

صفر

ZERONE

یک

نشریه انجمن علمی مهندسی
کامپیوتر دانشگاه ارومیه
شماره یک
بهار ۱۴۰۳



بسم الله الرحمن الرحيم

ترم نامه انجمن علمی مهندسی کامپیوتر دانشگاه ارومیه

شماره یک ، بهار ۱۴۰۳



صاحب امتیاز: انجمن علمی مهندسی کامپیوتر دانشگاه ارومیه

مدیر مسئول: میرسامان تاجبخش

سردبیر: هما کاوند

هیئت تحریریه: هومن امیدیان، امید پیشرو، علیسا حسین‌آبادی، سارا خسروزاده، حدیثه خراسانی، آریا رافع، شبینم رنجبری، معین رضازاده، میلاد رضوانی، اکرم سلطانی، رضا شیخلمزاری و پردیس مدد

صفحه آرایی و گرافیک: سارا خسروزاده و پردیس مدد

ویراستار: سارا خسروزاده، محمد مهدی فرجبخش و پردیس مدد

طراح جلد: سارا خسروزاده و پردیس مدد

فهرست مطالب

- ۰ ۲ بازی دوز
اکرم سلطانی
- ۰ ۶ دنیای سینمایی مارول
هومن امیدیان
- ۱ ۰ گراف سریال فرار از زندان
حذیثه خراسانی
- ۱ ۴ بررسی آسیب پذیری SQL Injection
پر迪س مدد
- ۲ ۰ PFsense
میلاد رضوانی
- ۲ ۴ OpenVpn
پر迪س مدد و علیسا حسینآبادی
- ۳ ۰ توسعه نرم افزار موفق در پلتفرم های آنلاین
رضا شیخلمزاری
- ۳ ۶ Edit Distance
سارا خسروزاده و معین رضازاده
- ۴ ۸ بخش ۱
حافظه ها
- ۴ ۸ آریا رافع
بخش ۱
- ۵ ۴ Byte-sized experiments
امید پیشرو
- ۶ ۴ بخش ۱
سفری به باغ وحش گوگل!
حذیثه خراسانی و شبنم رنجبری
- ۶ ۴ بخش ۱

سخن آغازین

میرسامان تاجبخش

عضو هیأت علمی مهندسی کامپیوتر دانشگاه ارومیه

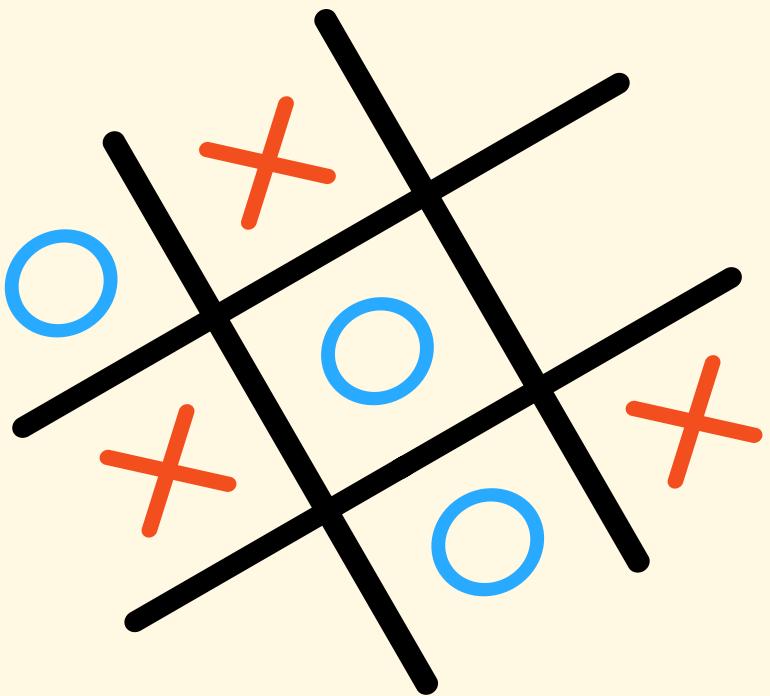
بسمه تعالی

سپاس خدای را که نعمت همکاری با این نشریه و دانشجویان مشتاق و فعال رشته کامپیوتر را به بندۀ ارزانی کرد. در این سال‌هایی که با دانشجویان رشته کامپیوتر بوده‌ام، همیشه شور و هیجانی در راستای فعالیت‌های علمی در دانشجویان را دیده‌ام که در صورت شکوفا شدن، منجر به فعالیت‌های بسیار بزرگی شده است. این فعالیت‌ها از مسابقات علمی ملی تا برگزاری کنفرانس و مسابقات بین‌المللی بوده است. این نشریه نیز به همت همین عزیزان منتشر می‌شود. این نشریه نتیجه فعالیت یک ترم دانشجویان رشته مهندسی کامپیوتر دانشگاه ارومیه می‌باشد. هر فعالیتی که انجام می‌شود نتیجه سلسله فعالیت‌های جزئی‌تر است که در نهایت به عنوان یک دستآورد به نمایش گذاشته می‌شود. این نشریه نیز از تهیه محتوای علمی، ویراستاری و طراحی تشکیل شده که جا دارد از تک تک عزیزانی که در این حوزه‌ها بوده‌اند تقدیر شود. مطالب ارائه شده در نشریه شامل مطالب و دستاوردهای دانشجویان کامپیوتر است که در قالب آموزش و مقالات علمی منتشر می‌شود. از این نظر مطالب ارائه شده مورد استفاده علاقه‌مندان حوزه کامپیوتر و همچنین متخصصان این حوزه می‌باشد.

در پایان بایستی از شما که در هر زمانی این مطلب را می‌خوانید تشکر کرده و آرزوی سلامتی و موفقیت برایتان داشته باشم.

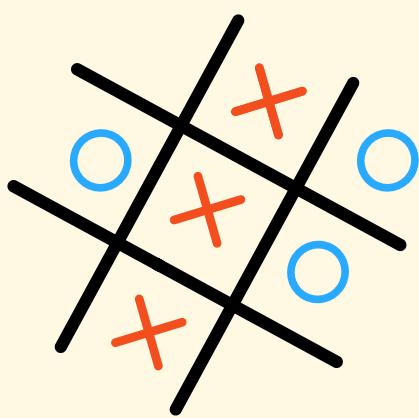


ms.tajbakhsh@urmia.ac.ir



Tic Tac Toe

نویسنده: اکرم سلطانی

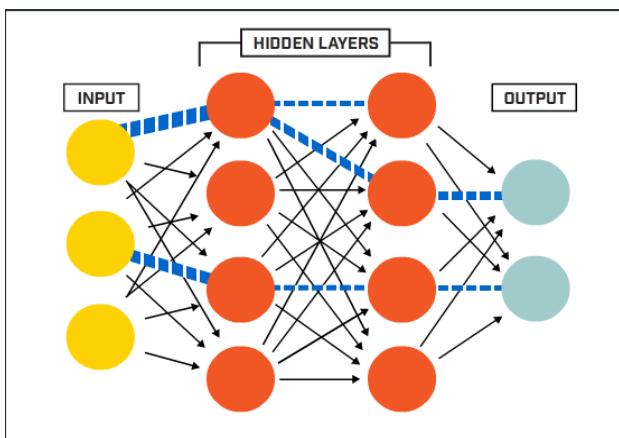




آموزش شبکه عصبی

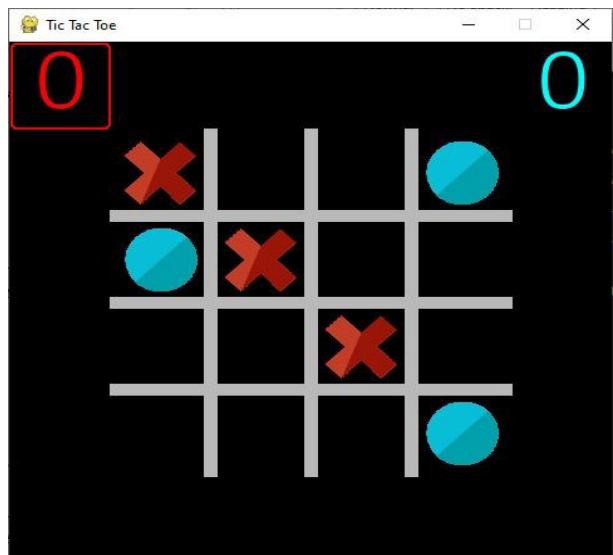
در هنگام آموزش بخش ویژوال برنامه غیرفعال می‌شود تا سرعت کار افزایش یابد. الگوریتم آموزشی با استفاده از توصیه‌های DQN Deep Mind و با استفاده از تکرار تجربه، پیاده سازی شده است. دسته‌ای از تجربیات تصادفی تکراری برای هر دور تمرین استفاده می‌شود. یک شبکه عصبی ثانویه برای محاسبه مقادیر Q آینده استفاده شده و سپس با وزن شبکه اصلی در هر ۱۰ دور بازی، به روز رسانی می‌گردد. محاسبه مقدار تابع Q بصورت تقریبی یا با معادله بلمن می‌باشد. یعنی شبکه آموزش می‌بیند تا مقدار هر حرکت ممکن در هر وضعیت بازی را پیش‌بینی کند. نام فایل کد اصلی وضعیت بازی را پیش‌بینی کند. نام فایل کد اصلی OwnNNNet_TicTacToe.py است و از کتابخانه ML استفاده می‌کند.

کتابخانه ML بصورت اولیه با هدف عمومی ساخته شده است و متشکل از چندین لایه و چندین نرون پرسپترون در هر لایه می‌باشد. این شبکه عصبی، وزن‌ها را در یک فایل ذخیره و بارگذاری می‌کند، forward propagation با انجام SGD، با استفاده از forward propagation و آموزش رو به جلو، وزن‌ها و بایاس‌ها را بهینه سازی کرده و خودش را برای انجام بازی دوز آموزش می‌دهد.

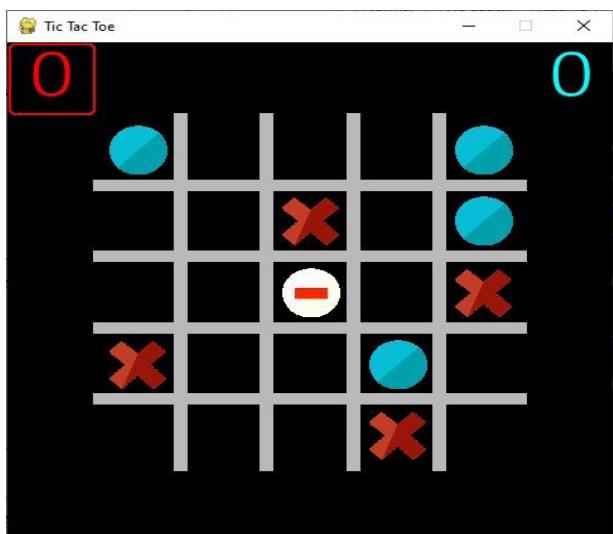


بازی به صورت شبیه سازی شده انجام می‌گیرد و خروجی‌ها در یک حافظه بازپخشی ذخیره

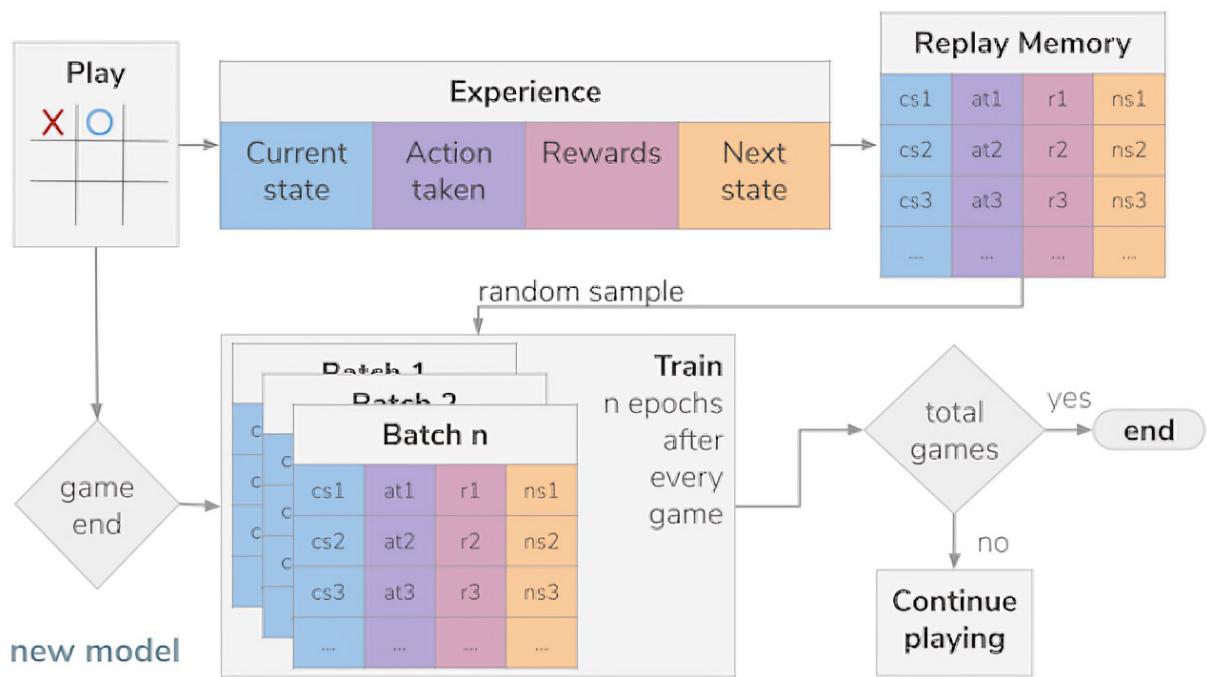
در این بخش یک بازی دوز n در n با پایتون طراحی شده است. یکی از بازیکنان انسان و حریف مقابلش هوش مصنوعی (یک شبکه عصبی آموزش‌دهنده با الگوریتم بازی‌های دو نفره) می‌باشد.



