲۰۰۷ این مبلغ تا ۶۰۰ دلار افزایش می‌یابد.
بنابراین موجودی کمپانی تا ۳ بیلیون دلار افزایش می‌یابد و آن‌ها بیلیونر می‌شوند!

طی این سال‌ها این شرکت هر روز روند تکاملی را طی می‌کند، رویکردی که تا به امروز ادامه یافته است.
امروزه محصولات گوگل در حوزه‌های متعددی ارائه می‌شوند: ابزارهای اجتماعی، ابزارهای developer ها، ابزارهای امنیتی، ابزارهای مرتبط به نقشه، ابزارهای نموداری، سیستم عامل‌ها، برنامه‌های دسکتاپ، برنامه‌های موبایل، سخت افزار، خدمات و غیره.
بسیاری از ما، هر روز از بعضی محصولات گوگل استفاده می‌کنیم؛ شاید بتوان گفت زندگی دیجیتالی بدون سرویس‌های گوگل ناممکن به نظر می‌رسد.
حتی با صرف نظر از موتور جست و جوی گوگل، سیستم اندروید تلفن همراهتان، نقشه‌های گوگل، یوتیوب، مرورگرهای گوگل (مثل گوگل کروم)، Gmail و بسیاری از محصولات دیگر که همه روزه از آن‌ها استفاده می‌کنیم، از یک ملاقات دانشجویی آغاز شده است!

منابع:

APA Style
Harvard Style
MLA Style
about.google
Wikipedia
astrumpeople.com

Programmer dads be like



نویسنده: حامد محمدی

بیت‌کوین چیست و بلاک‌چین چگونه کار می‌کند؟

بیت کوین چیست؟

برای پاسخ به این سوال به نظر می‌رسد در ابتدا لازم است نگاهی کوتاه به تاریخچه پول بیاندازیم؛ همانطور که می‌دانید در ابتدا روش‌هایی مانند مبادله کالا با کالا و یا قرار دادن کالای خاصی به عنوان مرجع مبادلات در بین انسان‌ها مرسوم بود، به طوری که رفته رفته فلزات گران‌بهایی مانند طلا و نقره در بین همه ملیت‌ها به عنوان پول پذیرفته شدند، اما در برخی موارد برای معاملات تجاری سنگین حمل مقدار زیادی طلا و نقره توسط تاجران امکان پذیر نبود؛ بنابراین برخی از ارگان‌ها اقدام به ثبت برات‌هایی کردند که در حقیقت معادل مقدار مشخصی طلا بودند و به این ترتیب مفهوم اسکناس و پولِ با پشتوانه شکل گرفت؛ تا اینکه برخی دولت‌ها بنابر دلایلی اقدام به تولید پول‌های دستوری یا بی‌پشتوانه کردند که در حقیقت این پول‌ها دارای ارزش ذاتی نبودند و هیچ معادلی نداشتند، بلکه فقط دولت صادر کننده آن اسکناس ارزش آن را تضمین می‌کرد؛ در حقیقت ارزش این پول‌ها به میزان ارزش دولت صادر کننده آنها بود که مشکلاتی از قبیل تورم‌های اقتصادی و یا نوسانات نرخ ارز و تغییر ارزش یک پول نسبت به پول دیگر، در حقیقت ناشی از همین بی‌پشتوانه بودن پول‌های چاپ شده توسط دولت‌ها است. رفته رفته و با فراگیر شدن سیستم‌های رایانه‌ای عده‌ای از فعالان این حوزه در صدد ساخت ارزهای دیجیتال بر آمدند تا بتواند همه مشکلات دیگر ارزها را کنار زده و معاملات را در سطح جهانی ساده‌تر کنند؛ بنابراین تعدادی ارز دیجیتال تولید شد؛ اما همه این ارزها خود دارای مشکلاتی از قبیل سرعت انجام تراکنش، محدودیت در میزان تراکنش و از همه مهم‌تر، نیازمند وجود یک شخص ثالث برای احراز هویت تراکنش‌ها و جلوگیری از ایجاد تراکنش‌های جعلی بودند. به این ترتیب همه افراد موجود در آن شبکه پولی می‌بایست به آن شخص ثالث اعتماد کامل کرده و همه اطلاعات مالی خود را در اختیار آن فرد قرار دهند.

با وجود همه تلاش‌های شکست خورده قبلی در نهایت در سال ۲۰۰۹ فرد یا گروه برنامه نویسی‌ای با نام مستعار ساتوشی ناکاموتو توسط یک الگوریتم کامپیوتریِ انقلابی، موفق به حل مشکل تایید هویت تراکنش‌ها و جلوگیری از خرج کردن دوباره بدون نیاز به شخص ثالث شدند. این الگوریتم در حقیقت همان بلاک‌چین (BlockChain) و یا زنجیره بلوکی بود که در ادامه به بررسی آن و ارتباط آن با بیت کوین خواهیم پرداخت.

بلاک‌چین چگونه کار می‌کند؟

همانطور که از اسمش پیداست در حقیقت بلاک‌چین شامل زنجیره‌ای از بلاک‌های داده‌ای است که توسط توابع رمزنگاری یک طرفه مانند SHA256 به یکدیگر مرتبط شده‌اند و کنار

یکدیگر قرار گرفته‌اند تا توالی‌ای از دیتاهای به هم مرتبط را نگه‌داری کنند؛ در نتیجه با این تعریف می‌توان بلاک‌چین را یک پایگاه داده امن تصور کرد که هر فرد می‌تواند سطری داده به آن اضافه کند اما امکان حذف کردن یا تغییر دادن داده‌های قبلی به سادگی امکان پذیر نیست، چرا که تغییر دادن یک داده باعث تغییر مقداد هش(Hash) آن بلاک شده و به همین ترتیب لازم است تا انتهای زنجیره این تغییرات اعمال شوند؛ بنابراین تغییر یا حذف داده بسیار پرهزینه خواهد بود. باتوجه به این تعریف می‌توان بلاک‌چین را معادل دفترِ کل (ledger) در نظر گرفت؛ پس می‌توان گفت بلاک‌چین یک دفتر کل توزیع شده و مشترک است که کار فرآیند ثبت تراکنش‌ها و ردگیری دارایی‌ها را در یک شبکه کسب و کار ساده می‌کند.

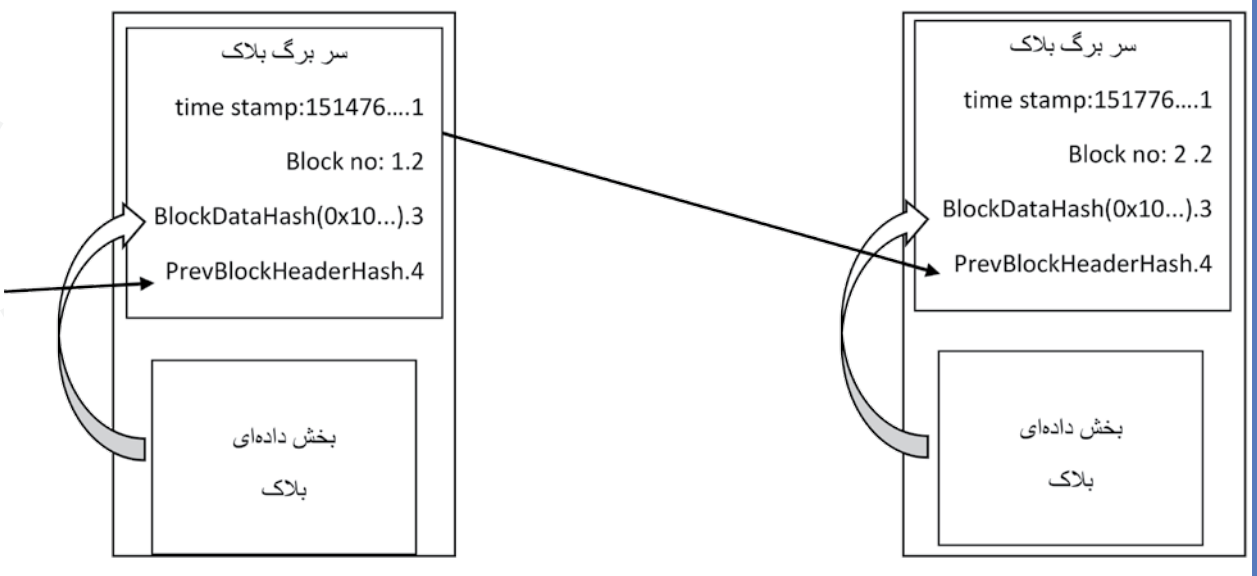
حال به بررسی دقیق‌تر بلاک‌چین و نحوه قرار گرفتن این بلاک‌های داده‌ای کنار هم و مرتبط کردن آنها با یکدیگر می‌پردازیم. همانطور که پیش‌تر گفته شد، هر بلاک در بلاک‌چین حاوی مقداری داده است که می‌تواند مرتبط با مقوله خاصی باشد و حجم و فرمت و قالب مشخصی نیز داشته باشد. در نهایت نیز در هر بلاک علاوه بر داده‌هایی که از آنها نگه‌داری می‌کند یک بخش سربرگ (header) وجود دارد که این سربرگ حداقل شامل موارد زیر می‌باشد:

۱. زمان تشکیل بلاک و یا time stamp که به طور معمول به صورت تعداد ثانیه از زمان unix در نظر گرفته می‌شود.
۲. شماره بلاک که بیانگر عدد بلاک در توالی بلاک‌ها می‌باشد.
۳. خروجی تابع رمزنگاری یک‌طرفه و یا به اصطلاح hash اطلاعات داده‌های بلاک
۴. اطلاعات سربرگ بلاک قبلی که در حقیقت این بخش باعث ایجاد زنجیره بلاکی می‌شود.

همانطور که مشاهده می‌شود، هر بلاک حاوی اطلاعات سربرگ بلاک قبل از خود است و از آن‌جا که بلاک قبل به بلاک قبلی مرتبط است، در نتیجه توالی ایجاد شده به صورتی خواهد بود که همواره آخرین بلاک حاوی اطلاعات کل بلاک‌های قبل از خود به صورت فشرده خواهد بود، به طوری که برای ایجاد تغییر در داده هر یک از بلاک‌های قبلی مجبور خواهیم بود به محاسبه مجدد سربرگ همه بلاک‌های بعد از آن نیز بپردازیم. این محاسبه مجدد با تعداد بلاک‌های بعدی رابطه توانی دارد و هرچقدر تعداد بلاک‌های بعد از آن بلاک مورد نظر برای تغییر بیشتر باشد، این محاسبات سخت‌تر و زمانبرتر خواهند بود. از طرفی همواره بلاک‌های جدیدی نیز در شبکه درحال ایجاد شدن هستند؛ بنابراین اگر یک فرد بخواهد اطلاعات یک بلاک را تغییر دهد نه تنها باید اطلاعات سربرگ همه بلاک‌های بعد از آن تا زمان حاضر را تغییر دهد، بلکه باید اطلاعات بلاک‌هایی که بعدا تولید می‌شوند را نیز تغییر دهد، که این بدان معنی است که فرد مورد نظر باید سیستمی کامپیوتری با قدرت محاسباتی بیشتر از تمام سیستم‌های دیگر در شبکه داشته باشد که چون خود آن فرد را نیز می‌توان جزئی از شبکه دانست در نتیجه فرد مورد نظر حداقل باید دارای ۵۱ درصد از قدرت محاسباتی کل شبکه باشد؛ که به طور مثال برای شبکه‌ای با عظمت bitcoin دست یافتن به چنین قدرت محاسباتی برای یک نفر بسیار سخت خواهد بود.