تعداد سطر و ستون یعنی n، در ابتدای بازی از کاربر دریافت می‌شود تا یک صفحه بازی با ابعاد سفارشی ایجاد و نمایش داده شود. برای ایجاد چالش بیشتر، در n های فرد خانه‌ی وسط صفحه‌ی بازی بصورت مسدود در نظر گرفته شده است و هیچ یک از بازیکنان نمی‌توانند خانه مذکور را انتخاب نمایند.



می‌شوند. وقتی تجربه کافی گردآوری شد (حداقل به اندازه batch size) شبکه با استفاده از نمونه‌های تصادفی از تجربه‌های برگرفته از حافظه بازپخشی و به تعداد epoch تعریف شده، آموزش می‌بیند. مثلاً اگر تعداد epoch‌ها برابر با ۱۰ انتخاب شده باشد و batch size را مساوی ۸۱ در نظر گرفته باشیم، شبکه ۱۰ بار و هر بار با ۸۱ تجربه رندم از حافظه بازپخشی آموزش داده خواهد شد. سپس بازی ادامه پیدا می‌کند و راندهای آموزشی تکرار می‌شوند.



تصویر زیر محاسبه می‌شوند:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2$$

m_t و v_t به ترتیب، تخمین لحظه اول (میانگین) و لحظه دوم (واریانس بدون مرکز) از گرادیان‌ها هستند، و نام روش هم از همین گرفته شده است. از آنجایی که m_t و v_t به صورت بردارهای ۰ مقداردهی اولیه می‌شوند، مشاهده می‌شود که آنها به سمت صفر سوگیری می‌کنند، به خصوص در مراحل اولیه زمانی، و به ویژه زمانی که نرخ‌های فروپاشی کوچک هستند (یعنی β_1 و β_2 نزدیک به ۱ هستند).

با محاسبه تخمین لحظه اول و دوم که با ایاسشان تصحیح شده:

روش بهینه سازی ADAM

برای بهینه سازی شبکه در طی backpropagation از چندین الگوریتم تکانه‌ای (Momentum) از جمله Adaptive Moment ADAM استفاده شده است. Adam Estimation (Adam) یک روش بهینه‌سازی است که نرخ یادگیری تطبیقی را برای هر وزن و بایاس محاسبه می‌کند. علاوه بر ذخیره سازی میانگین کاهشی نمایی از مجدد گرادیان‌های قبلی v_t ، و همچنین میانگین کاهشی شبیه قبلی m_t ، شبیه به تکانه عمل می‌کند. در حالی که می‌توان تکانه را به صورت سُر خوردن توپی در یک شبیه مشاهده کرد، Adam مانند یک توپ سنگین با اصطکاک رفتار می‌کند و بنابراین مینیمم‌های مسطح در سطح خطای را ترجیح می‌دهد. میانگین کاهشی قبلی و میانگین کاهشی مجدد گرادیان‌های گذشته

Numpy برای کار با ماتریس‌ها و آرایه‌های چندبعدی Json برای مدیریت فایل‌های json و تبدیل دیکشنری به جیسون.

Random برای انتخاب نمونه‌های تصادفی از بین تجربه‌های بازی ذخیره شده در اندازه‌ی batch تکرار و یادگیری بازی توسط شبکه عصبی.

Time برای محاسبه مدت زمان اجرای فرایند یادگیری برای تعداد مشخصی از Epoch‌ها.



در این مطلب به طراحی و پیاده‌سازی بازی دوز n در n با استفاده از شبکه عصبی مصنوعی و الگوریتم‌های یادگیری تقویتی پرداخته شده است. شبکه عصبی با استفاده از الگوریتم‌های یادگیری عمیق و بهینه‌سازی مانند DQN و Adam آموزش می‌بیند و از تکنیک‌هایی مانند تجربه بازپخشی برای بهبود کارایی استفاده می‌کند.

برای مشاهده و دسترسی به کدهای این قسمت، می‌توانید کد QR زیر را اسکن کنید یا بر روی لینک کلیک کنید:



<https://github.com/UUCESSC/TicTacToe>

$$\hat{m}_t = m_t / (1 - \beta_1^t)$$

$$\hat{v}_t = v_t / (1 - \beta_2^t)$$

سپس با استفاده از این مقادیر، وزن‌ها و بایاس‌ها را به روزرسانی می‌کنیم و قانون بهروزرسانی Adam به شکل زیر ایجاد می‌شود:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

مقادیر پیشنهادی برای β_1 برابر با ۰/۹ و برای β_2 برابر با ۰/۹۹۹ و برای ϵ برابر با 10^{-8} می‌باشد.

و به این شکل با استفاده از الگوریتم‌های بهینه سازی فراپارامترهایی همچون نرخ یادگیری و نرخ اکتشاف آدایپته می‌شوند و الگوریتم یادگیری تقویتی عمیق استراتژی‌های خود را با ابعاد مختلف در بازی تطبیق و آموزش می‌دهد.

کتابخانه‌های مورد استفاده

PyGame برای ایجاد اینترفیس بازی. Math برای استفاده از عملگرهای ریاضی مانند floor در کد.

sys برای استفاده از توابعی مانند exit(). ماژول constants از utilities برای تعریف و استفاده از ثابت‌ها در تمام فایل‌ها

Entities برای مدیریت، اعتبار سنجی و سریال سازی خودکار اشیا مانند Game, Agent, Memory.

itertools برای ایجاد لوپ‌های کارا با استفاده از ماژول count (ایجاد تکرار کننده برای تولید مقادیر با فاصله مساوی).

Matplotlib برای ایجاد نمودارهای گرافیکی (این بخش در هنگام تست الگوریتم‌ها و کارایی روش‌های بهینه سازی و رسم نمودار loss مورد استفاده قرار گرفته است).



نویسنده: هومن امیدیان

MARVEL
CINEMATIC
UNIVERSE

در این مطلب، به بررسی دنیای سینمایی مارول (MCU) می‌پردازیم که با ترکیب داستان‌ها و شخصیت‌های متنوع از کتاب‌های کمیک مارول، یک دنیای داستانی گستردگی و پیچیده را ایجاد کرده است. شبکه گراف دنیای سینمایی مارول، که در نرم‌افزار Gephi پیاده‌سازی شده است، ارتباطات و تعاملات بین شخصیت‌های مختلف این دنیا را به خوبی نمایش می‌دهد. این شبکه، نودها و یال‌های مختلفی را شامل می‌شود که هر کدام نشان‌دهنده شخصیت‌ها و روابط بین آن‌ها هستند. با بررسی این شبکه، می‌توان به اهمیت و تأثیر هر شخصیت و گروه در دنیای سینمایی مارول پی‌برد و درک بهتری از ارتباطات و تعاملات پیچیده بین آن‌ها داشت.

نرم افزار Gephi

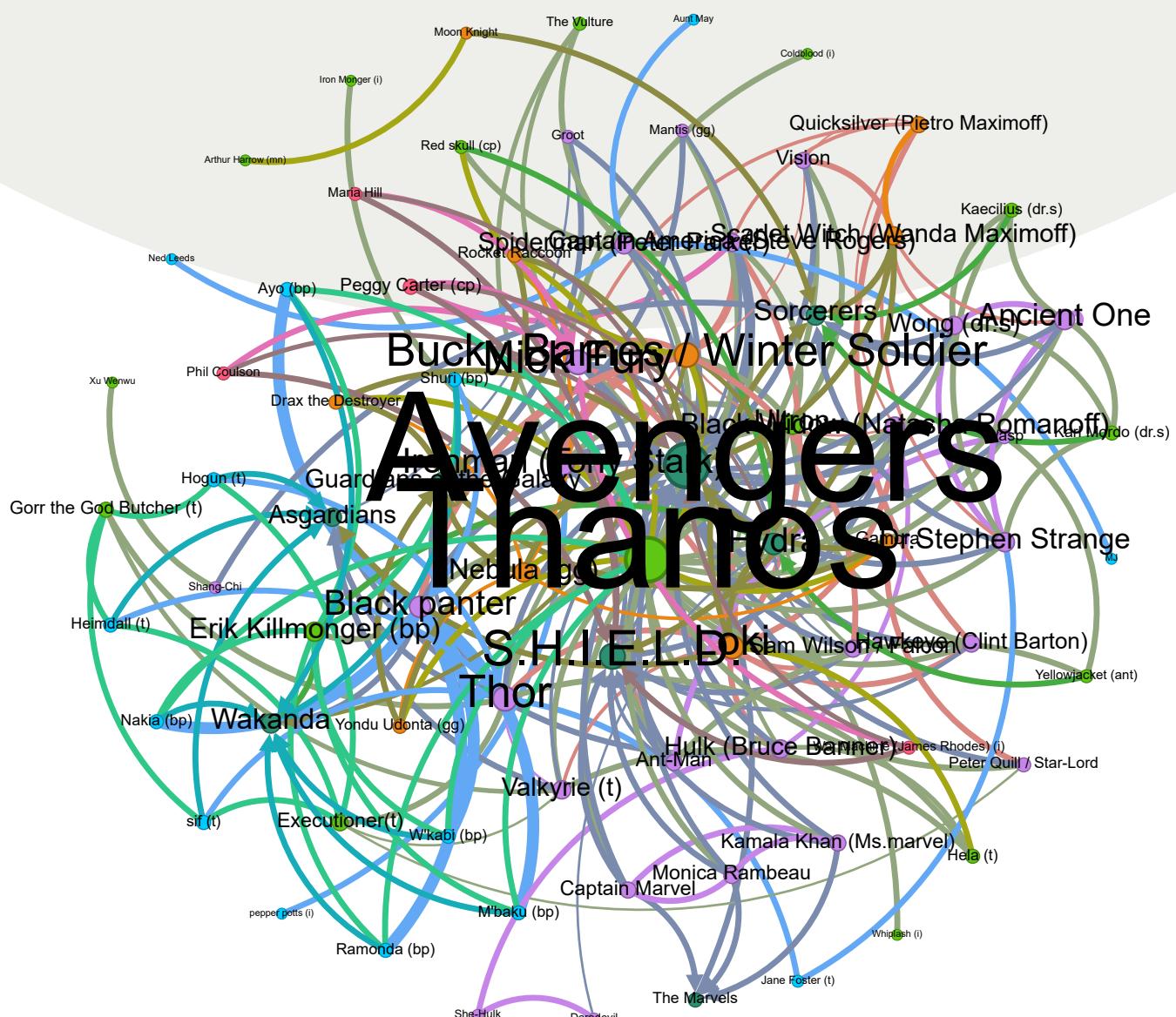
نرم افزاری متن باز (open-source) جهت مصورسازی و انجام عملیات و الگوریتم‌های مختلف روی گراف می‌باشد که توسط دانشجویان دانشگاه University of Technology of Compiègne (UTC) در فرانسه توسعه داده شده و آخرین نسخه آن ۱.۰/۱.۰ می‌باشد.

این نرم افزار برای محققان، دانشجویان و حتی کاربران عادی بسیار مفید است و می‌تواند در تحلیل داده‌های اجتماعی، شبکه‌های ارتباطی، وب‌سایت‌ها و بسیاری دیگر از حوزه‌های مختلف استفاده شود.