توابع رمزنگاری یک‌طرفه:

در بخش‌های قبلی در مورد استفاده از توابع رمزنگاری یک‌طرفه و یا Hash Functionها در شبکه بلاک‌چین صحبت کردیم. در اینجا به بررسی فنی این توابع می‌پردازیم. این توابع در حقیقت با انجام پردازش‌هایی روی هر رشته یکتای ورودی از اطلاعات در مبنای دودویی یک خروجی یکتا و با طول ثابت تولید می‌کنند؛ به طوری که از روی رشته خروجی به هیچ عنوان نمی‌توان رشته ورودی را به دست آورد و به همین دلیل



نیز به آن‌ها لفظ یک طرفه اطلاق می‌شود. از طرفی نیز این توابع تضمین می‌کنند که رشته تصادفی خروجی تولید شده به ازای هر رشته ورودی با آن hash function خاص ثابت و یکتا باشد؛ به طوری که با داشتن رشته ورودی و دانستن نوع hash function همواره بتوان رشته خروجی را بازیابی کرد و به ازای دو رشته ورودی متفاوت همواره دو رشته خروجی متفاوت تولید شوند که فارغ از طول رشته‌های ورودی رشته‌های خروجی همواره طول ثابتی نیز داشته باشند. از این رو این توابع را بعضا با نام تعداد بیت رشته خروجی‌ای که تولید می‌کنند نیز می‌شناسند مانند SHA256 و SHA512.

در بلاک‌چین از هر یک از این HashFunction ها می‌توان استفاده نمود. به طور مثال در سیستم بیت‌کوین از مکانیزمی موسوم به double SHA۲۵۶ استفاده شده است که به معنی دو بار پشت سر هم هش کردن با تابع SHA۲۵۶ می‌باشد.

خطای double spending و مکانیزم POW (Proof Of Work) :

یکی از خطاهای رایج در ارزهای دیجیتال خرج کردن دوباره (double spending) می‌باشد که این خطا ناشی از آن است که فردی که تعداد مشخصی ارز دیجیتال (مثلا بیت‌کوین) دارد، به طریقی این بیت‌کوین‌ها را به حساب ۲ نفر متفاوت واریز کند؛ یعنی یک بیت‌کوین را دوبار خرج کند. دو مورد از ساده‌ترین راهکارهای این خرج کردن دوباره آن است که به تغییر دادن اطلاعات بلاک‌های قبلی بپردازد و یا در مدت زمانی که خرج کردن بیت‌کوین‌ها بین همه افراد در شبکه پخش شود و همه اطلاعات مورد نظر را دریافت کنند، به خرج کردن دوباره آن‌ها بپردازد که مکانیزم POW به مقابله با هر دوی این راهکارها می‌پردازد.

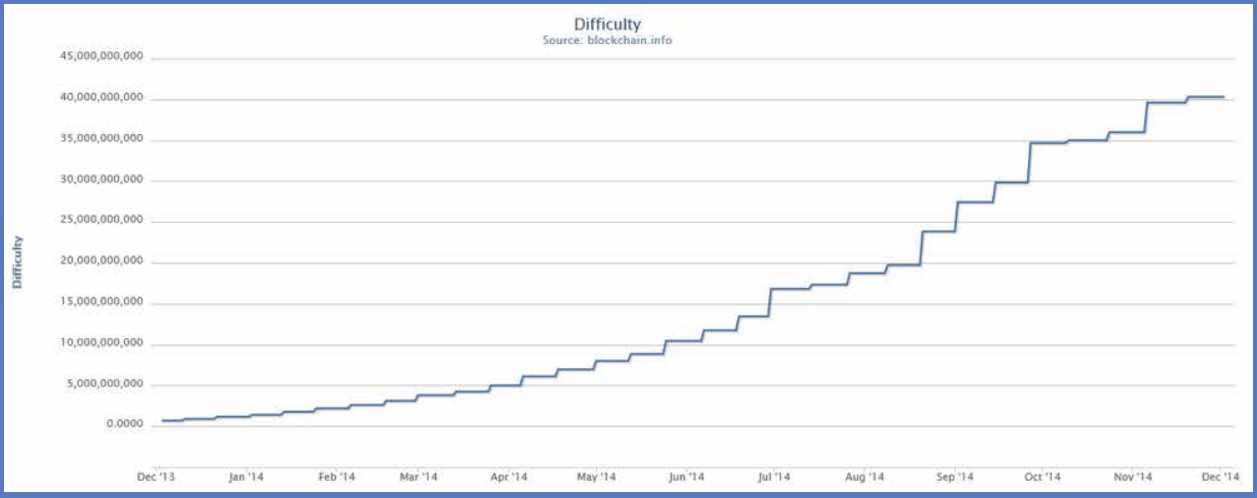
در بخش قبلی به سخت بودن تغییر دادن اطلاعات بلاک‌های قبلی اشاره شد، اما با وجود سیستم‌های کامپیوتری جدید، امنیتی که خود بلاک‌چین به تنهایی دارد کافی به نظر نمی‌رسد؛ چرا که همچنان هش کردن اطلاعات سربرگ یک بلاک چند میلی‌ثانیه طول می‌کشد و با وجود اینکه تعداد درخواست‌ها برای نوشتن اطلاعات روی بلاک‌ها محدود است، فرد خاطی می‌تواند در زمان مورد نیاز برای رسیدن اطلاعات یک بلاک به حد نصاب، اطلاعات بلاک مورد نظر خود را تغییر دهد و همه بلاک‌های بعدی را نیز مجددا با تغییرات لازم ایجاد نماید؛ بنابراین در شبکه بیت‌کوین مکانیزمی به اسم گواهی اثبات کار و یا proof of work وجود دارد که انجام این محاسبات را بسیار سخت‌تر می‌کند و همانطور که از اسمش پیداست، هر فردی که می‌خواهد اطلاعات یک بلاک جدید را به شبکه معرفی کند، مجبور به اثبات میزان معینی کار انجام شده است. با انجام دادن این کار مشخص که در حقیقت همان hash کردن است، امنیت شبکه نیز تضمین می‌شود.

در مکانیزم proof of work و یا به اختصار POW یک شرط برای خروجی توابع رمزنگاری یک‌طرفه که قبلا استفاده کردیم، تعریف می‌کنیم؛ شرط اینکه خروجی به دست آمده همواره باید ارزش عددی ای کمتر از یک عدد مشخص (که با نام difficulty یا سختی شناخته می‌شود) داشته باشد. به طور ساده‌تر می‌توان گفت که تعداد معینی از بیت‌های پر ارزش خروجی به دست آمده، باید ۰ باشند و بعد از آن می‌توانند ۰ یا ۱ باشند. هرچقدر که مقدار سختی بیش‌تر باشد، تعداد این ۰ ها بیش‌تر شده و از آنجا که نمی‌توان خروجی HashFunctionها را حدس زد، افراد مجبور می‌شوند به تولید خروجی‌های بیشتر بپردازند تا خروجی تولید شده شرط مورد نظر را داشته باشد؛ بنابراین به جای ۱ بار باید میلیون‌ها بار به hash کردن اطلاعات سربرگ بلاک بپردازند، اما همانطور که گفته شد به ازای هر رشته ورودی همواره یک رشته خروجی توسط hash function ها تولید می‌شود؛ پس چگونه می‌توان با وجود اطلاعات سربرگ بلاک که ثابت است، خروجی های مختلفی توسط hash function تولید کرد تا یکی از آن‌ها از درجه سختی کم‌تر باشد؟

در پاسخ به سوال فوق به معرفی متغییر nonce در سربرگ بلاک‌ها می‌پردازیم. به این صورت که به سربرگ‌هایی که قبلا گفته شد، مورد پنجمی به اسم nonce نیز اضافه می‌کنیم که می‌تواند هر مقدار تصادفی‌ای داشته باشد؛ بنابراین یکی از اطلاعات موجود در سربرگ بلاک قابل تغییر دادن است. حال هر بار با تغییر دادن این متغیر به hash کردن اطلاعات سربرگ بلاک می‌پردازیم تا در نهایت یکی از خروجی‌های تولید شده شرط مورد نظر ما را داشته باشد. از طرفی نیز قابل ذکر است مقدار متغییر nonce با تعداد ۰ های خروجی hash function از نظر ریاضی هیچ ارتباط تابعی ندارند؛ بنابراین نمی‌توان به حدس زدن مقدار nonce پرداخت و تنها راهکار موجود آزمون و خطای مقدار زیادی عدد برای این متغییر و hash کردن می‌باشد، تا به طور کاملا اتفاقی یکی از خروجی‌های تولید شده شرط مورد نظر را داشته باشد.

مورد دیگری که شایان ذکر به نظر می‌رسد، آن است که در بلاک‌چینِ بیت‌کوین مقدار درجه سختی به ازای هر تعداد مشخصی بلاک، با یک فرمول ریاضی مشخص تغییر می‌کند؛ به طوری که همواره در شبکه زمان میانگین پیدا شدن آن خروجی مشخص، معین باشد تا در این زمان معین اطلاعات کافی برای بلاک نیز تولید شوند. این فرمول به این صورت عمل می‌کند که به ازای هر تعداد بلاک مشخص، زمان کل ایجاد شدن آن بلاک‌ها به تعداد بلاک‌ها تقسیم شده و اگر از آن زمانِ میانگین بیش‌تر باشد، درجه سختی کم‌تر می‌شود و اگر بیش‌تر باشد، درجه سختی نیز بیش‌تر می‌شود. درجه سختی در بلاک‌چین بیت‌کوین از زمان ایجاد شدن آن تا امروز به طور نمایی افزایش یافته است. این پیش‌رفت ناشی از افزایش تعداد افرادی که به اصطلاح miner شناخته می‌شوند و هم‌چنین استفاده آن‌ها از متدهای بهتر و کارآمدتر است.

نمودار افزایش درجه سختی بیت‌کوین



استخراج و یا ماینینگ بیت‌کوین:

ماینینگ در حقیقت به انجام عملیات هش کردن متوالی اطلاعات یک بلاک با nonce های متفاوت تا به دست آوردن خروجی متناسب گفته می‌شود. این عمل هنگامی که توسط افراد صادق انجام بپذیرد، تغییر دادن اطلاعات بلاک‌ها را برای افراد غیر صادق سخت‌تر می‌کند، چرا که همانطور که گفته شد فرد خاطی باید به محاسبه خروجی hash function بلاکی که می‌خواهد تغییر دهد و همه بلاک‌های بعد از آن بپردازد، از طرفی حالا با وجود مکانیزم POW این عمل به طور چشم‌گیری سخت می‌شود؛ بنابراین امنیت شبکه بیت‌کوین فقط و فقط توسط الگوریتم‌های کامپیوتری و بدون نیاز به هیچ دولت یا مرکز قدرتی تامین می‌شود. این امنیت در حقیقت مرهون تلاش ماینرهای صادق می‌باشد؛ بنابراین تعدادی بیت کوین نیز به عنوان جایزه برای این ماینرها در نظر گرفته شده است، به طوری که هر ماینری که به ازای بلاک فعلی موفق شود زودتر خروجی hash function با شرط مورد نظر را پیدا کند، این جایزه را دریافت می‌کند که به block reward معروف است.