دنیای سینمایی مارول (MCU) یک مجموعه چندرسانه‌ای آمریکایی است که داستان‌ها و شخصیت‌های خود را از کتاب‌های کمیک مارول کامیکس می‌گیرد. این دنیا توسط مارول استودیوز ساخته شده و شامل فیلم‌ها، سریال‌های تلویزیونی، فیلم‌های کوتاه، سریال‌های دیجیتال و کتاب‌ها می‌شود. فیلم‌ها و سریال‌های این مجموعه در یک دنیای داستانی مشترک به نام زمین ۱۹۹۹۹ مارول جریان دارند. شخصیت‌های این داستان‌ها از کتاب‌های کمیک مارول که در زمین ۶۱۶ مارول جریان دارند، الهام گرفته شده‌اند. در برخی موارد نیز از نسخه‌های دیگر، مانند زمین ۱۶۰ (که به عنوان دنیای نهایی شناخته می‌شود)، استفاده می‌شود. به عنوان مثال، نیک فیوری در فیلم‌های MCU سیاپوست است، مشابه نسخه او در زمین ۱۶۰.

این دنیا، همانند دنیای اصلی مارول در کتاب‌های کمیک، با ایجاد ارتباطات و تقاطع‌های داستانی بین شخصیت‌ها، مکان‌ها و داستان‌های مختلف شکل گرفته است. در نتیجه، فیلم‌ها و سریال‌ها در دنیای سینمایی مارول به یکدیگر مرتبط هستند و داستان‌ها و شخصیت‌ها به صورت مشترک در آن‌ها ظاهر می‌شوند.



از جمله نودهای مهم شبکه گروه انتقام جویان (Avengers) که شخصیت‌های زیادی عضو این گروه‌اند، شخصیت منفی تانوس (Thanos) که دشمن مشترک تقریباً تمام شخصیت‌های این یونیورس هست، می‌باشد.

در گراف‌ها، روابط بین نودها می‌توانند به دو نوع اصلی تقسیم شوند: روابط جهت‌دار (Directed) و روابط بدون جهت (Undirected).

روابط Directed به روابطی اشاره دارند که جهت خاصی دارند. در این نوع روابط، ارتباط بین دو عنصر به صورت یک جهت بوده و به معنای وجود یک رابطه یک‌طرفه است. به عبارت دیگر، در روابط Directed، اطلاعات و تأثیر بین دو عنصر به صورت یک جهت منتقل می‌شود و ممکن است یک طرف از رابطه تأثیری بر طرف دیگر نداشته باشد.

در این جدول، روابطی که با نوع (undirected) مشخص شده‌اند، نشان دهنده روابط دوجانبه و بدون جهت هستند که به معنای تعامل و ارتباط دو طرف است.

شبکه گراف دنیای سینمایی مارول (MCU)

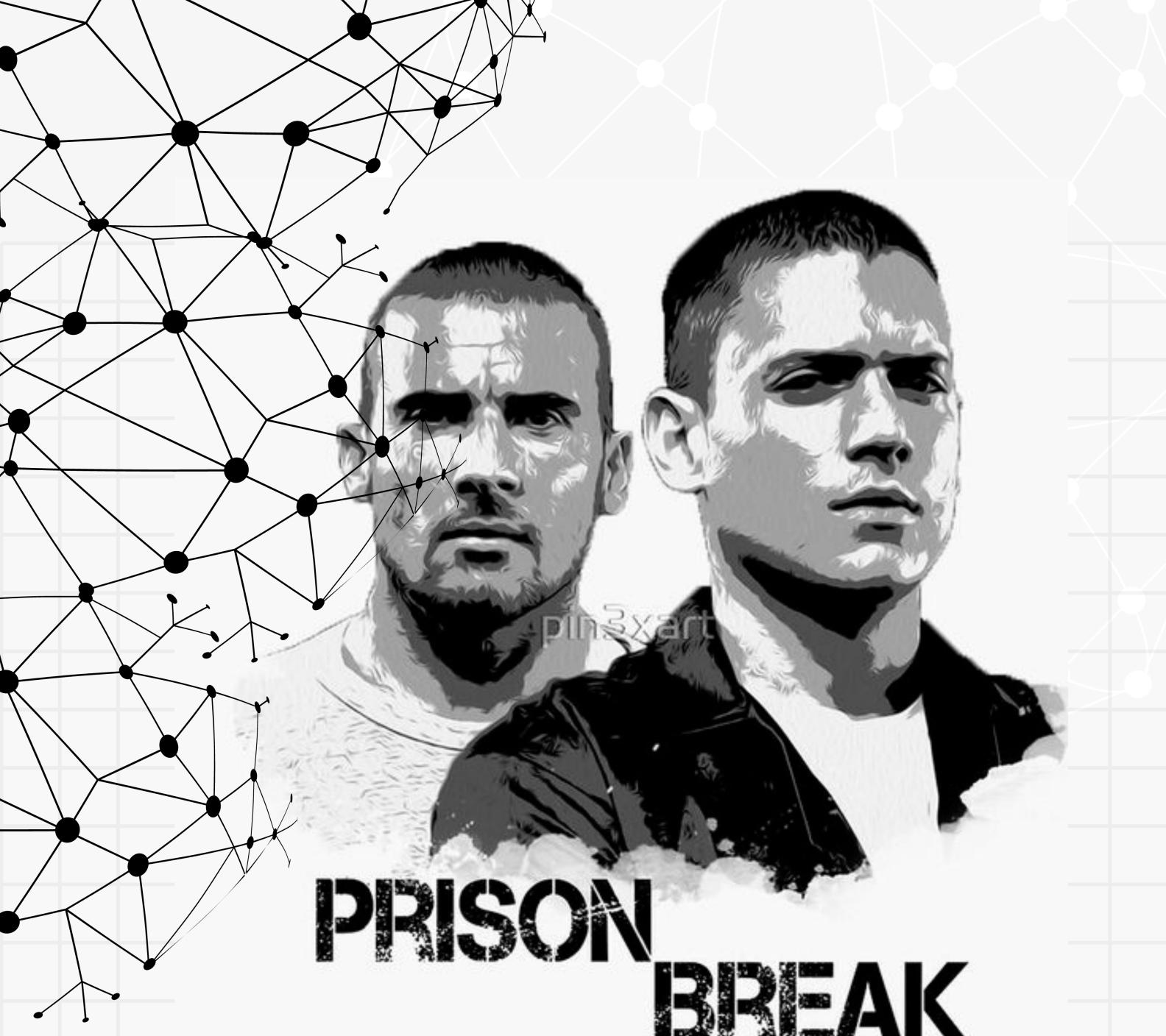
در این شبکه کاراکترها و ارتباط بین آن‌ها در مجموعه فیلم‌های سینمایی و سریال‌های دنیای سینمایی مارول در نرم افزار Gephi پیاده سازی شده است.

هر نod از این گراف یک شخصیت و یا یک گروه می‌باشد که در دسته قهرمان (Hero)، دشمنان (Villain)، ضدقهرمان‌ها (anti-hero)، ماموران (Groups)، گروه‌ها (Agent) و شخصیت‌های جانبی (side characters) قرار می‌گیرد.

از طرفی هر یال بین دو Nod نشانگر ارتباط بین نودها و شدت ارتباطشان را تعیین می‌کند که حداقل مقدار آن ده می‌باشد و شامل دشمن بودن دو Nod (Enemy)، عضویت یک Nod در یک گروه (Member)، معشوقه بودن دو Nod (Lover)، روابط خانوادگی مانند Sibling یا Aunt، دوستی دو Nod (Friend) و در نهایت رئیس بودن یک Nod (Boss) می‌باشد.

در جدول زیر نمونه یال‌های این شبکه برای شخصیت spider-man / peter parker شده است :

Source	Target	Type	Id	Label	Weight
Spiderman	The Vulture	Undirected	101	Enemy	10
Spiderman	Thanos	Undirected	102	Enemy	10
Spiderman	Winter Soldier	Undirected	103	Enemy	4
Spiderman	Avengers	Directed	104	Member	10
Spiderman	S.H.I.E.L.D	Directed	105	Member	6
Spiderman	Aunt May	Undirected	106	Aunt	10
Spiderman	MJ	Undirected	107	Lover	10
Spiderman	Ned Leeds	Undirected	108	Friend	10



گراف سریال فرار از زندان



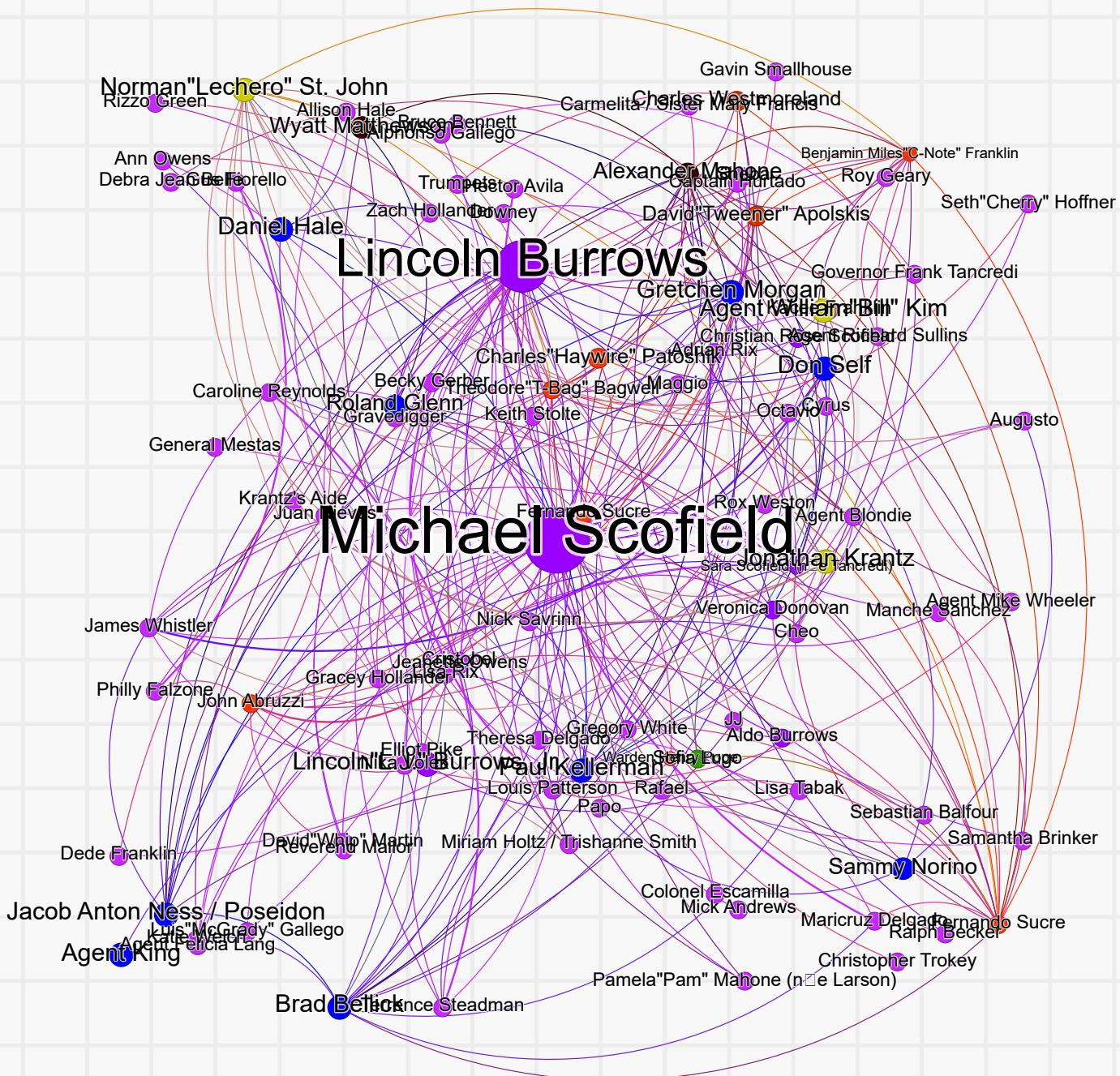
نویسنده: حدیثه خراسانی

گراف ارتباطات شخصیت‌های «فرار از زندان»: سفری به دنیای توطئه و رفاقت



سریال «فرار از زندان» با داستانی پر پیچ و خم و شخصیت‌های جذاب، به یکی از محبوب‌ترین سریال‌های تاریخ تلویزیون تبدیل شده است. اما در ورای این داستان پرهیجان، شبکه‌ای پیچیده از روابط بین شخصیت‌ها وجود دارد.