امروزه با افزایش تعداد زیاد ماینرهای بیت‌کوین و استفاده این افراد از ابزارهایی نظیر GPUها یا تراشه‌های FPGA و یا تراشه‌های مخصوصی که برای ماینینگ بیت‌کوین تولید شده‌اند، درجه سختی کار به طرز عجیبی افزایش یافته است؛ به طوری که افراد برای به دست آوردن این خروجی به تنهایی نمی‌توانند به رقابت با دیگران بپردازند؛ مگر اینکه عملیات mining را در farm های کامپیوتری بزرگ و اختصاصی انجام دهند. برای افراد دیگری که این farm ها را در اختیار ندارند، راهکاری به نام استخراج اشتراکی یا pool mining ایجاد شده است؛ افرادی که عضو یک pool هستند همه با هم همکاری می‌کنند و فردی که موفق به پیدا کردن خروجی مورد نظر شود همه جایزه را دریافت نمی‌کند، بلکه جایزه بدست آمده بین همه افرادی که در یک mine با هم همکاری می‌کنند تقسیم می‌شود؛ به این ترتیب همه افراد mine همواره مقداری

از این جایزه را دریافت می‌کنند. مقدار جایزه دریافتی نیز ارتباط مستقیم با قدرت سخت افزار و توان محاسباتی‌ای دارد که وارد شبکه کرده‌اند. (به طور دقیق‌تر تعداد خروجی‌های hash ای که سخت‌افزار آن‌ها در یک ثانیه توانایی محاسبه آن را دارد که به hash rate معروف است) در یک pool مقدار hash rate همه افراد حاضر در آن باهم جمع می‌شود و جایزه به دست آمده بین افراد بر اساس hash rate آنها تقسیم می‌گردد.

توجه: همان‌طور که ذکر شد مکانیزم ماینینگ ارتباط مستقیم با سخت‌افزار شما و توان محاسباتی آن دارد و برای ماین کردن شما نیازمند اتصال به شبکه اینترنت برای دریافت اطلاعات بلاک و ارسال خروجی مورد نظر خواهید بود؛ بنابراین تمامی کانال‌های تلگرامی و یا برنامه‌هایی که ادعا میکنند شما با تعداد معینی تپ کردن صفحه و یا کلیک کردن موس قادر به ماین کردن و به دست آوردن بیت کوین یا هر ارز دیجیتال دیگری هستید، صرفا جنبه سرکاری و کلاه برداری داشته و غیر واقعی هستند. امروزه برای ماین کردن با سود دهی، شما حداقل نیاز به ۴ کارت گرافیک قدرتمند و عضویت در یک pool mine بیت کوین مانند antpool و ... را دارید، از طرف دیگر بیت‌کوین یک پول است که به سختی توسط محاسبات کامپیوتری به دست میآید؛ بنابراین هیچ کانال تلگرامی با عضویت در کانال نمیتواند به شما بیت‌کوین پرداخت کرده و شما را پولدار کند؛ پس هشیار باشید و گول تبلیغات کلاه بردارها را نخورید.

تراکنش:

در شبکه‌هایی نظیر بیت‌کوین هر نقل و انتقال در قالب یک تراکنش ذخیره می‌شود. اطلاعات این تراکنش‌ها در حقیقت همان اطلاعات بلاک هستند که در بخش‌های قبلی ذکر شد. بلاک حاوی تعداد معینی تراکنش است. در بخش‌های فوق به توصیف کارکرد بلاک‌چین پرداختیم. حال به طور دقیق‌تر ماهیت این تراکنش‌ها را بررسی می‌کنیم.

هر تراکنش حاوی اطلاعات انتقال تعداد معینی بیت کوین از حساب فردی به حساب فرد دیگر می‌باشد، اما حساب هر فرد در شبکه بیت‌کوین چه معنایی دارد؟

حساب هر فرد در شبکه بیت‌کوین در حقیقت یک دوتایی از کلیدهای اختصاصی و اشتراکی می‌باشد. کلید اشتراکی به عنوان آدرس فرد شناخته می‌شود و برای واریز بیت‌کوین به حساب هر فردی از کلید اشتراکی‌اش استفاده می‌گردد. کلید اختصاصی هر فردی به طور امن فقط نزد آن فرد قرار دارد که به کمک آن می‌تواند تراکنش ارسالی خود را امضای دیجیتال نماید و به این صورت اثبات کند که مالک بیت‌کوین‌های موجود در آن تراکنش است؛ در حقیقت اثبات کند که قبلا فرد دیگری این بیت‌کوین‌ها را به حساب کلید اشتراکی آن فرد که دارای کلید اختصاصی مرتبط با آن است واریز نموده است.

مکانیزم گفته شده در فوق به رمزنگاری نامتقارن یا asymmetric cryptography معروف است که درآن به هر فرد ۲ کلید داده می‌شود؛ به طوری که اطلاعاتی که با کلید عمومی رمز شوند فقط توسط کلید خصوصی قابل بازگشایی هستند و اطلاعاتی که توسط کلید خصوصی رمز شوند فقط توسط کلید عمومی قابل بازگشایی خواهند بود. از اولین الگوریتم‌های رمزنگاری نامتقارن می‌توان به الگوریتم RSA اشاره کرد که بر پایه همنهشتی‌ها کار می‌کند، اما الگوریتم استفاده شده در شبکه بیت‌کوین به ECC یا elliptic curve cryptography مشهور است که توسط منحنی‌های بیضی شکل و الگوریتم‌های پیچیده ریاضی به رمز کردن اطلاعات می‌پردازد.

امضای دیجیتال:

امضای دیجیتال یک مکانیزم بر روی رمزنگاری نامتقارن است که در آن فرد امضا کننده ابتدا اطلاعات را با کلید اختصاصی خود رمز می‌کند و این رمز به دست آمده را به همراه یک رونوشت از خود اطلاعات به فرد دیگر ارسال می‌کند؛ فرد دریافت کننده با داشتن کلید عمومی فرد امضا کننده اطلاعات رمز شده را رمزگشایی می‌کند و برابری آن‌ها را با رونوشت ارسال شده توسط فرد امضا کننده می‌سنجد. به این ترتیب در صورت برابری می‌تواند مطمئن باشد که اطلاعات ارسالی قطعا توسط مالک آن کلید عمومی که از قبل شناخته شده است برای فرد دریافت کننده ارسال شده‌اند؛ بنابراین در امضای دیجیتال رمز کردن ومخفی کردن اطلاعات مورد بحث نیست؛ بلکه آنچه مطلوب است اثبات هویت ارسال کننده اطلاعات است. در شبکه بیت‌کوین نیز هر فرد اطلاعات ارسال بیت‌کوین‌های مورد نظر خود را توسط کلید خصوصی خود که در حقیقت حکم همان wallet شما را دارد امضا می‌کند و افراد دیگر موجود در شبکه با دانستن کلید عمومی آن فرد می‌توانند این امضا را اعتبارسنجی کنند و با مشاهده سابقه تراکنش‌ها، ارسال شدن آن بیت‌کوین‌ها را به آن کلید عمومی راستی آزمایی کنند.

When you delete a block of code
that you thought was useless

مصاحبه

مصاحبه با امیرراد کیمیایی
تهیه کنندگان: امیررضا یزدان‌پناه، غزال تاجیک

یکی از مهم‌ترین عوامل اختلاف سطح دانشگاه‌ها، محیط‌وجوعملی‌اون‌هاست که‌به‌واسطه‌دانشجوهای اون دانشگاه به وجود میاد. ارزشمندترین چیزی که یه دانشجو در دوران تحصیلش میتونه به دست بیاره تجربه ترم بالایی‌هاست که چند سال ازش جلوترن و چند پیرهن بیشتر پاره کردن. برای همین به سراغ امیر کیمیایی یکی از دانشجوهای موفق ورودی ۹۳ در زمینه هوش مصنوعی و بینایی ماشین رفتیم تا چند کلمه‌ای از تجربه‌هاش برامون بگه. در ادامه با ما همراه‌باشید.

سلام! اول خودتون رو معرفی کنید تا مخاطبان هم با شما آشنا بشن.

سلام. من امیرراد کیمیایی هستم، البته تو شناسنامه امیر ثبت شده ولی اکثرا امیرراد صدام می‌کنن. ورودی ۹۳ مهندسی نرم افزار خواجه نصیر بودم که آذرماه پارسال فارغ التحصیل شدم.

تابستون برای اکثر دانشجوها یه فرصت فوق العاده است که از وقت آزادشون استفاده کنن و مهارت‌های خودشون رو ارتقا بدن. شما به عنوان یه دانشجوی موفق تابستون‌ها به چه کارهایی مشغول بودین؟

لطف دارین ولی من خودم رو یه دانشجوی موفق نمی‌دونم! اگه اشتباه نکنم تابستون سال اول به زمینه‌های مربوط به کامپیوتر نپرداختم و مثل اکثر ورودی‌های جدید خیلی تو باغ نبودم. تابستون سال دوم برای اینکه با محیط‌های کاری آشنا بشم مشغول کارآموزی تو یه شرکت شدم که کارشون تولید اپلیکیشن برای اتوماتیک کردن نصب یک پکیج نرم افزاری بود و کارها بیشتر مربوط به فایل، اتوران ویندوز و ... بود. اما مهم‌ترین ویژگی اون کارآموزی نظمی بود که به من و زندگی دانشجوییم داد. سال بعدش هم باز کارآموزی رفتم اما یه کارآموزی کاملا متفاوت! باعث شد بتونم تشخیص بدم به چی علاقه دارم و از چی خوشم نمیاد. قبل از این کارآموزی علاقه من بیشتر سمت بازی سازی بود ولی با گذشت زمان و گذروندن این کارآموزی به سمت هوش مصنوعی، یادگیری ماشین و به طور اختصاصی بینایی ماشین رفتم و در ادامه در همین شرکت مشغول به کار شدم.

آیا «ای کاش»ای دارین؟ تاحالا پیش اومده با خودتون فکر کنید اگه فلان کارو می‌کردم بهتر بود؟

بله ای کاش‌ها که برای هر کسی وجود داره. اولین ای کاشی که به ذهنم میرسه کم شرکت کردن در مسابقات گروهی دانشجوییه. بهتر بود خودم رو با مسائل مسابقات ACM-ICPC بیشتر درگیر می‌کردم. چالش‌هایی که تو این سبک مسابقات طراحی میشه شاید تو کار تاثیری نداشته باشه، اما درگیر شدن باهاشون باعث تقویت قدرت تصمیم گیری، فکر کردن و حل مسئله میشه و به دنبالش دید بازتری نسبت به مسائل بهتون میده که این دید در آینده کاری به کار میاد.

یه حسرت دیگه‌ای که همراهم دارم و خیلی مهمه، نداشتن ارتباطات اجتماعی گسترده است. وقتی با یه دیدگاه منفی وارد هرجایی بشید باعث میشه که خودتون رو از خیلی چیزها محروم کنید. خوبه که بچه‌ها با این دید به دانشگاه بیان که خودشون رو با یک سری ادم دیگه که مثل خودش هستن ببینن. به این فکر کنن که دوست دارن یک سری کارها به صورت مشترک انجام بدن. خودشون رو با دانشگاه درگیر کنن تا قبل از خروج از دانشگاه یک سری مهارت‌ها مثل ارتباط برقرار کردن، کارگروهی و... رو یاد بگیرن. این حسرت من بوده که ای کاش بیشتر سر این موضوعات وقت می‌گذاشتم.

یکی از مهم‌ترین چالش‌ها برای دانشجوهای کامپیوتر انتخاب زمینه مورد علاقه‌شون بین این همه زمینه جذاب و متنوعه. توصیه شما چیه؟

این انتخاب کاملا تدریجی رخ میده. مهم‌ترین چیزی که به این فرآیند سرعت میده تجربه کردنه. زمینه‌های مختلف رو تا حد امکان تجربه کنید. من ترم پنج تازه هوش منصوعی و بینایی ماشین رو تجربه کردم. نکته خیلی جالب دیدگاه منفی من نسبت به هوش مصنوعی بود. اینکه باعث افزایش هوش ماشین‌ها بشم به نظرم تهدیدی برای انسان بود و به همین دلیل دیدگاه خوبی بهش نداشتم اما بعد از تجربه کردنش بهش علاقه‌مند شدم. تنها راه پیدا کردن علاقه تجربه کردن تا زمانیه که حس کنید این زمینه انتخاب نهاییه.