در این بخش، به بررسی گراف ارتباطات شخصیت‌های این سریال می‌پردازیم. این گراف که شامل ۱۰۰ نод و ۲۵۹ یال است، روابط بین شخصیت‌ها را به تصویر می‌کشد. هر یال در این گراف نشان‌دهنده نوعی ارتباط بین دو شخصیت است، مانند روابط خانوادگی، قاتل و مقتول، تعقیب‌کنندگان، توطئه‌کنندگان و زندانیان فراری. با بررسی این گراف، می‌توانید به درک عمیق‌تری از روابط پیچیده بین شخصیت‌های «فرار از زندان» دست پیدا کنید.



تحلیل گرافیکی ارتباطات



تقسیم‌پذیری و تعداد اجتماعات گزارش می‌شود. تقسیم‌پذیری معمولاً به میزانی از انطباق اجزای یک شبکه با یکدیگر درون هر اجتماع اشاره دارد، به طوری که ارتباطات داخلی اجتماع قوی‌تر و ارتباطات بین اجتماعی ضعیفتر باشد.

پارامترها:

Randomize: On

Use edge weights: On

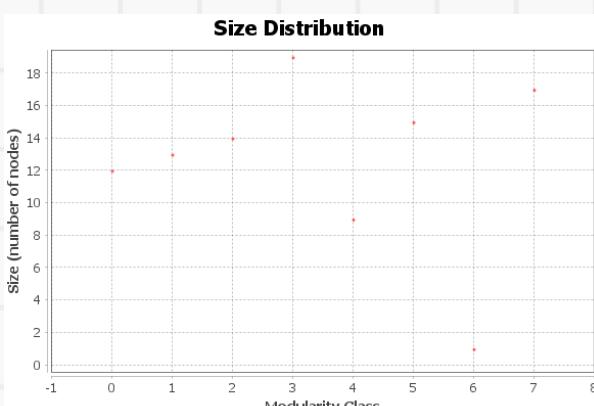
Resolution: 1.0

نتایج:

Modularity: 0.389

Modularity with resolution: 0.389

Number of Communities: 8



الگوریتم:

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, Fast unfolding of communities in large networks, in Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000

در این گراف ارتباطات کاراکترهای فصل‌های موجود در سریال فرار از زندان مورد بررسی قرار گرفته شده است.

این گراف دارای ۱۰۰ نод با ۲۵۹ یال است برای طراحی دقیق تر این گراف از سایت ویکی پدیا استفاده شده است.

در ابتدا با ایجاد دو فایل با پسوند .csv. که یکی که شامل ارتباطات کاراکترها و دیگری فایل edge که نام شخصیت‌های سریال هستند.

ارتباطات بصورت ارتباط خانوادگی، قاتل مقتول، تعقیب کنندگان، توطئه کنندگان، زندانیان فراری و... در نظر گرفته شده است.

شخصیت‌های اصلی این سریال مایکل و برادرش لینکولن هستند که بیشترین ارتباط را با یال‌های دیگر دارند و همچنین خانواده آن‌ها که سایز آنها نسبت به نودهای دیگر بزرگتر و رنگ آنها بنفش است. زندانی‌های فاکس ریور به رنگ نارنجی، افرادی که به فراری دادن زندانیان کمک کردند به رنگ سیاه، افرادی که زندانیان را تعقیب میکردند به رنگ قهوه‌ای و کسانی که علیه لینکولن توطئه کردند به رنگ آبی و کارکنان زندان به رنگ زرد استایل دهی شده‌اند.

گزارش تقسیم‌پذیری

گزارش تقسیم‌پذیری یک گزارش است که نتایج تقسیم‌بندی یک شبکه یا گراف را بر اساس اجتماعات مختلف یا مازول‌ها نشان می‌دهد. در این گزارش، اطلاعاتی مانند پارامترهای استفاده شده برای تقسیم‌بندی (مانند تصادفی‌سازی و استفاده از وزن‌های یال) و نتایجی از جمله میزان

P. Doreian, V. Batagelj, A. Ferligoj (Wiley, 2019)
<https://dx.doi.org/10.1002/9781119483298.ch11>

رزویشن:

R. Lambiotte, J.-C. Delvenne, M. Barahona
Laplacian Dynamics and Multiscale Modular Structure in Networks 2009

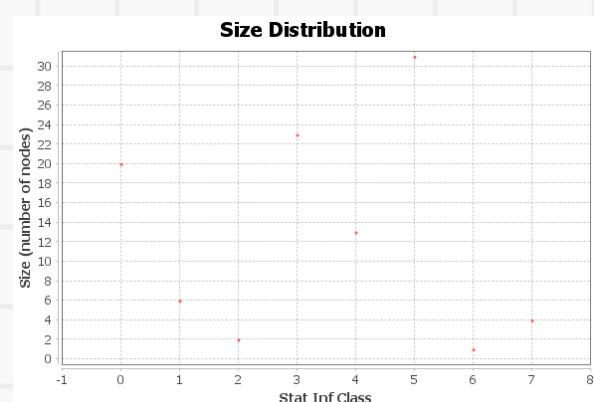


گزارش استنتاج آماری

نتایج :

Description Length: 1006.081

Number of Communities: 8



الگوریتم:

Statistical inference of assortative community structures

Lizhi Zhang, Tiago P. Peixoto

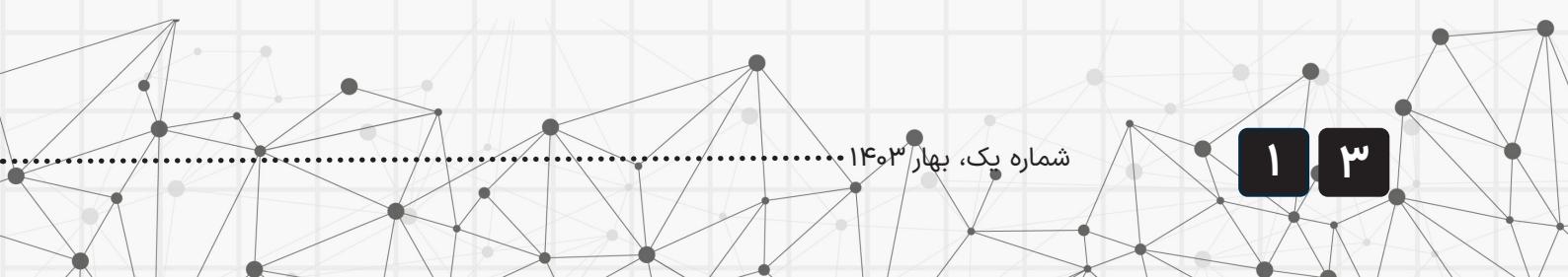
Phys. Rev. Research 2 043271 (2020)

<https://dx.doi.org/10.1103/PhysRevResearch.2.043271>

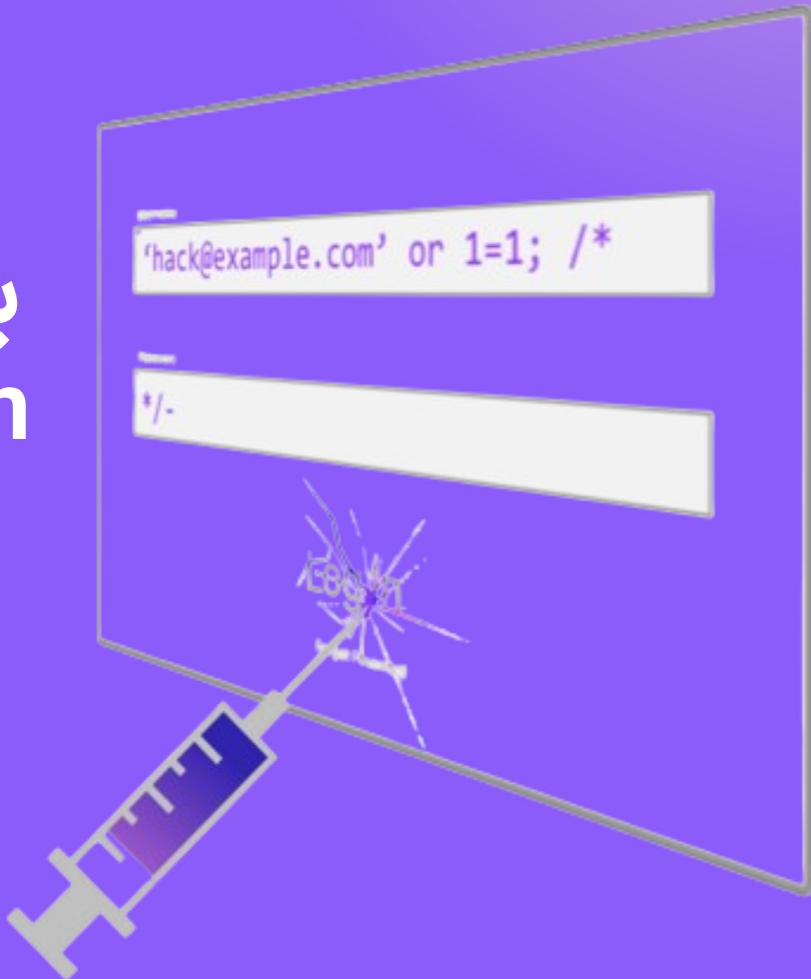
مدل‌سازی بلوکی احتمالاتی بیزین

Tiago P. Peixoto

Chapter in "Advances in Network Clustering and Blockmodeling," edited by



بررسی حملات SQL Injection



حمله SQL Injection یک نوع حمله به امنیت وب است که هنگامی که یک ورودی کاربر در یک فرم یا پارامتر وبسایت به نادرستی اعتبارسنجی نشده یا فیلتر نشده باشد، به مهاجم اجازه می‌دهد تا کد SQL ناخواسته را به داخل یک دستور SQL قرار دهد و اجازه دسترسی به پایگاه داده را به دست آورد یا آن را متلاشی کند. این یک مشکل امنیتی جدی است و توسعه‌دهندگان باید برنامه‌های خود را به این نوع حملات مقاوم کنند.

روش های کلی SQL Injection



دیتابیس می کند. به عنوان مثال:

```
' UNION SELECT username, password  
FROM users; --
```

دستور SQL به صورت زیر ترکیب می شود:

```
SELECT * FROM some_table WHERE column = '' UNION SELECT username, password FROM users; --
```

۱ تزیریق SQL مبتنی بر خطا

در این نوع حمله، مهاجم از ایجاد خطاهای SQL برای استخراج اطلاعات استفاده می کند. به عنوان مثال:

```
' OR 1=CONVERT (int, (SELECT @@version)); --
```

اگر این دستور باعث خطا شود، خطا پیام های مفیدی درباره سیستم می دهد.

۲ تزیریق Boolean-Based SQL

در این حمله، مهاجم از تست شرطی مثل `1=1` یا `1=0` برای استخراج اطلاعات استفاده می کند. به عنوان مثال:

```
' OR 1=1; --
```

```
' OR 1=0; --
```

از تغییرات در عملکرد سیستم برای استنتاج اطلاعات می پردازد.

۳ تذکر

این مثالها صرفاً به عنوان نمونه برای توضیح انواع SQL injection آورده شده اند و در سیستم های واقعی، مهاجمان از ترکیب های مختلف و دستورات پیچیده تری استفاده می کنند تا به اطلاعات محروم نه دست یابند.

حملات SQL Injection انواع گوناگونی دارند که هر کدام به تکنیک ها و نیازهای مهاجمان مرتبط هستند. در ادامه، نمونه هایی از انواع مختلف SQL Injection همراه با مثال آورده شده اند:

۱ تزیریق کلاسیک SQL

در این نوع حمله، مهاجم با تزیریق دستورات SQL مستقیماً به دستور SQL اصلی، سعی در تغییر عملکرد آن دارد. به عنوان مثال، فرض کنید یک فرم ورود با ورودی نام کاربری و رمز عبور داریم:

Username: ' OR '1'='1'; --

Password: (هر مقدار)

دستور SQL به صورت زیر ترکیب می شود:

```
SELECT * FROM users WHERE username = '' OR '1'='1'; --
```

۲ تزیریق Blind SQL مبتنی بر زمان

در این حمله، مهاجم از تأخیر زمانی در پردازش دستورات SQL برای استخراج اطلاعات استفاده می کند. به عنوان مثال:

```
' OR IF (1=1, SLEEP (5), 0); --
```

اگر اجرای دستور باعث تأخیر شود، مهاجم می تواند اطلاعاتی را استخراج کند.