خیلی از افراد معتقد هستن که مهارت‌های کامپیوتری صرفا با خودآموزی و بدون نیاز به دانشگاه و تحصیلات آکادمیک قابل دستیابی هست. نظر شما راجع به این موارد چیه؟

متاسفانه اکثر افراد حتی مهندسین فعال دید منفی‌ای نسبت به دانشگاه دارن و دروس دانشگاهی رو اتلاف وقت میدونن. اما متاسفانه متوجه نیستن که این درس‌ها، تک تکشون پایه آینده کاریشون هستند. برای مثال در زمینه کاری من، بینایی ماشین، کاملا به دانش ریاضیات، آمار و احتمال و درس اختیاری جبر خطی وابسته است! البته این درس‌ها واقعا پایه خیلی از موضوعات هستند و توصیه می‌کنم که جدی بگیریدشون. کار خوبی که امسال انجمن علمی انجام داد برگزاری همایش‌های آشنایی با زیرشاخه‌های کامپیوتر بود که باعث شد ورودی‌ها تا حد خوبی با زمینه‌ها آشنا بشن و اهمیت درس‌ها براشون مشخص شد. جدا از این موارد بقیه درس‌ها به آدم دید میدن. برای مثال در درس شبکه نحوه کار TCP/IP رو یاد می‌گیریم ولی احتمالا اگه تو زمینه شبکه کار نکنید، مستقیم به کارتون نمیاد اما وقتی یه جایی با مشکلی مواجه بشیم می‌تونیم از اون اطلاعات قبلی استفاده کنیم و مشکل رو حل کنیم. در چنین موقعیت‌هایی فرق مهندس خوب و بد مشخص میشه.

فارغ از موضوعات درسی دانشگاه امکاناتی مثل آزمایشگاه روباتیک و کارگاه‌ها وجود داره. شما از این امکانات استفاده کردین؟ و نظرتون چیه؟

خب آزمایشگاه‌ها و کارگاه‌های خیلی متنوع و زیادی داخل دانشگاه وجود داره اما اکثرا مختص به برق هستن؛ ولی آزمایشگاه روباتیک برای کامپیوتری‌ها هست که متاسفانه زمینه مورد علاقه من نبوده برای همین فعالیتی نداشتم.

کارآموزی اختیاری این روزها خیلی بین دانشجوها رایج شده و در مقابل اون دیدگاه خودآموزی به جای کارآموزی وجود داره. نظر شما راجع به این دوتا موضوع چیه؟

کارآموزی باید مفید باشه! اما متاسفانه در کشور فرهنگ بدی هست که بعضی از شرکت‌ها به کارآموزها کارهای پیش پا افتاده‌ای رو محول میکنن و دیدگاهشون نسبت به کارآموزها دید برده داریه. پس خیلی خودتون رو برای ورود به بازار کار به آب و آتش نزنید. چون ممکنه به قیمت تلف شدن وقتتون با عنوان کارآموزی بشه. اما جدا از این مورد یک سری از مهارت‌ها الزامات رشته کامپیوتر هستن و به زمینه کاری خاصی مربوط نیست به نظرم بهتره قبل از کارآموزی حداقل این مهارت‌ها رو یاد بگیرید که تا حد امکان کارآموزی مفیدی داشته باشید. فقط در آخر یه مورد رو اضافه می‌کنم. توصیه اکید می‌کنم که بچه ها سعی کنن دنبال موضوعات غیر کامپیوتری نرن و وقتشون رو صرف موضوعات کامپیوتری کنن.

به مهارت‌های الزامی رشته کامپیوتر اشاره کردید. میشه یکم بیشتر توضیح بدین؟

بعضی از مهارت‌ها پایه‌ای هستند. مثل کار با shell یا bash که حتما نیازه دستورات و امکاناتش رو بدونید. مورد خیلی مهم بعدی بلد بودن GIT هست. بدونید دقیقا چیه، چجوری باید باهاش کار کرد و چه امکاناتی رو در اختیارتون قرار میده. در آخر حتما مفهوم و کارکرد make یا cmake رو یاد بگیرید. بعد از یادگیری این مهارت‌های پایه‌ای به مرور نیازهای تخصصی زمینه مورد علاقه‌تون رو پیدا می‌کنید و به مرور یاد می‌گیرید.

گفته بودین زمینه کاریتون بینایی کامپیوتره. امکانش هست کمی برامون توضیح بدین؟

بینایی کامپیوتر زمینه خیلی گسترده‌ایه که به طور کلی به دو بخش دو بعدی و سه بعدی تقسیم میشه. تو دانشگاه متاسفانه بخش سه بعدی پوشش داده نمیشه ولی بخش دو بعدی رو به تازگی دکتر نصیحت کن ارائه میدن. همونطوری که می‌دونید هر ویدئو استریمی از عکس‌هاست که ما بر حسب نیاز روی این عکس‌ها پردازش انجام می‌دیم. برای مثال شناسایی یه جسم خاص یا تشخیص حرکت جسم. در بخش سه بعدی میشه با کمک چند دوربین یک شیء رو به صورت ابر نقاط بازسازی کرد؛ که کاربردش تو سامانه‌های رانندگی(تشخیص موانع، تحلیل سرعت و ...) هست که بالاترین سطحش میشه ماشین‌های خودران. تو شرکت ما در سطح جلوگیری از برخورد و تشخیص تابلوهای راهنمایی رانندگی بهش پرداخته میشه و تمام اینا با استفاده از دوربین، بدون هیچ‌گونه سنسوری انجام میشه. علاوه بر اینا کاربردهایی تو روباتیک و پزشکی برای تشخیص سرطان داره.

وضعیت بینایی ماشین در ایران چطوره؟

متاسفانه به ویژن نسبت به بقیه زمینه‌های یادگیری ماشین تو ایران کم لطفی شده و دلیلش صنعتی بودن این حوزه

است که نیاز به سرمایه گذاری دولتی و هزینه‌های زیاد این موضوعه. حتی موضوعات کم هزینه‌تر مثل فیلترهای دوربین، مشابه اسنپ چت هم داخل کشور ندیدم. دلیلش می‌تونه دانش کم و جدید بودن این زمینه باشه. علاوه بر اینا اساتیدی که در صنعت کار کنن هم خیلی کم هستن ولی در کل بقیه زمینه‌های یادگیری ماشین وضعیت به مراتب بهتری دارن.

این روزها اکثر بچه‌ها دنبال یادگیری ماشین و بینایی کامپیوتر هستن. نظرتون چیه و پیشنهادتون برای موفقیت تو این زمینه‌ها چیه؟

تب یادگیری ماشین این روزها بین بچه‌ها افتاده و خیلی‌ها احساس می‌کنن اگه نرن سمت این زمینه از دنیا عقب میوفتن. به نظرم خودتون رو به این جو نبازید! اول مطمئن بشید که علاقه دارید بعد برید سراغش. اما اگه می‌خواید یه مقداری زودتر شروع کنید و وارد این زمینه بشید بهتره اول کدها و مدل‌های آماده که وجود داره رو دانلود کنید و باهاشون کار کنید تا به مرور طرز کارشون رو متوجه بشید. سعی کنید این یادگیری تدریجی باشه و چیزی رو رد نکنید؛ چون اکثر موضوعات اهمیت دارن و به دردتون میخوره. درس‌هایی مثل احتمال، معادلات دیفرانسیل و ... خیلی اهمیت دارن پس سعی کنید خوب یادشون بگیرید. مخصوصا احتمال رو تاکید می‌کنم که بیشتر از سطح دانشگاه یاد بگیرید. زبان‌های مورد استفاده تو اون زمینه رو بررسی کنید و به مرور یادشون بگیرید. اینترنت هم این روزها کار رو خیلی ساده کرده. کلی کورس و مطلب رایگان هست که می‌تونید ازشون استفاده کنید و یاد بگیرید. در آخر هم توصیه می‌کنم درس جبر خطی که اختیاری هست رو حتما بردارید.

برای ادامه تحصیل چه برنامه‌ای دارید؟ کمی از دورنمای آینده‌تون برامون بگید.

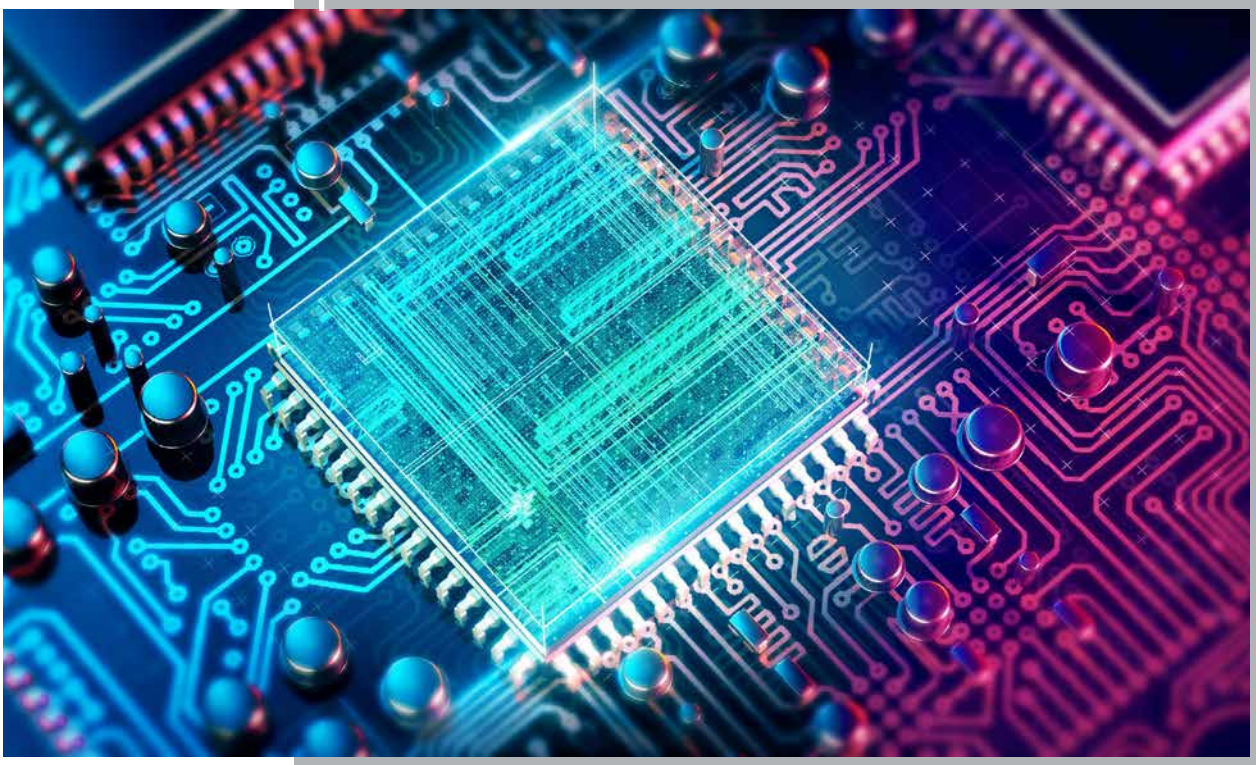
من تازگی برای دانشگاه ویرجینیا تِک آمریکا اپلای کردم. از اون‌جایی که من خیلی به ریسرچ علاقه داشتم و به نظرم خیلی موضوع مهمی بود، دوست دارم کسی باشم که بین صنعت و ریسرچ پل بزنم و تحقیقات رو در مقیاس صنعتی و کاربردی انجام بدم. به همین دلیل اپلای کردم تا ریسرچ‌های کاربردی بعدا در شرکت‌ها به کار بره.

حرفی، سخنی برامون دارید؟

به عنوان پیشنهاد دوستانه اگر قصد اپلای دارید، تو دانشگاه ریسرچ انجام بدید. درمورد موضوعی که علاقه دارید با استادها صحبت کنید و مشغول بشین. زبان موضوع خیلی مهمیه. حتی از موضوع کاراموزی هم به نظرم مهم‌تره حتما جدی بگیرید. تو پروژه‌های اوپن سورس هم فعالیت داشته باشین که در آینده رزومه خوبی براتون می‌سازه. به درسای دانشگاه هم اهمیت بدین. حواستون به ارتباط گرفتن با افراد جدید و کسایی که مثل خودتون دنبال پیشرفت هستن هم باشه و موفق باشید.

Programmers at Day

Programmers at Night



نویسنده: محمدمهدی محمودیان

پردازش کوانتومی، از رویا تا واقعیت

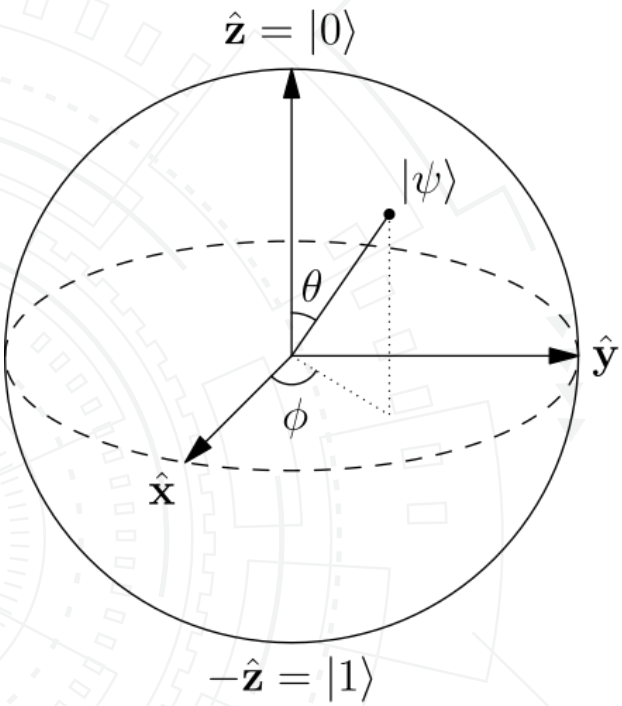
با شنیدن لغت کوانتم احتمالا ذهن‌ها به سمت فیزیک کوانتمی و اتم‌ها می‌رود و لغت پردازش، اگر دانشجوی کامپیوتر باشید، شما را یاد پردازنده ی کامپیوترها خواهد انداخت. اما این ترکیب عجیب چه معنایی دارد؟ پردازش‌های مربوط به حرکت اتم‌ها؟ یا پردازشی که در ابعاد اتم انجام می‌شود؟

پردازش کوانتمی یکی از زیرشاخه‌های علوم اطلاعات کوانتمی[۱] است. ایده پردازش‌های کوانتمی به حدود سال‌های ۱۹۸۰ برمی‌گردد؛ زمانی که ریچارد فیمن[۲] و یوری مانین[۳] ایده‌ی ساخت یک کامپیوتر کوانتمی را مطرح کردند. در این ایده کامپیوترهای کوانتمی توانایی شبیه سازی و انجام پردازش‌هایی را داشتند که کامپیوترهای معمولی قادر به انجام آن نبودند. این موضوع به شکل یک ایده باقی ماند تا روزی که پیتر شور[۴] الگوریتمی ارائه کرد که حل برخی از مسائل رمزنگاری که برای کامپیوترهای معمولی بسیار زمانبر و پیچیده بود، در ساختار کامپیوترهای کوانتمی به صورت بهینه‌تر و سریع‌تر قابل انجام شد. این الگوریتم پایه‌های توسعه‌ی کامپیوترهای کوانتمی را کلید زد.

۱. Quantum Information Science
۲. Richard Feynman
۳. Yuri Manin
۴. Peter Shor

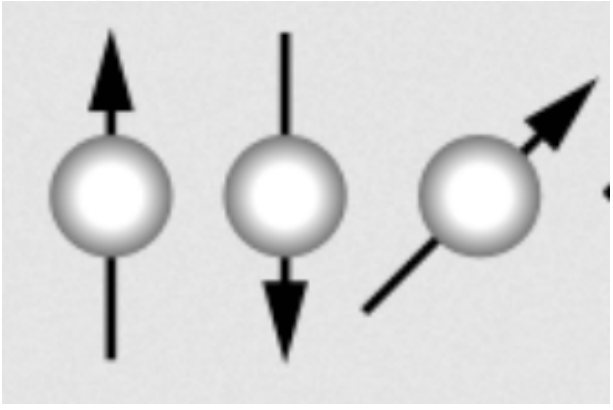
کامپیوتر کوانتومی چیست؟

کامپیوتر کوانتمی یا به عبارتی پردازنده کوانتمی، ساختاری مشابه پردازنده‌های عادی، اما با تفاوت‌های بنیادین دارد. در پردازنده کوانتمی به جای بیت از کیوبیت (Q-bit) استفاده می‌شود. مقدار کیوبیت، سوپر پوزیشنی از صفر و یک است. سوپر پوزیشن به معنای یک حالت بینابینی است. برای مثال، فرض کنید سکه ای را به هوا انداخته اید. در هر لحظه که این سکه در هوا در حال چرخیدن است، یا شیر است یا خط؛ اما مشخص نیست که دقیقا کدام حالت رخ خواهد داد. به چنین حالتی سوپرپوزیشن گفته می‌شود؛ به عبارت دیگر در کیوبیت مقدار ذخیره شده، صفر، ۱ یا حالتی بین این دو خواهد بود. بیایید باز به مثال سکه برگردیم. فرض کنید ما با استفاده از دستگاهی سکه را به هوا پرتاب کرده باشیم که زاویه، نیروی پرتاب و... قابل اندازی گیری باشد. اگر تمام پارامترهای این پرتاب را بررسی کنیم، می‌توانیم احتمال شیر یا خط آمدن را محاسبه کنیم. در کیوبیت نیز با انجام فرآیندی مشابه احتمال صفر یا یک شدن نهایی کیوبیت قابل محاسبه خواهد بود. قدرت پردازشی متفاوت پردازنده‌های کوانتمی از وجود سوپر پوزیشن‌ها و احتمالات سرچشمه می‌گیرد.



ساختار فیزیکی کیوبیت‌ها

همانطور که می‌دانید پردازنده‌های عادی از ترانزیستورها استفاده می‌کنند.
اگر ترانزیستور جریان را عبور دهد آن را یک در نظر می‌گیرند و اگر جریان را عبور ندهد به آن مقدار صفر اطلاق می‌گردد. اما پیاده سازی ماهیتی که صفر، یک یا مقداری بین این دو را ذخیره می‌کند موضوع متفاوتی است. برای ساخت یک کیوبیت به طور معمول از اتم‌های مصنوعی استفاده می‌شود. اتمی که مشابه هیدروژن تنها یک پروتون و الکترون دارد. جهت چرخش و زاویه فعلی الکترون نسبت به هسته بیانگر مقدار ذخیره شده در کیوبیت خواهد بود؛ اگر جهت محورِ چرخش الکترون رو به بالا باشد، آن را یک و اگر جهت محور چرخش رو به پایین باشد، آن را صفر در نظر خواهیم گرفت. حال اگر محور چرخش را ۴۵ درجه بچرخوانیم در نیمی از مواقع رو به بالا و نیم دیگر رو به پایین خواهد بود.



چگونه یک کامپیوتر کوانتومی عمل می‌کند؟

تا اینجا با ساختار یک پردازنده کوانتمی آشنا شدیم؛ اما این سوپر پوزیشن چگونه به تسریع پردازش‌ها کمک می‌کند؟ فرض کنید دو پردازنده داشته باشیم که یکی ۳ بیت و دیگری ۳ کیوبیت داشته باشد. در پردازنده اول هر کدام از بیت‌ها می‌توانند دو مقدار بگیرند و در مجموع ۸ حالت را می‌سازد که مقدار ذخیره شده یکی از حالات ۰۰۰، ۰۰۱، ۰۱۰، ۰۱۱، ۱۰۰، ۱۰۱، ۱۱۰، ۱۱۱ خواهد بود. شاید بگویید خب در سه کیوبیت نیز همین ۸ حالت رخ خواهد داد اما در اینجا تفاوتی عمیق وجود دارد. کل حالاتی که ۳ کیوبیت می‌تواند داشته باشد، همان ۸ حالتی است که پیش‌تر مطرح شد، اما مقداری که می‌تواند به خود بگیرد سوپرپوزیشنی از این ۸ حالت خواهد بود؛ یعنی به عبارت دیگر حالتی که ۳ کیوبیت در خود ذخیره خواهند کرد به احتمال مشخصی بین این حالت‌ها تقسیم خواهد شد. مقدار ذخیره شده همواره در حال جا به جایی بین این ۸ حالت است. به عبارت دیگر ۳ کیوبیت همه این ۸ حالت را در آن واحد پوشش خواهد داد. به این ترتیب اگر n بیت و n کیوبیت داشته باشیم اولی از بین 2^n حالت تنها یک حالت را به خود می‌گیرد و دومی همزمان 2^n حالت مختلف خواهد بود. این ویژگی باعث توسعه الگوریتم‌های مخصوص برای این ساختار شده است که می‌تواند دنیای پردازش آینده را به کلی عوض کند.

با اینکه حالت سوپرپوزیشنی در حال تغییر بین حالات مختلف است، اما نیاز است که در یک لحظه خاص مقدار فعلی مشخص شود. مثلا در حال حل یک مسئله هستیم و ناگهان حالت خاصی به جواب مطلوب منجر می‌شود. اکنون نیاز داریم تا مقدار آن حالت خاص را بدست آوریم. برای چنین حالاتی با کمک ابزارها و الگوریتم‌های مشخصی می‌توان به نوعی از حالت فعلی کیوبیت‌ها عکس گرفت و کیوبیت را باثبات کرد و سپس مقدار آن لحظه را محاسبه نمود. البته انجام چنین کاری اصطلاحا باعث از بین رفتن کیوبیت می‌شود؛ چرا که وقتی کیوبیت ثابت شود، هیچ تفاوتی با بیت معمولی نخواهد داشت و دوباره باید کیوبیت را تنظیم کرد.