۳ Union-Based SQL

در این تکنیک، هکر با استفاده از اپراتور UNION SQL برای ترکیب چند دستور انتخاب و برگرداندن یک پاسخ HTTP، اقدام به استخراج اطلاعات از

روش‌های جلوگیری از این آسیب پذیری

با WAF

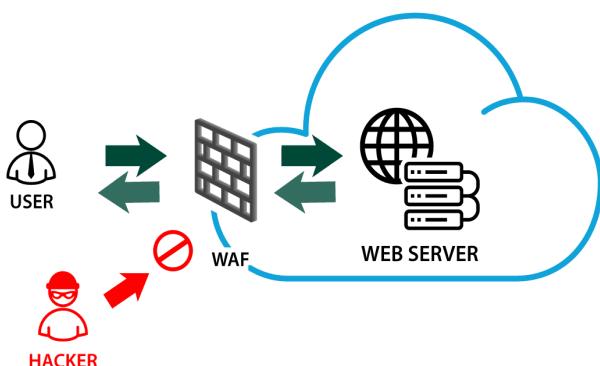
دیواره آتش وب (WAF) باید قابلیت مانیتورینگ و گزارش‌دهی داشته باشد تا تیم امنیتی بتواند حملات SQL Injection را شناسایی و اقدامات اصلاحی را انجام دهد. اطلاعات لازم برای تجزیه و تحلیل بعدی و برطرف کردن آسیب پذیری‌ها باید در دسترس باشد.

توجه به ورودی‌ها

دیواره آتش وب (WAF) می‌تواند ورودی‌ها را به خوبی فیلتر کند، اما توسعه‌دهندگان نیز باید مطمئن شوند که ورودی‌ها به درستی اعتبارسنجی و فیلتر می‌شوند. زمانی که WAF به عنوان لایه دفاعی آخر فعال باشد، اعتبارسنجی ورودی در کد نیز اهمیت دارد.

استفاده از تکنولوژی‌های امنیتی دیگر

دیواره آتش وب (WAF) باید جزء یک استراتژی امنیتی کلی باشد. از تکنولوژی‌های دیگر نیز نظیر IPS و IDS به عنوان لایه‌های اضافی برای حفاظت از سیستم استفاده شود. به طور کلی، استفاده از WAF همراه با تنظیمات صحیح و توجه به توسعه امن در کد، می‌تواند به مقابله با حملات SQL Injection کمک زیادی کند.



دیواره آتش وب (WAF) یکی از ابزارهای مؤثر برای جلوگیری از حملات SQL Injection است. WAF مسئول فیلتر کردن ترافیک وب و جلوگیری از حملات امنیتی در سطح وب سرور می‌باشد. در زیر روش‌هایی جهت استفاده از WAF برای جلوگیری از آسیب پذیری SQL Injection آورده شده‌اند:

۱ ویژگی‌های آماده WAF

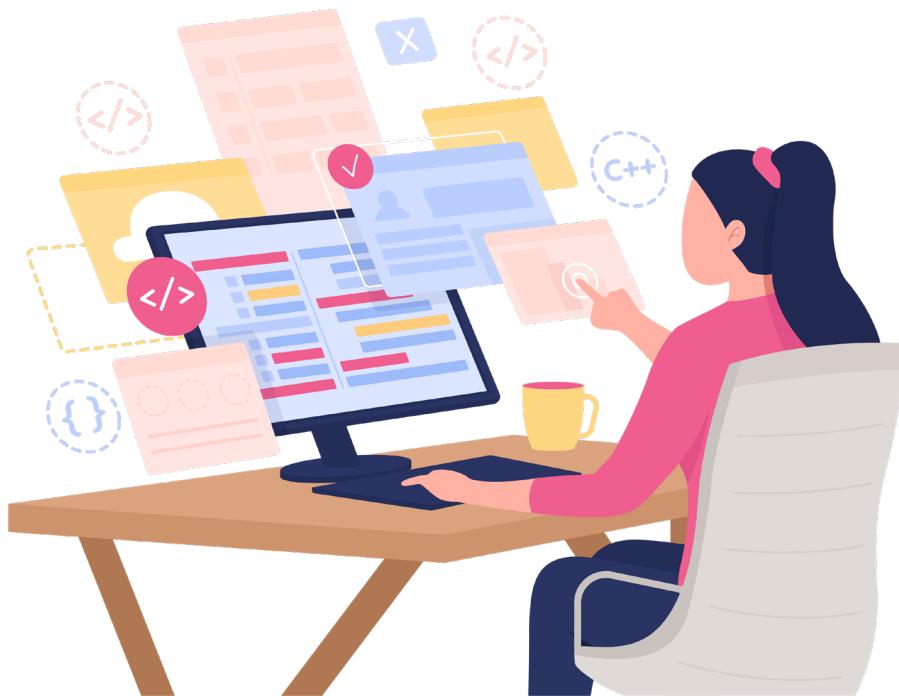
اغلب WAF‌ها ویژگی‌های آماده‌ای برای جلوگیری از حملات SQL Injection ارائه می‌دهند. این ویژگی‌ها معمولاً شامل تشخیص و جلوگیری از الگوهای معروف SQL Injection می‌شوند. به عنوان مثال، تشخیص کلماتی مانند UNION یا SELECT در ورودی‌های کاربر.

۲ تنظیمات سفارشی

دیواره آتش وب (WAF) اغلب امکانات تنظیمات سفارشی برای فیلتر کردن ورودی‌ها و دستورات SQL دارد. این تنظیمات به توسعه‌دهندگان امکان می‌دهند قوانین مخصوص به سیستم خود را تعریف کرده و از الگوهای مخرب مشخص شده جلوگیری کنند.

۳ آموزش WAF

دیواره آتش وب (WAF) اغلب امکانات تنظیمات سفارشی برای فیلتر کردن ورودی‌ها و دستورات SQL دارد. این تنظیمات به توسعه‌دهندگان امکان می‌دهند قوانین مخصوص به سیستم خود را تعریف کرده و از الگوهای مخرب مشخص شده جلوگیری کنند.



روش های جلوگیری از این آسیب پذیری با کدنویسی



در Python با استفاده از کتابخانه sqlite3

```
cursor.execute("SELECT * FROM users WHERE username = ? AND password =?", (user_input_username, hashed_password))
```

برای جلوگیری از آسیب پذیری SQL Injection توسعه دهنده‌گان می‌توانند از چند روش کدنویسی استفاده کنند. در زیر، روش‌هایی که می‌توانند به تقویت امنیت در برنامه‌ها کمک کنند، آورده شده‌اند:

اعتبارسنجی و فیلترینگ داده‌های ورودی ۲

اعتبارسنجی دقیق و فیلترینگ داده‌های ورودی از تزریق دستورات SQL جلوگیری می‌کند. بررسی و اعتبارسنجی داده‌ها بر اساس نوع مورد انتظار و حداقل و حداقل طول آنها اهمیت دارد.

استفاده از Prepared Statements ۱

استفاده از Prepared Statements یا Parameterized Queries دستورات SQL با ارسال پارامترهای جداگانه؛ این روش اجازه می‌دهد تا ورودی‌های کاربر به عنوان داده تفسیر شوند و اجتناب از ترکیب مستقیم با دستورات SQL را فراهم می‌کند.



با استفاده از کتابخانه `re` در پایتون:

```
import re
def validate_input(input_data):
    if re.match("^[a-zA-Z0-9_-]+$", input_data):
        return True
    else:
        return False
```

۵ تنظیمات امنیتی در سرور و فریمورک

تنظیمات امنیتی مرتبط با سرور و فریمورک مورد استفاده نیز باید به روز باشند. این شامل HTTP Strict Transport Security (HSTS) و Content Security Policy (CSP) می‌شود.

۶ تعلیمات امنیتی برای توسعه‌دهنگان

توسعه‌دهنگان باید با تکنیک‌ها و مهارت‌های مرتبط با امنیت آشنا باشند. ارائه دوره‌های آموزشی و تعلیمات به توسعه‌دهنگان در خصوص مسائل امنیتی و بهروزرسانی آنها از اهمیت زیادی برخوردار است.

ترکیب این روش‌ها به یک استراتژی جامع امنیتی منجر خواهد شد و احتمال حملات SQL Injection را به حداقل می‌رساند.

۳ استفاده از Object-Relational Mapping

رابطه نقشه برداری به شی (ORM) به توسعه‌دهنگان امکان می‌دهند با استفاده از شیء‌گرایی و بدون نیاز به نوشتن دستورات SQL مستقیماً با دیتابیس ارتباط برقرار کنند. این کتابخانه‌ها اغلب دستورات SQL را به صورت خودکار تولید می‌کنند و از این راه از آسیب‌پذیری‌های SQL Injection جلوگیری می‌کنند.

۴ تقویت امنیت پایگاه داده

استفاده از مواردی مانند فیلترینگ دسترسی به جداول (Table Permissions) و تنظیمات دسترسی به داده‌ها (Data Permissions) در سطح دیتابیس به تقویت امنیت کمک می‌کند. همچنین، استفاده از رمزنگاری برای اطلاعات حساس در دیتابیس اهمیت دارد.

بررسی یک سایت آسیب پذیر



A screenshot of a Google search results page for the query "dork for sql injection". The results are filtered under the "All" category. The page shows approximately 346,000 results found in 0.36 seconds. The URL for the search is https://www.boxpiper.com/posts/google-dork-list-s... . A red oval highlights the search term "dork for sql injection" in the search bar.

در ابتدا به دنبال سایتی بودیم که آسیب پذیری SQL Injection داشته باشد. برای این کار باید دورک هایی پیدا میکردیم که احتمال وجود آشتن این آسیب پذیری در آن ها باشد.

به ترتیب شروع به امتحان کردن دورک ها کردیم و سایت هارا از نظر SQL Injection بررسی کردیم. که سپس با دورک به index.php?id= این سایت رسیدم که از نظر SQL Injection آسیب پذیر بود:

A list of common SQL injection attack vectors extracted from the search results. The vectors listed are news.pnp?ID=, photogallery.php?id=, index.php?id=, product/product.php?product, nyheder.htm?show=, book.php?ID=, and print.php?id=.

A screenshot of a French e-commerce website (epe.fr) showing a product listing for a laptop. The product is labeled "PEEL 002" with a price of "13,50 € TTC". The URL is https://www.eshop/achat/index.php?id=1.

با وارد کردن 'OR+1=1--' بعد از id=1 متوجه این آسیب پذیری شدیم.

همانطور که در تصویر دیده می‌شود نوع دیتابیس سایت را برمی‌گرداند.

A screenshot of a MySQL error log. The errors shown are:

```
Vous êtes ici : accueil >
[104] [INFO] mysql_num_rows(): supplied argument is not a valid MySQL result resource #104/sda/1/c/www.eshop/achat/index.php on line 4
[104] [INFO] mysql_num_rows(): supplied argument is not a valid MySQL result resource #104/sda/1/c/www.eshop/achat/index.php on line 4
[104] [INFO] produit disponible dans cette catégorie
```

A screenshot of a MySQL error log. The errors shown are:

```
[110] [INFO] the back-end DBMS is MySQL
[110] [INFO] application technology: PHP, PHP 4.4.3, Apache ProXAD
[110] [INFO] end DBMS: MySQL > 5.0.12
[111] [INFO] fetching database names
[111] [INFO] able databases [1]:
[111] [INFO] information_schema
[112] [INFO] fetched data logged to text files under
[112] [INFO] an output /www.eshop.free.fr
```

برای اطمینان بیشتر SQL Map نیز انجام دادم که در اینجا نیز اشاره کرده بود قسمت آسیب پذیر است و دیتابیس سایت URL MySQL است.

SQL Injection یک تهدید جدی برای برنامه‌های وب و پایگاه‌های داده آنها است. با درک انواع مختلف SQL Injection و نحوه جلوگیری از آنها، می‌توانید به محافظت از داده‌های خود و حفظ امنیت برنامه‌های وب خود کمک کنید.





نویسنده: میلاد رضوانی



اینترفیس‌های خود را کانفیگ می‌کنیم و سپس باید مشخص کنیم که برای LAN یا WAN کدامیک را کانفیگ کنیم. این مراحل را با زدن عدد ۱، برای WAN انجام می‌دهیم. جزئیات تنظیمات پایینی نشان داده شده است.

نصب و پیکربندی PFsense

آیا می‌خواهید که اینترفیس WAN از طریق DHCP آدرس IP بگیره؟

- خیر، زیرا می‌خواهیم آدرس IP را به صورت دستی تنظیم کنیم.