برای روشن تر شدن، یک مسئله ساده را بررسی می‌کنیم. فرض کنید می‌خواهیم یک رمز ۳ رقمی را پیدا کنیم. در حالت عادی باید اعداد ۱۰۰ تا ۹۹۹ را تک به تک تست کنیم. برای اینکار به ۱۰ بیت نیاز خواهیم داشت که تا عدد ۱۰۰۰ را پوشش دهد. سپس از ۱۰۰ شروع می‌کنیم و تک تک مقدار این ۱۰ بیت را اضافه می‌کنیم تا رمز عبور پیدا شود و وارد سامانه شویم. اما با استفاده از ۱۰ کیوبیت اینکار می‌تواند به شکلی خارق العاده سریع‌تر انجام شود. وقتی که ۱۰ کیوبیت داشته باشیم مقدار ذخیره شده همواره درحال تغییر بین ۱ تا ۱۰۲۴ خواهد بود و هر عدد در هر ثانیه بارها به دست می آید و دوباره تغییر می‌کند. حال اگر مقدار ذخیره شده در

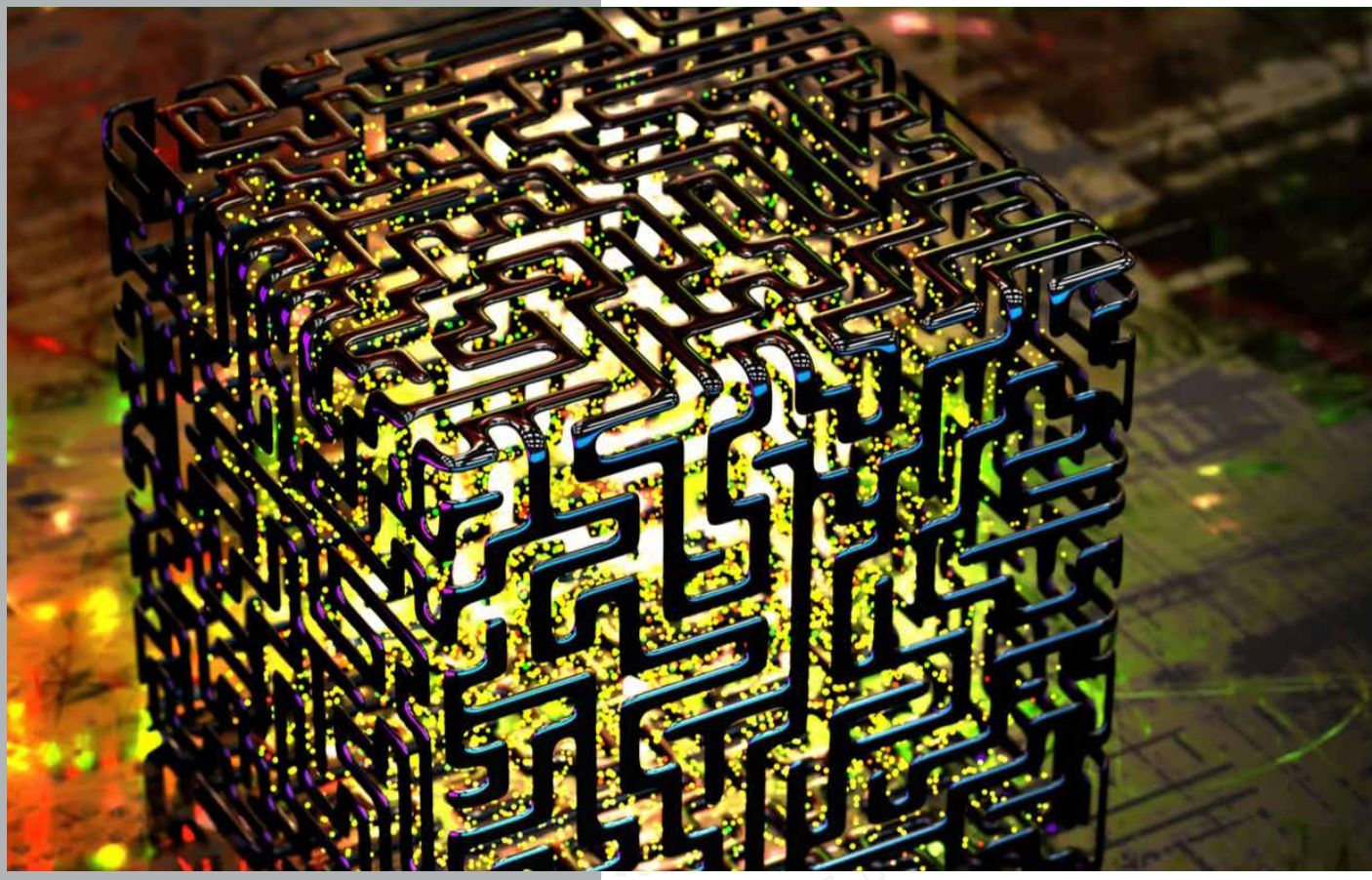
این ۱۰ کیوبیت را به عنوان رمز وارد کنیم، در لحظه وارد سامانه می‌شویم و رمز پیدا می‌شود؛ چرا که تمامی اعداد در یک لحظه به دست می‌آید و رمز عبور قطعا یکی از این اعداد خواهد بود. حال با کمک برخی الگوریتم‌ها و روش‌ها می‌توان رمز دقیق را نیز محاسبه کرد ولی تا همینجای کار به راحتی توانستیم در زمانی بسیار کمتر وارد سیستم شویم.

با این تفاسیر رمزگشایی و رمزنگاری‌های فعلی به شدت در معرض خطر خواهند بود؛ چرا که این مسائل که دارای پیچیدگی‌های زمانی بسیار بالایی بوده‌اند، به وسیله کوانتم کامپیوترها خیلی سریع‌تر انجام خواهند شد. البته دانشمندان و متخصصان به دنبال روش های رمزگذاری جدید هستند که اصطلاحا quantum secure یا ایمن در برابر پردازنده‌های کوانتمی خواهند بود.

اما با وجود تمام این برتری‌ها، توسعه‌ی پردازنده‌های کوانتمی خیلی سریع نیست. شرایط فیزیکی (از جمله دمای بسیار پایین و خنک نگه داشتن، ساخت کیوبیت‌ها و...) و نویز یا error rate به وجود آمده در ابعاد بزرگ‌تر، توسعه‌ی این سبک پردازنده‌ها را بسیار کند کرده است.

در حال حاضر طبق گفته های شرکت IBM، هر ۸ ماه تعداد کیوبیت‌ها و یا به عبارت دیگر قدرت پردازش دوبرابر می‌شود. شرکت های IBM، Intel، گوگل و مایکروسافت رقابت تنگاتنگی برای توسعه سریع‌تر کوانتم کامپیوترها دارند. طبق آخرین اخبار منتشر شده، قدرتمندترین پردازنده‌ی کوانتمی ساخته شده تاکنون، دارای ۵ هزار کیوبیت و متعلق به شرکت D-Wave Systems است که منحصرا بر روی کوانتم کامپیوترها سرمایه گزاری کرده است و در حال مطالعه و توسعه آنهاست.

باید دید که آینده‌ی پردازش کوانتمی به کدام سو خواهد رفت و کدام شرکت موفق خواهد شد مسائل دنیای واقعی را به وسیله این ساختار حل کند. علاوه بر حل مسائل، این روزها جاه طلبی انسان بسیار فراتر رفته و تحقیقاتی درباره پیاده‌سازی هوش مصنوعی برروی کوانتم کامپیوترها درحال انجام است. با توجه به محدودیت سایز ترانزیستورها و تقریبا به انتها رسیدن توسعه پردازنده‌های ترانزیستوری، سرمایه گذاری و تحقیقات در این حوزه‌ها بسیار مورد توجه قرار گرفته است.



تی‌اِی شدن یا نشدَن، مسئله این است!

نویسندگان: محمدامین پرچمی، حسین ریماز

تی‌ای یا همون Teaching Assistant یکی از فعالیت‌های دانشجویی هست که مثل بقیه فعالیت‌های مشابهش، هیچ اجباری در انجامش نیست و شرط اصلیش علاقست و یکسری خوبی‌ها و بدی‌ها داره. یکی از خوبی‌هاش اینه که یه فعالیت جهانی توی تقریبا همه‌ی دانشگاه‌های دنیاست. خیلی‌ها به همین علت سابقه‌ی تی‌ای شدنشون رو داخل رزومشون قرار میدن تا به قوی بودن رزومشون کمک کنه. ولی خب، نمیشه تی‌ای بودن رو به " برای رزومت خوبه" خلاصه کرد. حالا مزیت‌ها و معایب رو در آخر این متن به صورت جمع‌بندی می‌گیم که بهتر درک کنید.

تی‌ای چیکار می‌کنه؟

همه‌ی تی‌ای‌های دنیا یک وظیفه دارند و اون هم کمک به استاد برای برگزار شدن بهتر درسه. در هر مورد این کمک می‌تونه متفاوت باشه. اصلی‌ترینش حداقل توی دانشگاه خواجه نصیر، برگزاری کلاس‌های حل تمرینه؛ البته به همین راحتی‌ها نیست. طرح سوال و نمره‌دهی تکالیف و پیدا کردن تقلب، مراقب بودن سر جلسه، آپدیت کردن سایت درس، طرح و تحویل پروژه‌های درس و در نهایت پاسخگویی و کمک به دانشجویانی که اون درس رو برداشتند، از کارهایی هستند که تی‌ای باید اگر ازش خواسته شد، بدون چون و چرا انجام بده و حتی خودش پیش‌قدم بشه.

ولی خب این که شد فقط زحمت! پس لذتش کجاست؟

خب در واقع لذتش در حین انجام همین کارهاست؛ مثلا همین کلاس حل تمرین، اگر واقعا به تدریس علاقه داشته باشید، می‌تونه بهترین زمان روز و چه بسا اون هفته رو براتون رقم بزنه.

معمولا بچه‌ها تی‌ای درسی میشن که با استادش به خوبی کنار میان و به خود درسش علاقه دارند. شما برای این که بتونید به سوالات جواب بدید یا کلاس‌های حل تمرین رو به خوبی برگزار کنید، باید حتما قبل از هر جلسه حداقل کمی مطالعه داشته باشید، مطالعه‌ی درسی که دوست دارید!

خیلی از درس‌ها چند تا تی‌ای دارن و همه‌ی این کارها وقتی گروهی با تی‌ای‌های دیگه انجام بشه خیلی جذاب تره؛ مثلا از لذت‌هایی که شاید کمی شیطانی به نظر برسه، اینه که با سایر تی‌ای‌ها هم‌فکری کنید که چجوری تکلیف رو سخت طرح کنید یا پروژه‌ای بدید که بچه‌ها مجبور شن وقت بیشتری برای یادگیری بذارند.

چیزی هم یاد می‌گیریم؟

اول از همه اگر این موضوع واقعا براتون مهمه بهتون تبریک می‌گم.

در جواب باید بگم بله!

اگر قرار باشه برای یک کلاس حل تمرین آماده بشید، باید خودتون رو برای سوال‌هایی که ممکنه ازتون پرسیده بشه هم آماده کنید. از سوال‌هایی که ممکنه خیلی اساسی و پایه‌ای باشند تا سوال‌های سختی که تاحالا به ذهن شما خطور

نکرده؛ برای همین دانشتون در اون درس تا حدی وسیع‌تر و عمیق‌تر میشه. البته چیزهای دیگه‌ای هم غیر از مسائل مرتبط با اون درس یاد می‌گیرید؛ مثل مدیریت زمان. شما در کنار دانشجو بودن و تمام تکالیف و ددلاین‌هایی که باید رعایت کنید، یکسری ددلاین‌های تی‌ای بودن هم دارید و فرق ددلاین‌های تی‌ای اینه که بدترین حالتشون این نیست که نمرشون رو نمی‌گیرید، بلکه شاید باعث بشید یک عده درس رو یاد نگیرن و شاید حتی عده‌ای رو از درس زده کنید. یکسری مهارت‌های جانبی هم هستند که شاید کوچیک به نظر برسن ولی می‌تونن به رشدتون کمک کنن؛ مثلا یاد می‌گیرید با دانشجویانی که تی‌ایشون هستید جدی برخورد کنید، حتی اگر دوست صمیمی باشید. یاد می‌گیرید چجوری روش‌هایی رو انتخاب کنید که عدالت حفظ بشه و در بسیاری از موارد یاد می‌گیرید چجوری کنترل خودتون رو حفظ کنید. یک سری مهارت‌های مربوط به تدریس مثل بیان خوب مطالب، مشارکت مخاطبین و آماده کردن مطالب آموزشی رو هم یاد می‌گیرید

چجوری تی‌ای بشیم؟

بطور کلی شرایط لازم در این که بتونید اعتماد استاد درس رو جلب کنید خلاصه میشن. اولین چیزی که به ذهن می‌رسه این هستش که نمره‌ی خوبی از اون درس (ترجیحا با همون استاد) داشته باشید. دقت کنید، لزومی نداره که درس رو به تازگی گذرونده باشید؛ همون‌طور که خیلی وقت‌ها دانشجویان دکتری و ارشد، تی‌ای‌های دروس کارشناسی میشن.

البته صرفا یک نمره‌ی خوب نمی‌تونه کافی باشه. معمولا برای یک سری درس‌ها، متقاضی‌های تی‌ای شدن بیش‌تر از ظرفیت هستن و کمی رقابت ایجاد میشه که اگر بتونید ثابت کنید احساس مسئولیت خوبی دارید و واقعا می‌خواید تی‌ای بشید و وقت بذارید، معمولا تی‌ای می‌شید.

البته ذکر این نکته مهمه که برای اعلام آمادگی برای تی‌ای شدن باید حداقل چند ماه قبل از شروع ترم با استاد اون درس صحبت کنید. بعضی درس‌ها هستن که استاد ثابتی اون‌ها رو ارائه میدن که می‌تونید حتی برای اطمینان زودتر با اون استاد صحبت کنید؛ اما قطعا اگر یک هفته مونده به شروع ترم بخواید برای اینکار اقدام کنید و اون درس و یا استاد مخاطب زیادی داشته باشه، احتمالا شانس چندان زیادی ندارید؛ پس باید زودتر به فکر باشید.

از مزایای جانبی تی‌ای بودن؟

اگر خواستید استادی بیشتر شمارو بشناسه (برای مثال برای برداشتن پروژه، گرفتن توصیه‌نامه و سایر فعالیت‌های علمی‌ای که براش به یک استاد مناسب خودتون احتیاج دارید) می‌تونید با تی‌ای شدن، خودتون رو تا حدی به اون استاد اثبات کنید. اگر هدف جدی برای اپلای دارید قطعا به توصیه‌نامه اساتید احتیاج خواهید داشت. گذراندن یک درس با یک استاد و گرفتن نمره بیست از اون درس، معمولا برای گرفتن یک توصیه‌نامه خوب از اون استاد کافی نیست. ترجیحا کار پژوهشی یا دستیار آموزشی اون استاد بودن هم لازمه. چرا که با صرف گذراندن یک درس، اون استاد یا توصیه‌نامه چندان خوبی برای شما نخواهد نوشت یا این که تعداد کمی توصیه‌نامه برای شما پر می‌کند.

از طرفی مواردی بودن که به صورت رندم و از سر ناچاری و بخاطر گرفتن همین توصیه‌نامه، دستیار آموزشی درسی شدن. این می‌تونه تصمیم چندان درستی نباشه. اگر از سر ناچاری دستیار درسی شدین و به اون علاقه‌ای ندارین، احتمالا اون کلاس حل‌تمرین هم کابوسی برای شما میشه و هم تجربه بسیار بد برای دانشجویان. در نهایت هم استاد با فهمیدن کیفیت پایین باز هم توصیه‌نامه چندان خوبی برای شما نخواهد نوشت و احتمالا در ترم‌های آینده هم رغبتی برای هم‌کاری مجدد با شما نخواهد بود.

به عنوان یک توصیه به جای اینکه دستیار آموزشی چندین درس و چندین استاد باشید، بهتره دستیار آموزشی دروس محدود و اساتید محدودتری بشید. به عنوان مثال اگر چهار ترم متوالی تی‌ای یک استاد باشید، رابطه خیلی صمیمانه‌تری با اون استاد خواهید داشت. علاوه بر اینکه باعث گرفتن یه توصیه‌نامه خوب برای شما میشه، فرصت‌های دیگه‌ای مثل کارهای پژوهشی در آینده هم در اختیارتون خواهد بود. ولی اگر یک بار تی‌ای یک استاد باشید، احتمالا بعد چهار پنج سال حتی اسم شما هم در خاطر اون استاد نخواهد موند.

پول هم میدن؟

داخل دانشگاه ما من از هرکسی که پرسیدم گفت مبلغ چندانی نیست و ارزش پیگیری‌هاش رو نداره، پیشنهاد می‌کنم اگر بخاطر پولش می‌خواید تی‌ای بشید، کمی بیش‌تر فکر کنید.

برای درسم مشکلی پیش نمیاد؟

همون‌طور که گفتیم یکی از مزیت‌های تی‌ای بودن اینه که یاد می‌گیرید زمان خودتون رو مدیریت کنید. زمانی که ازتون می‌گیره معمولا به اندازه‌ای هستش که بتونید وقت مناسبی بهش اختصاص بدید.

بدترین قسمت تی‌ای بودن؟

بدون شک بدترین قسمت تی‌ای بودن قسمت نمره دادن هست. معمولا بیش‌ترین تنش‌ها و برخوردها در این قسمت روی میده؛ چرا که طبیعتا توقع فرد مقابل شما گرفتن نمره کامل هست، حتی اگر زحمتی برای پروژه‌اش نکشیده باشه. از طرفی توقع استاد از شما رعایت عدالت و داشتن معیارهای درست و دادن نمرات یک دست و دقیقه. یعنی چی؟ خیلی وقتا دانشجو در برگه و امتحان از مجموع بیست نمره حتی موفق به کسب ۱ نمره هم نمیشه، اما نمره حل‌تمرینش کامل و با هزاران مثبت شده! این به این معنیه که یا اون دانشجو دوست حل‌تمرین بوده که حتی اگر استاد هم چیزی به شما نگه، متوجه میشه و برای اعتبار شما بد میشه یا در حالت دیگه همه نمرات شما به این شکل بوده و خیلی مهربانانه به همه نمره کامل دادین. در این صورت استاد قطعا به شما تذکر خواهد داد و در بهترین حالت احتمالا تاثیر نمره حل‌تمرین در نمره پایانی بسیار کم خواهد شد. این یعنی کسانی که برای حل‌تمرین تلاش کردند و برای اون نمره حساب و کتاب کرده بودند، تلاششون از بین میره. به علاوه این که اعتبار شما هم از بین رفته.

تصمیم سخت‌تر اینه که اون تنش‌ها رو قبول کنید. فحش‌هایی که قراره پشت سرتون داده بشه رو به تن بخرید و به هرکس به قدر زحمتی که کشیده، بدون هیچ ارفاقی نمره بدین. حتی اگر دانشجو مقابل شما بخاطر نیم نمره حل تمرین قرار باشه از دانشگاه اخراج بشه یا دوست صمیمیتون بخاطر نمره حل‌تمرین از شما ناراحت بشه.

برای رزومه خوبه؟

بستگی به کیفیت کار شما و شیوه تعاملتون با استاد داره.

آیا برای کار پیدا کردن به دردتون می‌خوره؟ به شکل مستقیم طبیعتا نه. برای کار پیدا کردن نه مدرک دانشگاهی شما مهمه، نه مدرک‌های موسسه‌های دیگه‌ای که زحمت کشیدین و گرفتین و نه داشتن مقاله و دستیار آموزشی بودن. تنها چیزی که نیاز دارید، مهارت و دانش عمیق برای اون کاره. البته موارد فوق می‌تونن به شکل غیر مستقیم منجر به همون مهارت و دانش عمیق بشن، ولی نه لزوما.

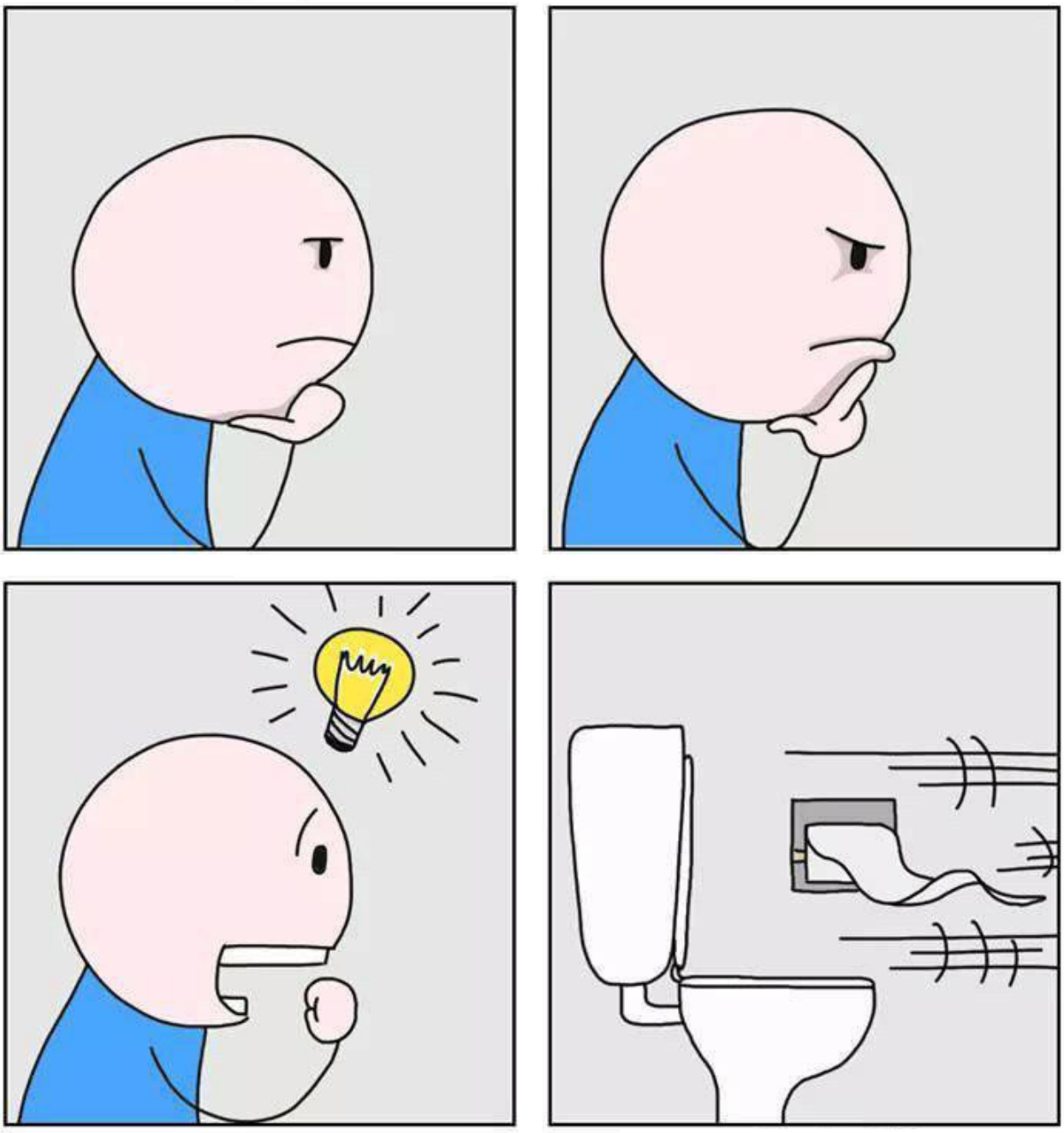
برای اپلای چی؟ باز هم به شکل مستقیم نه. البته اپلای یه بسته‌ی کلیه و معیارهای زیادی درش دخیله. همون‌طور که گفتیم برای اپلای نیاز به توصیه‌نامه‌ی اساتید دارید و یه توصیه‌نامه‌ی خوب می‌تونه خیلی کمک کنه و دستیار آموزشی شدن یکی از راه‌های گرفتن یه توصیه‌نامه خوبه. معمولا دوستان اساتید و همکارانشون، اساتید جاهای دیگه هم هستن؛ چه در داخل چه در خارج از کشور. بنابراین داشتن یک توصیه‌نامه خوب می‌تونه فرصت‌های خوبی براتون فراهم کنه. معمولا اساتید مطرح و شناخته شده که صد البته گرفتن توصیه‌نامه ازشون کار بسیار سخت و رقابتی‌ای هست، می‌تونن همچین فرصت‌هایی برای شما فراهم کنن.