لطفاً آدرس IPv4 را وارد کنید.

- ۱۹۲.۱۶۸.۷۷.۱۲۵ به عنوان آدرس اینترفیس تعیین می‌شود.

لطفاً ماسک زیرشبکه را وارد کنید.

- ما از ماسک ۲۴ استفاده می‌کنیم. (این عدد بسته به اندازه شبکه‌ای که قصد داریم ایجاد کنیم قابل تغییر است، اما به دلیل سادگی، اینجا از ۲۴ استفاده می‌شود).

لطفاً آدرس IPv4 تلفن همراه/مودم را وارد کنید.

- ما آدرس ۱۰۲.۱۰۶.۱۳۶.۱۲۵ را وارد می‌کنیم.

آیا می‌خواهید که این Gateway به عنوان way way پیشفرض تشخیص داده شود؟

- بله.

آیا می‌خواهید که آدرس IPv6 توسط DHCP ارائه شود؟

- خیر.

لطفاً آدرس IPv6 را وارد کنید.

- از آنجایی که از IPv6 استفاده نمی‌کنیم،

PFsense چیست؟

PFsense یک سیستم عامل فایروال و مسیریابی مبتنی بر FreeBSD است که برای مدیریت شبکه‌های کوچک و بزرگ استفاده می‌شود. این سیستم، ابزارها و قابلیت‌های متنوعی را برای ایجاد یک فایروال قدرتمند، مدیریت ترافیک، پیکربندی DHCP، NAT و سایر وظایف شبکه فراهم می‌کند. در این متن، ما به توضیح نحوه نصب و پیکربندی سرویس‌های DHCP و NAT در PFsense می‌پردازیم.

راهاندازی PFsense

هدف از راهاندازی PFsense، ایجاد یک فایروال می‌باشد. ما قصد داریم یک شبکه داخلی ایجاد کنیم و رنج IP را که به Default Gateway اختصاص می‌دهیم، به صورت زیر باشد: ۱۰.۰.۰.۱۰/۲۴. این رنج IP برای دستگاه‌هایی استفاده می‌شود که در داخل یک شرکت یا اتاق قرار دارند.

مرحله‌ی بعدی، پیدا کردن رنج IP مودم یا تلفن همراه است. می‌توانیم از دستور ipconfig در CMD استفاده کنیم تا آدرس IP دستگاه متصل به مودم/تلفن همراه را ببینیم. همچنین، اگر می‌خواهیم از مودم استفاده کنیم و آن را به صورت IP Static Gate-Default تنظیم کرده‌ایم، از آدرس ۱۰.۰.۰.۱۰ آغاز هستیم. اما برای تلفن همراه باید در تنظیمات آن را پیدا کنیم.

بعد از این مراحل، باید PFsense را بر روی یک ماشین مجازی نصب کنیم. حتماً بهیاد داشته باشید که بعد از نصب، برای PFsense دو عدد اینترفیس تعریف کنید؛ یکی برای LAN و دیگری برای WAN. پس از نصب، به یک صفحه‌ی ورودی هدایت می‌شویم که می‌توانید آدرس‌های IP به دست‌آمده را وارد کنید. با زدن عدد ۲،

لطفاً آدرس IPv6 را وارد کنید.

- با فشردن Enter ادامه می‌دهیم زیرا از این بخش استفاده نمی‌کنیم.

آیا می‌خواهید که DHCP داشته باشید؟

- در این قسمت می‌توانیم آن را تعریف کنیم یا اینکه در تنظیمات PFsense در وب آن را تغییر دهیم، تا دستگاه‌های جدید که به شبکه متصل می‌شوند از این رنج IP بگیرند.

بعد از انجام دستورالعمل‌های بالا، می‌خواهیم تا دستگاه را از قسمت WAN به قسمت LAN منتقل کنیم، به همین خاطر لازم است تا آدرس IP دستگاه را به صورت دستی وارد کنیم.

ابتدا در قسمت Control Panel\Network and Internet\Network and Sharing Center گزینه change adapter settings را انتخاب کرده و اینترفیسی که با آن به اینترنت وصل هستیم را پیدا می‌کنیم. سپس بر روی آن کلیک راست کرده و properties را انتخاب می‌کنیم. در بخش internet properties را انتخاب کرده و دوباره tocol version 4 را انتخاب کرده و از حالت اتوماتیک به حالت دستی تغییر می‌دهیم. سپس آدرس IPv4 را به صورت دستی تنظیم می‌کنیم (برای مثال، آدرس 10.0.0.15) و سابت مسک را 255.255.255.0 می‌گذاریم. برای default gateway از آدرس 10.0.0.10 استفاده خواهیم کرد. برای DNS می‌توانیم از همان آدرس استفاده کنیم تا بعداً از طریق pfSense DNS را تعریف کنیم.

حالا که تنظیمات اولیه انجام شد، می‌توانیم با ورود به pfSense از طریق مرورگر وب ادامه دهیم. در

می‌توانیم با فشردن Enter ادامه دهیم.

آیا می‌خواهید که DHCP فعال باشد؟

- خیر، زیرا این بخش برای اتصال دستگاه‌های دیگر استفاده نمی‌شود و ما باید یک سرور DHCP برای بخش LAN تنظیم کنیم تا دستگاه‌های متصل به آن آدرس IP دریافت کنند.

حالا که بخش WAN را پیکربندی کردیم، باید به همین روند برای بخش LAN ادامه دهیم:

آیا می‌خواهید که اینترفیس LAN از طریق آدرس IP بگیره؟

- خیر، زیرا می‌خواهیم آدرس IP را به صورت دستی تنظیم کنیم.

لطفاً آدرس IPv4 را وارد کنید.

ما آدرس 10.0.0.10 را به عنوان آدرس اینترفیس تعیین می‌کنیم.

لطفاً ماسک زیرشبکه را وارد کنید.

ما از ماسک 24 استفاده می‌کنیم. (همانطور که گفته شد، این عدد بسته به اندازه شبکه‌ای که قصد داریم ایجاد کنیم قابل تغییر است).

لطفاً آدرس IPv4 تلفن همراه/مودم را وارد کنید.

با فشردن Enter ادامه می‌دهیم زیرا از این بخش استفاده نمی‌کنیم.

آیا می‌خواهید که آدرس IPv6 توسط DHCP ارائه شود؟

خیر، در اینجا از IPv6 استفاده نمی‌کنیم.



کنیم، هم می‌توانیم انواع DHCP ها را که تنظیم کرده‌ایم را مشاهده کنیم و هم دستگاه‌هایی که آدرس IP گرفته‌اند را ببینیم.

کانفیگ NAT :

در بخش NAT، وقتی می‌خواهیم از Port Forward استفاده کنیم، زمانی استفاده می‌کنیم که یک سرور در داخل شبکه راه‌اندازی کرده‌ایم، اما آدرس IP مشخصی ندارد، به این معنی که از طریق شبکه خارجی نمی‌توانیم به صورت مستقیم، add به آن دسترسی پیدا کنیم. بعد از انتخاب IP اینترفیس را در WAN قرار می‌دهیم، ورژن آدرس IP را انتخاب می‌کنیم و پروتکل اتصالی را مشخص می‌کنیم. سپس، آدرس و پورت مقصد را انتخاب یا به صورت دستی وارد می‌کنیم. آدرس IP سرور مورد نظر را وارد می‌کنیم. بنابراین هر زمان که به آن آدرس IP با پورت مشخص درخواست فرستاده شود، به سمت سرور در داخل شبکه ارسال خواهد شد.

با انجام مراحل ذکر شده، می‌توانید یک شبکه داخلی امن با PFsense راه‌اندازی کنید. برای مشاهده آموزش ویدیویی آن کد QR زیر را اسکن کنید و یا بر روی لینک کلیک کنید:



<https://youtu.be/nzwgcn69sBg>

بخش Diagnostics می‌توانیم Ping را انتخاب کرده و مشاهده کنیم که آیا از قسمت WAN به اینترنت دسترسی داریم یا خیر. در بخش Host-DNS باید آدرس IP یک سایت یا یک سرور را وارد کنیم و در بخش آدرس مبدا باید LAN را انتخاب کنیم تا مشاهده کنیم که آیا از طریق شبکه داخلی به اینترنت دسترسی داریم. سپس همین کار را از طریق WAN تکرار می‌کنیم. در صورتی که جواب Ping به دست آید، این معنی می‌دهد که اینترنت از PFsense عبور می‌کند.

کانفیگ DHCP :

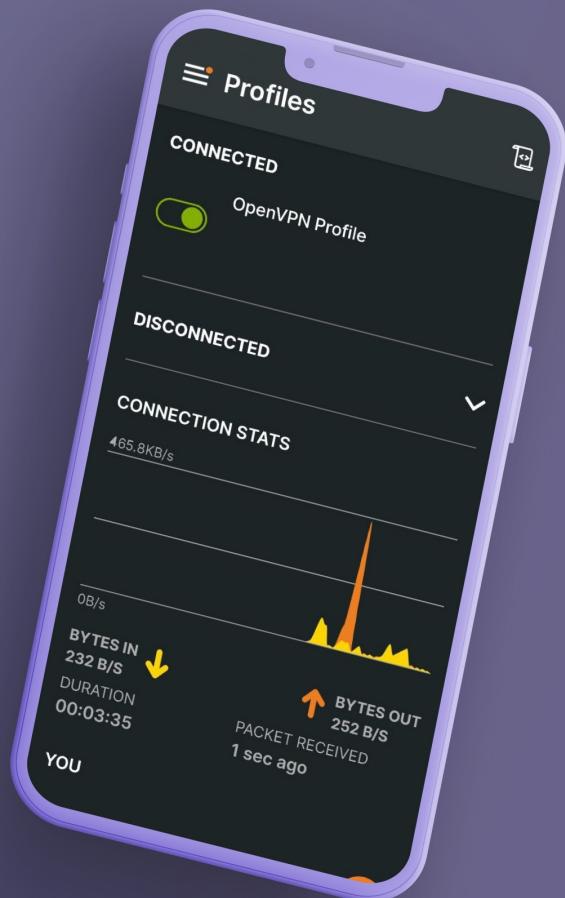
- ابتدا در قسمت Services کلیک کرده و DHCP Server را انتخاب می‌کنیم.
- سپس چون می‌خواهیم در شبکه داخلی DHCP داشته باشیم، کافی است که قسمت LAN را انتخاب کرده و در قسمت General Options گزینه Enable را فعال کنیم.
- در کمی پایین‌تر، از ما می‌خواهد تا یک رنگی را مشخص کنیم تا دستگاه‌های جدید بتوانند در آن محدوده آدرس IP بگیرند. می‌توانیم از 10.0.0.20 شروع کرده و تا 10.0.0.100 قرار دهیم.
- اگر بخواهیم DHCP را امتحان کنیم، می‌توانیم یک سیستم عامل لینوکس یا ویندوز را در VM راه‌اندازی کرده و وقتی که به اینترنت متصل می‌شویم، مشاهده می‌شود که به صورت اتوماتیک یک آدرس IP گرفته و مثلًاً 10.0.0.21 است. همچنین، در PFsense هم اگر به قسمت DHCP Leases رفته و Status را انتخاب



OPENVPN

نویسنده: پر迪س مدد و علیسا حسین آبادی

نرم افزار OpenVPN چیست؟



نرم افزار OpenVPN یک نرم افزار متن باز و رایگان برای ایجاد شبکه های خصوصی مجازی (VPN) است. نرم افزار OpenVPN از پروتکل های رمزگذاری قوی مانند AES و TLS برای محافظت از ترافیک شبکه در برابر شنود و دستکاری استفاده می کند.

در دنیای امروز که پویایی و ارتباطات حرف اول را می زنند، شبکه های VPN سازمانی نقشی حیاتی در اتصال امن و کارآمد افراد، دفاتر و منابع مختلف یک سازمان ایفا می کنند. این شبکه ها، حکم تونل های خصوصی را دارند که به واسطه آنها، می توان از طریق اینترنت عمومی، به صورت امن و رمزگاری شده به منابع شبکه داخلی شرکت دسترسی پیدا کرد. در این بخش نحوه راه اندازی OpenVPN در کالی لینوکس و PFsense مورد بررسی قرار می گیرد.

با این دستور، OpenVPN به سیستم اضافه می‌شود و ما آماده ورود مراحل بعدی برای تنظیمات سرور خواهیم بود.

مرحله ۲: تنظیمات سرور

ایجاد پوشه برای مخاطبین VPN

در این مرحله، ما یک پوشه برای نگهداری تنظیمات کلاینت‌های VPN ایجاد می‌کنیم.

- ایجاد پوشه: با استفاده از دستور زیر، یک پوشه به نام "clients" برای نگهداری تنظیمات کلاینت‌های VPN ایجاد می‌شود.

```
mkdir /etc/openvpn/clients
```

این پوشه به عنوان مکانی برای نگهداری گواهینامه‌ها و کلیدهای مربوط به کلاینت‌های VPN شما خواهد بود. این کار به نظم دادن تنظیمات و افزایش امنیت کل سیستم کمک می‌کند.

ایجاد فایل تنظیمات سرور : "server.conf"

حالا که پوشه کلاین‌های VPN ایجاد شده است، باید یک فایل تنظیمات برای سرور ایجاد کرده و تنظیمات مورد نیاز را در آن قرار دهیم.

از دستور زیر برای ایجاد و باز کردن فایل تنظیمات سرور استفاده می‌کنیم.

```
sudo nano /etc/openvpn/server.conf
```

۱. محتوای فایل "server.conf"

در این فایل، تمامی تنظیمات مورد نیاز برای سرور OpenVPN قرار می‌گیرد. این تنظیمات شامل اطلاعاتی مانند پورت، پروتوكول، مسیرهای گواهینامه، و سایر تنظیمات امنیتی است. محتوای زیر را در

مقدمه‌ای بر راه اندازی OpenVPN در کالی لینوکس:

راهاندازی یک سرور VPN با استفاده از OpenVPN در سیستم عامل کالی لینوکس می‌تواند یک راه حل امن و کارآمد برای اتصال به شبکه اینترنت از دور باشد. OpenVPN یک نرم‌افزار متن‌باز و پرکاربرد برای پیاده‌سازی شبکه‌های خصوصی مجازی (VPN) است که اطلاعات را به طور رمزشده از طریق اتصالات امن ارسال و دریافت می‌کند.

در این بخش به توضیح کامل نصب و پیکربندی OpenVPN در کالی لینوکس پرداخته ایم. این شامل بهروزرسانی سیستم، نصب OpenVPN، تنظیمات سرور، ایجاد گواهینامه‌ها و کلیدها، راهاندازی سرویس OpenVPN، و نیز ایجاد فایل کانفیگ برای کلاینت‌ها می‌شود.

همچنین، در مراحل اتصال کلاینت به سرور، نحوه انتقال فایل‌های تنظیمات از سرور به کلاینت از طریق ابزارهایی مانند SCP و نیز نصب و راهاندازی نرم‌افزار OpenVPN بر روی کلاینت توضیح داده شده است. این راهنمای استفاده از مفاهیم امنیتی و روش‌های بهینه‌سازی، یک محیط VPN امن و پایدار را فراهم می‌کند.

مرحله ۱: نصب OpenVPN Server

۱. به روزرسانی کالی لینوکس:

```
sudo apt update
```

```
sudo apt upgrade
```

ابتدا، اطمینان حاصل می‌کنیم که سیستم به روزرسانی شده باشد. این باعث ایجاد امنیت بیشتر و جلوگیری از مشکلات ناشی از نسخه‌های قدیمی می‌شود.

```
sudo apt install openvpn
```

از کلیدها و تنها حتی در صورت باز شدن و بسته شدن مجدد اتصال.

verb و status : تنظیمات گزارش‌دهی و نمایش اطلاعات.

۱. ذخیره و بستن: پس از انجام تغییرات، فایل را ذخیره کنید و خارج شوید (در نانو، این عمل با فشردن $Ctrl + O$ و سپس $Ctrl + X$ انجام می‌شود).

حالا فایل تنظیمات سرور "server.conf" آماده است و می‌توانیم به مرحله‌های بعدی برویم.

مرحله ۳: تنظیمات گواهینامه و کلیدها در این مرحله، ما گواهینامه‌ها و کلیدهای مورد نیاز برای امنیت ارتباطات بین سرور و کلاینت‌ها را ایجاد می‌کنیم.

ایجاد گواهینامه CA: ابتدا، یک گواهینامه اختصاصی برای اعتبار سازی (CA) ایجاد می‌شود.

```
cd /etc/openvpn/clients  
sudo openssl genrsa -out ca.key 2048  
sudo openssl req -new -x509 -days 3650  
-key ca.key -out ca.crt
```

```
[apar@Kali]:~$ sudo mkdir /etc/openvpn/clients  
cd /etc/openvpn/clients  
sudo openssl genrsa -out ca.key 2048  
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt  
sudo openssl dhparam -out dh.pem 2048  
mkdir: cannot create directory '/etc/openvpn/clients': File exists  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
Country Name (2 letter code) [AU]:ir  
State or Province Name (full name) [Some-State]:azarbajian gharbi  
Locality Name (eg, city) []:urmi  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ccuu  
Organizational Unit Name (eg, section) []:compu  
Common Name (e.g. server FQDN or YOUR name) []:Pardis  
Email Address []:mdpardis084@gmail.com  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
Country Name (2 letter code) [AU]:ir  
State or Province Name (full name) [Some-State]:azarbijan gharbi  
Locality Name (eg, city) []:urmi  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ccuu  
Organizational Unit Name (eg, section) []:compu  
Common Name (e.g. server FQDN or YOUR name) []:Pardis  
Email Address []:Mdpardis084@gmail.com
```

فایل "server.conf" قرار می‌دهیم:

```
port 1194  
proto udp  
dev tun  
ca /etc/openvpn/clients/ca.crt  
cert /etc/openvpn/clients/server.crt  
key /etc/openvpn/clients/server.key  
dh /etc/openvpn/clients/dh.pem  
server 10.8.0.0 255.255.255.0  
ifconfig-pool-persist ipp.txt  
push "redirect-gateway def1 bypass-dhcp"  
push "dhcp-option DNS 8.8.8.8"  
keepalive 10 120  
comp-lzo  
persist-key  
persist-tun  
status openvpn-status.log  
verb 3
```

• port: شماره پورت مورد استفاده برای اتصالات OpenVPN

• proto udp: استفاده از پروتوكول UDP برای افزایش سرعت انتقال داده.

• dev tun: استفاده از دستگاه مجازی (TUN) برای ایجاد اتصال VPN.

• :ca, cert, key, dh گواهینامه و کلیدهای مربوط به سرور.

• server: تعیین زیرشبکه IP برای کلاینت‌های VPN.

• ifconfig-pool-persist : ذخیره اطلاعات IP کلاینت‌ها.

• push: تنظیمات اضافی برای ارسال به کلاینت‌ها.

• Keepalive: تنظیمات نگهداشتن اتصال زنده.

• comp-lzo: فشرده‌سازی ترافیک.

• persist-tun و persist-key: ادامه استفاده

مرحله ۴: تنظیمات فایروال

در این مرحله، اگر از یک فایروال مانند UFW استفاده کرده‌اید، باید آن را تنظیم کنیم تا اجازه عبور ترافیک VPN را بدهد.

۱. **تنظیمات برای UFW:** اگر از UFW به عنوان فایروال استفاده می‌کنید، می‌توانید اجازه عبور ترافیک VPN را اعمال کنید.

```
sudo ufw allow 1194/udp
```

```
sudo ufw enable
```

- :sudo ufw allow 1194/udp
 - این دستور پورت 1194 UDP را برای اتصالات OpenVPN مجاز می‌کند.

- :sudo ufw enable
 - این دستور فایروال را فعال می‌کند.

طمئن می‌شویم که فایروال تنظیم شده و اجازه عبور ترافیک VPN را دارد.

حالا با تنظیمات فایروال، ما می‌توانیم به مرحله بعدی، یعنی راهاندازی سرویس OpenVPN بپردازیم.

مرحله ۵: راهاندازی سرویس

در این مرحله ما، سرویس OpenVpn را راه اندازی و فعال می‌کنیم.

۱. **اجرای سرویس OpenVPN:** از دستور زیر برای راهاندازی سرویس OpenVPN استفاده می‌کنیم:

```
sudo systemctl start openvpn
```

فعالسازی سرویس OpenVPN: با اجرای دستور زیر، سرویس OpenVPN بعد از بارگذاری سیستم

- .CA: کلید خصوصی CA
- .crt: گواهینامه عمومی CA

ایجاد کلید سرور: حالا باید یک کلید خصوصی و گواهینامه عمومی برای سرور ایجاد کنیم.

```
sudo openssl genrsa -out server.key 2048
```

```
sudo openssl req -new -key server.key -out server.csr
```

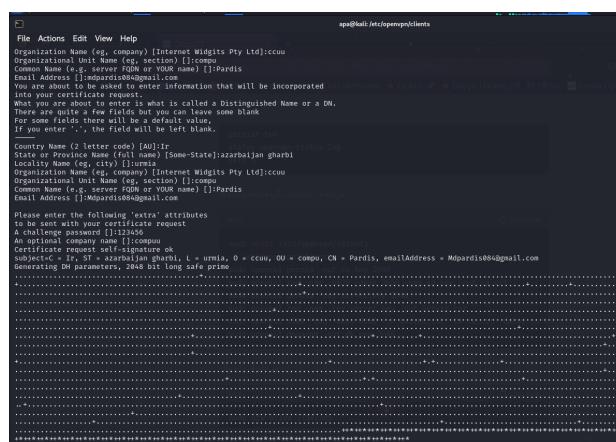
```
sudo openssl x509 -req -days 3650 -in server.csr -CA ca.crt -CAkey ca.key -out server.crt
```

ایجاد کلید DH (Diffie-Hellman): در نهایت، برای افزایش امنیت اتصال، یک کلید-Diffie-Hellman (DH) ایجاد می‌شود.

```
sudo openssl dhparam -out dh.pem 2048
```

۱. این کلید به عنوان یک کلید Exchange برای افزایش امنیت اتصالات مورد استفاده قرار می‌گیرد.

حالا با ایجاد این گواهینامه‌ها و کلیدها، مراحل تنظیمات امنیتی مربوط به سرور OpenVPN تکمیل شده است و می‌توان به مراحل بعدی ادامه داد.



۲. محتوای "client.ovpn":

در فایل "client.ovpn"، محتوای زیر را قرار دهید.
در اینجا <آدرس سرور> را با IP یا نام دامنه سرور
OpenVPN خود جایگزین کنید.

```
client
dev tun
proto udp
remote <Server Address>1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert client.crt
key client.key
remote-cert-tls server
comp-lzo
verb 3
```

ذخیره و بستن: فایل "client.ovpn" را ذخیره
می‌کنیم و ویرایشگر را می‌بندیم.



به صورت خودکار فعال می‌شود:

```
sudo systemctl enable openvpn
```

۱. حالا سرور OpenVPN شما آماده اتصال
است.

با این مرحله، تنظیمات سرور OpenVPN تکمیل
شده و می‌توانیم به مراحل بعدی برویم، که شامل
ایجاد فایل کانفیگ برای کلاینت‌ها است.

مرحله ۶: تهیه فایل کانفیگ برای کلاینت

در این مرحله، ما باید یک فایل کانفیگ برای هر
کلاینت VPN خود ایجاد کنیم.

۱. ایجاد پوشه برای کلاینت **VPN**: ابتدا
یک پوشه برای نگهداری تنظیمات کلاینت
VPN می‌سازیم:

```
mkdir ~vpn-client
```

کپی کردن فایل‌های مورد نیاز: حالا فایل‌های مورد
نیاز از سرور به پوشه کلاینت منتقل می‌شوند.

```
sudo cp /etc/openvpn/clients/ca.crt ~/vpn-client/
```

```
sudo cp /etc/openvpn/clients/client.crt ~/vpn-client/
```

```
sudo cp /etc/openvpn/clients/client.key ~/vpn-client/
```

```
sudo cp /etc/openvpn/clients/dh.pem ~/vpn-client/
```

۱. ایجاد فایل کانفیگ "client.ovpn":
سپس با استفاده از ویرایشگر متنی (مثل نانو)
یک فایل کانفیگ با نام "client.ovpn" برای
کلاینت ایجاد کنید:

```
nano ~vpn-client/client.ovpn
```

مقدمه ای بر راه اندازی OpenVPN در PFsense

راه اندازی OpenVPN در PFsense برای ایجاد یک اتصال امن و ایمن به شبکه مجازی (VPN) (VPN) بسیار اساسی است.