اما در نهایت معدل و زبان حرف اول رو توی اپلای می‌زنن. اینکه تی‌ای شدن به معدلتون لطمه بزنه، بستگی به خودتون و برنامه‌ریزیتون و شرایط اون کلاس حل‌تمرین داره. اگر برای اپلای کاری خواستید بکنید که امکان لطمه به معدلتون رو داره، مثل تی‌ای شدن، مقاله دادن، دوره‌های مختلف آنلاین و غیر آنلاین گذروندن و ... به احتمال خیلی زیاد تصمیم درستی نباشه.

در انتها، تی‌ای بودن یکی از فعالیت‌های دوران دانشجویتون خواهد بود که همیشه یادتون می‌مونه؛ اما می‌تونید ازش هم به خوبی و هم به بدی یاد کنید و همش برمی‌گرده به علتی که روز اول تی‌ای شدید. اگر هدفتون از تی‌ای شدن گرفتن پذیرش توسط یک دانشگاه خارجی یا کمک به آینده‌ی شغلیتونه، فعالیت‌های بهتری هستند که می‌تونن راحت‌تر و با تاثیر بیشتری کمکتون کنن و شاید اگر تی‌ای بشید در انتها احساس کنید که صرفا وقت خودتون رو تلف کردید؛

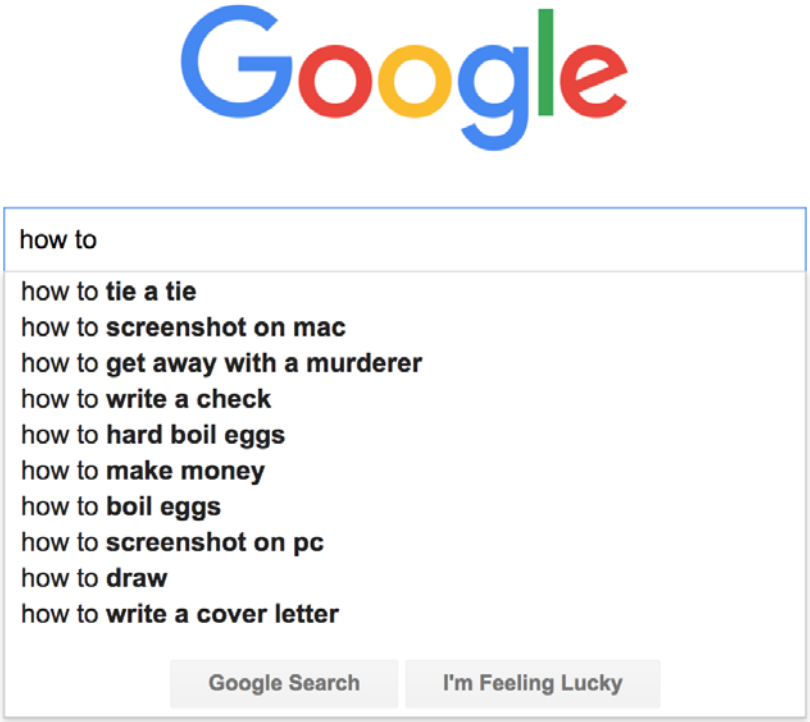
ولی اگر می‌خواید در کنار لذت تدریس و تمام چیزهایی که یاد می‌گیرید، چند خط هم به رزومتون اضافه بشه، تی‌ای شدن می‌تونه براتون مناسب باشه.

اتاق فکر



مقاله

Trie Tree



نویسندگان: محمدمهدی خدابنده، کیوان دهقان

آیا تا به حال از قابلیت autocomplete گوشی هوشمندتان استفاده کرده‌اید؟ مثلا وقتی حروفی مثل hel را تایپ می‌کنید و صفحه کلید گوشی شما کلماتی مثل hello یا help را به شما پیشنهاد می‌دهد، یا حتی وقتی کلمه‌ای مثل hepl را تایپ می‌کنید و صفحه کلید یک کلمه درست و مشابه این کلمه مثل help را برایتان نمایش می‌دهد. صفحه کلید گوشی یک برنامه‌ی کامپیوتری است که احتمالا در حال جستجو در یک لغت‌نامه برای حروف مرتبط است؛ اما اینکه چگونه این کار را می‌کند موضوع این نوشته است.

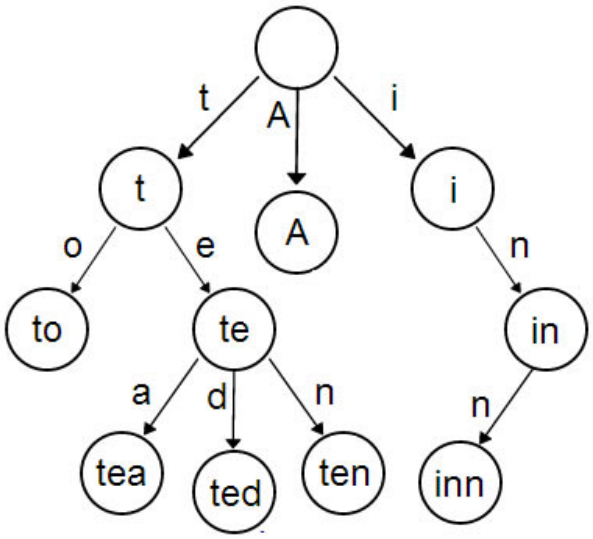
در حال حاضر فقط می‌خواهیم کلماتی را پیدا کنیم که حروفی را که ما تایپ کرده‌ایم به عنوان پیشوند داشته باشند. اگر شما حروف compl را تایپ کرده باشید احتمالا منظورتان کلمه ای مثل simple نبوده و در حال تایپ complete یا complex یا complement بوده‌اید؛ اما چطور می‌شود این حروف را از داخل یک لیست بلند بالا از کلمات پیدا کرد و به کاربر پیشنهاد داد؟ راه‌های مختلف با پیچیدگی‌های زمانی-حافظه‌ای متفاوت وجود دارد. برای مثال می‌توانیم به ازای همه‌ی کلمات داخل لغت‌نامه چک کنیم که آیا حروف ما پیشوندی از این کلمه هست یا نه؛ سپس از بین همه‌ی کلماتی که در مرحله‌ی قبل پیدا کرده‌ایم آن‌هایی که بیش‌ترین استفاده را دارند، به کاربر پیشنهاد کنیم. این کار کلمه‌های مناسب را به شما نشان می‌دهد ولی برای هر کلمه‌ای که می‌نویسید، تمام کلمه‌های لغت‌نامه را یک بار مرور می‌کند؛ یعنی به طور مثال اگر در حال نوشتن پیامی به دوستتان باشید که ۲۰ کلمه دارد، ۲۰ بار کل لغت‌نامه را جستجو می‌کند. حال فرض کنید که می‌خواهید یک کتاب بنویسید!

می‌توان حدس زد که این روش گشتن برای کلمه اصلا کاربردی نیست و قطعا گوشی‌های همراه شما از این روش استفاده نمی‌کنند. حال فرض کنید که کلمات داخل لغت‌نامه بر اساس حروف الفبا مرتب شده‌اند؛ می‌توانیم در هر مرحله کلمه وسط را برداریم و اگر از کلمه مورد نظر ما بزرگتر بود در سمت چپ آن جستجو کنیم و در غیر این صورت در سمت راست آن، یا به عبارت دیگر از روشی به نام جستجوی دودویی استفاده کنیم. این روش تعداد عملیات‌های انجام شده برای پیدا کردن کلمه‌های پیشنهادی را نسبت به روش قبلی تا حد چشم‌گیری کاهش می‌دهد؛ اما هنوز هم الگوریتم ما به تعداد کلمه های لغت نامه وابسته است. پس بیایید روش دیگری پیدا کنیم.

شما در حال جستجوی کلمه hello هستید. فرض کنید در یک میدان ایستاده‌اید که ۲۶ خیابان از آن خارج می‌شود. روی هر خیابان یک حرف از حروف زبان انگلیسی نوشته شده است. خیابان اول حرف a، خیابان دوم حرف b و همینطور تا خیابان ۲۶ ام که حرف z روی آن نوشته شده است. به شما

گفته شده برای پیدا کردن کلماتی که با a شروع می‌شوند باید به خیابان اول بروید (خیابانی که حرف a روی آن نوشته شده)، برای حروف دیگر نیز همین گونه عمل شود. شما برای پیدا کردن hello به ناچار وارد خیابان h می‌شوید چون شانسی برای پیدا کردن hello در خیابان‌های دیگر ندارید. به یک میدان دیگر می‌رسید و متوجه می‌شوید باز هم ۲۶ خیابان از این میدان خارج شده‌اند که روی هر کدام به ترتیب حروف ha و hb و ... hz نوشته شده است. متوجه می‌شوید که خیابان hz بن‌بست است چون هیچ کلمه‌ای با hz شروع نمی‌شود. اکنون برای پیدا کردن hello به کدام خیابان می‌روید؟ احتمالا خیابان he. بسیار خب؛ شما کلمه‌ی hello را با استفاده از این خیابان‌ها پیدا کردید. متوجه می‌شوید که بعد از پیدا کردن hello اصلا خسته نشده‌اید! چرا که فقط ۵ خیابان را طی کرده‌اید.

اگر بخواهیم این ساختار میدان و خیابان را داخل کامپیوتر پیاده کنیم، یک گراف گزینه‌ی بسیار خوبی است. این گراف دور نخواهد داشت چرا که دو کلمه‌ای که پیشوند یکسان دارند مسیر یکسانی را تا پیشوندشان طی می‌کنند و از آن به بعد مسیرشان روی گراف کاملا جدا می‌شود. پس این گراف یک درخت است که از یکسری راس و یال تشکیل شده است. هر راس حد اکثر ۲۶ فرزند دارد و هر یال یک پدر را به فرزندانش متصل می‌کند. این ۲۶ اشاره‌گر، نمادی از حروف الفبای انگلیسی هستند. برای هر حرف یک یال جداگانه در نظر می‌گیریم. در این درخت کلمات (یا به عبارت دقیق‌تر رشته‌های حروف) از بالا به پایین طبق پیشوندشان ذخیره می‌شوند. همه‌ی پیشوندهای به طول ۱ در سطح ۱ ذخیره می‌شوند. همه‌ی پیشوندهای به طول ۲ در سطح ۲ ذخیره می‌شوند و الی آخر. ریشه‌ی این درخت شامل هیچ کلمه‌ای نیست. روی این درخت مسیری که از ریشه طی کرده‌ایم پیشوند کلمه مورد نظرمان را مشخص می‌کند. درخت زیر یک مثال از پیاده سازی این مطلب است.



به این ساختار درختی، trie یا درخت prefix گفته می‌شود. برای جستجوی یک کلمه در trie اگر طول کلمه را n در نظر بگیریم می‌توان با پیچیدگی زمانی O (n) این کلمه را در درخت پیدا کرد.

از جمله کاربردهای این درخت می‌توان به autocomplete، spell checking، longest prefix matching، درصد استفاده لغت در مکالمات روزمره و کاربردهای بسیار دیگری اشاره کرد.

در ادامه، شبه کد برای دوعملیات درج و بررسی ساده ذکر شده است:

```
void insert(String s){for(every char in string s)
{if(child node belonging to current char is null)
{child node=new Node();}
current_node=child_node;}
boolean check(String s){for(every char in String s)
{if(child node is null){return false;} return true;}
```