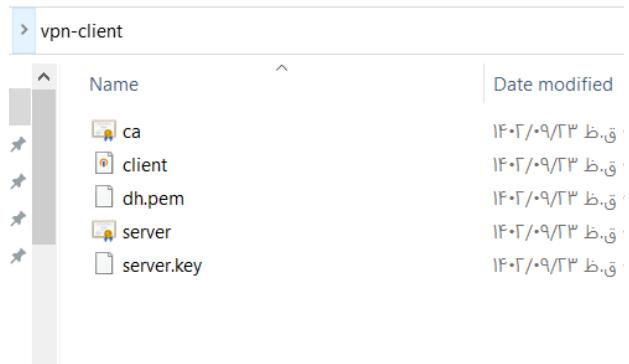
در این مطلب به نحوه راه اندازی OpenVPN در کالی لینوکس پرداخته شد، با اسکن کد QR زیر و یا کلیک بر روی لینک، به آموزش گام به گام راه اندازی OpenVPN در PFsense در OpenVPN که به صورت تصویری و با توضیحات کامل ارائه شده است، دسترسی خواهد داشت. این آموزش، تمام مراحل لازم برای پیکربندی OpenVPN در PFsense را به طور

کامل شرح می‌دهد:

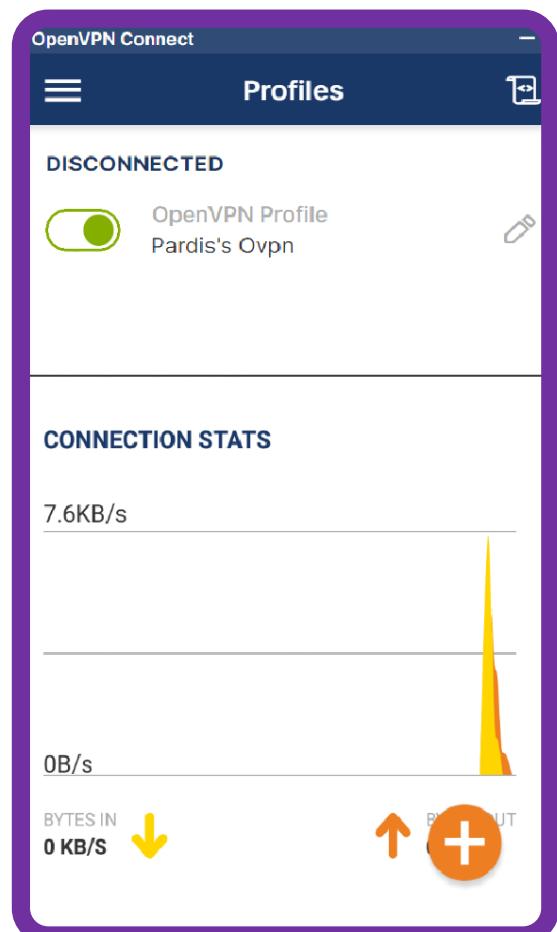


<https://youtu.be/ZSnl6w2vHic>

حالا فایل کانفیگ برای کلاینت ما آماده است. این فایل را به کلاینت VPN خود منتقل می‌کنیم و با استفاده از نرم‌افزار OpenVPN می‌توانیم به سرور OpenVPN متصل شویم.



و کلاینت را در برنامه import می‌کنیم و در نهایت به VPN متصل می‌شویم:

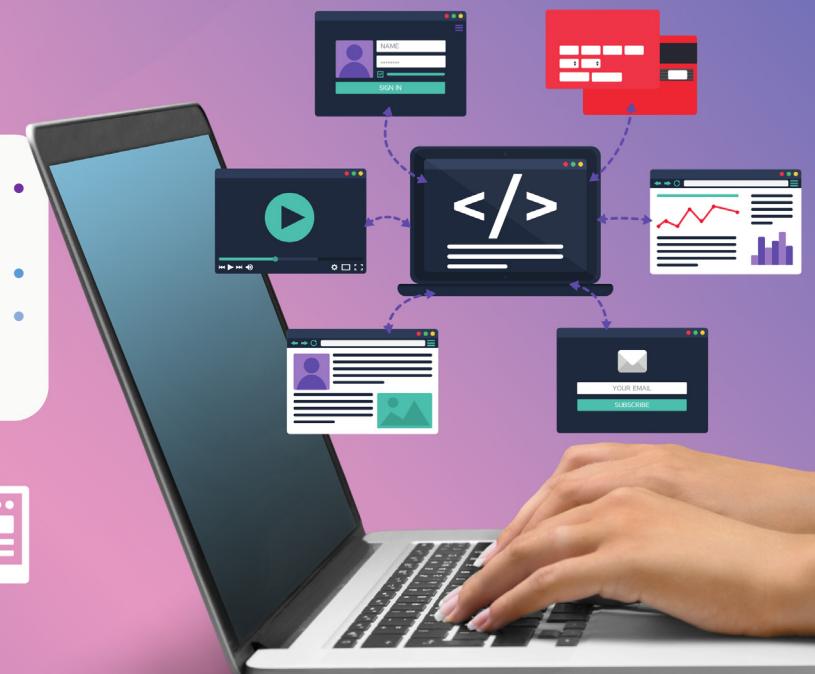


توسعه نرم افزار موفق در پلتفرم‌های آنلاین

نویسنده : رضا شیخ‌لمرزاري

پلتفرم آنلاین مدیریت جامع تعاملات میان کارفرما و پیمانکار داستان کاربری: کلیدی برای توسعه نرم افزار موفق روشهای تحلیل و طراحی سیستم: از تکنیک بارش فکری تا تکنیک اسم

-
-
-



234+ Job posted

Search projects

...

Newly Listed

Show all 232



Need An Experienced Graphic Designer

Posted in Graphic Design

1h ago

\$40

Per hour fixed price

Apply Job



Promotional video for language learning site

Posted in Video Production

1h ago

\$80

Fixed price

Apply Job



Need An Experienced Graphic Designer

Posted in Graphic Design

1h ago

Hello, I am looking for a graphic designer experienced in Adobe Photoshop and Adobe Illustrator for my clothing brand, Genesis.

We need a vectorized graphic design Bellerophon (file attached with a picture of a warrior on a horse). This job will consist of turning an image we provided into a graphic illustration.

We want the graphic illustration to come out like the angel attached to this job listing.

If you think you are the right fit for this job position please attach your resume with previous work.

Attachments (2)

egif.com-gif-maker [7].jpg (371 KB)

Screen Shot 2023-02-02 at 4.28.11 PM.png (1.8 MB)

Project Type

Contract

\$40

Per hour fixed price

Apply Job

Search

UI/UX Design

...

Remote

Fixed Price

Contract

120+ results found



Redesign and improve UI/UX of my portfolio website

Posted in UI/UX Design

1h ago

\$40

Per hour fixed price

Apply Job



UI/UX Pro for Mobile Automotive App

Posted in UI/UX

1h ago

\$1200

Fixed price

Apply Job



UI/UX Expert to redesign my Shopify Store

اولویت‌های ایشان انتخاب و تمرکز بر روی آنها قرار می‌گیرد.

ایجاد رابط کاربری برای پروژه: طراحی رابط کاربری که بهترین تجربه کاربری را ارائه دهد، انجام می‌شود.

ارزیابی نیازمندی‌های پروژه: نیازمندی‌ها با دریافت بازخورد از کاربران ارزیابی می‌شوند و در صورت نیاز، تغییرات اعمال می‌شود.

۲-معرفی ذی‌نفعان پروژه:

ذی‌نفعان پروژه افراد یا سازمان‌هایی هستند که از این پلتفرم بهره‌مند می‌شوند. این شامل پیمانکاران، کارفرمایان، مدیران ارشد شرکت‌ها و شرکت‌های تولید کننده ابزارهای ساختمانی است.

در نتیجه، با توجه به این گزارش، ایجاد پلتفرم تعاملی برای ساخت و ساز با توجه به نیازمندی‌ها و بازخوردهای ذی‌نفعان، یک گام مهم برای بهبود فرآیند ارتباطی و همکاری در این صنعت است.

پلتفرم آنلاین مدیریت جامع تعاملات میان کارفرما و پیمانکار

در این مطلب، به بررسی ایجاد بستری برای تعامل میان کارفرما و پیمانکار پرداخته می‌شود که بستر مناسبی برای نمایش شغل خود ندارند.

۱-فعالیت‌های تحلیل سیستم پروژه:

جمع‌آوری اطلاعات: در این مرحله، اطلاعات مورد نیاز برای طراحی و پیاده‌سازی پروژه جمع‌آوری می‌شود. از روش‌های مختلف مانند مصاحبه، استفاده از فرم‌های آماده و بررسی پروژه‌های مشابه استفاده می‌شود.

تعريف نیازمندی‌ها: نیازمندی‌های پروژه بر اساس اطلاعات جمع‌آوری شده و ایده‌های مطرح شده تعریف می‌شود. این نیازمندی‌ها به دو دسته عملکردی و غیرعملکردی تقسیم می‌شوند.

اولویت‌بندی نیازمندی‌ها: نیازمندی‌ها بر اساس



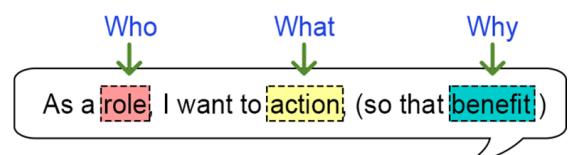
۲. مثال‌های داستان کاربری:

- به عنوان مشتری، من می‌خواهم پشتیبانی کارت بانکی داشته باشم تا بتوانم به راحتی آنلاین خرید انجام دهم.
- به عنوان مدیر، من گزارشی می‌خواهم تا بدامن کدام یک از دیارتمان‌ها نیاز به منابع بیشتری دارند.
- به عنوان مشتری، من می‌خواهم هنگامی که محصول به مقصد میرسد، از طریق پیامک مطلع شوم تا بتوانم سریع‌تر محصول را دریافت کنم.

ضوابط پذیرش: ضوابط پذیرش نیز ویژگی‌های کارکردی مشخصی را برای کسب رضایت کاربر

داستان کاربری: کلیدی برای توسعه نرم‌افزار موفق

۱. داستان کاربری و ضوابط پذیرش در فرآیند توسعه و مدیریت محصول نرم‌افزاری، داستان کاربر به عنوان توصیفی طبیعی و غیررسمی از یک یا چند خصوصیت سیستم نرم‌افزاری شناخته می‌شود. این داستان‌ها کمک می‌کنند تا نیازها به صورت ساده‌تری شناسایی شده و درک شود. این داستان‌ها معمولاً در جایی ثبت می‌شوند و ممکن است توسط افراد مختلفی مانند کاربران، مشتریان، مدیران یا اعضای تیم توسعه نوشته شوند. زبان داستان کاربران امروزی و هدفمند است و به دنبال رسیدن به یک هدف خاص می‌باشد.



میخواهم تا جستجوی دقیقی در آگهی‌های کاری انجام دهم تا به سریع‌ترین و راحت‌ترین شکل ممکن پروژه مناسب با مهارت خود را پیدا کنم.

ضوابط پذیرش:

- وجود بخش جستجوی پیشرفته در قسمت بالای برنامه.
- امکان فیلتر کردن جستجو بر اساس موضوع پروژه، قیمت و مهارت‌های مورد نیاز.
- نمایش آگهی‌های پیشنهادی بر اساس جستجوهای قبلی.
- امکان جستجو بر اساس موقعیت مکانی.
- ۲. تکنیک هدف کاربر در این تکنیک، نیازهای کارکردی پروژه با دسته‌بندی کاربران سیستم و لیست اهداف آنها در ارتباط با سیستم مشخص می‌شود.

تکنیک هدف کاربر بروزی پروژه تعریف شده:

- ۳. تکنیک تجزیه رخداد در این تکنیک، رخدادهایی که در زمان و مکان مشخصی رخ می‌دهند و سیستم باید به آنها پاسخ دهد، تشخیص داده می‌شوند و به سه دسته رخداد خارجی، زمانی، و حالتی تقسیم می‌شوند.



تعیین می‌کند که باید در زمان انجام وظیفه برقرار باشد.

دادستان‌های کاربری پروژه تعریف شده

(الف) داستان کاربری: به عنوان یک کارفرما، میخواهم تا اطلاعات پیمانکاران را به طور کامل مشاهده کنم تا بتوانم به راحتی از میان آنها گزینه مناسبی را انتخاب کنم.

ضوابط پذیرش:

- ثبت تمام مهارت‌ها و تجربیات پیمانکاران.
- مشاهده نظرات کارفرماهای قبلی درباره تجربه کاری با آن‌ها.
- مشاهده امتیاز کلی پیمانکار.
- امکان ارتباط با پیمانکار برای بحث در مورد موارد مختلف.

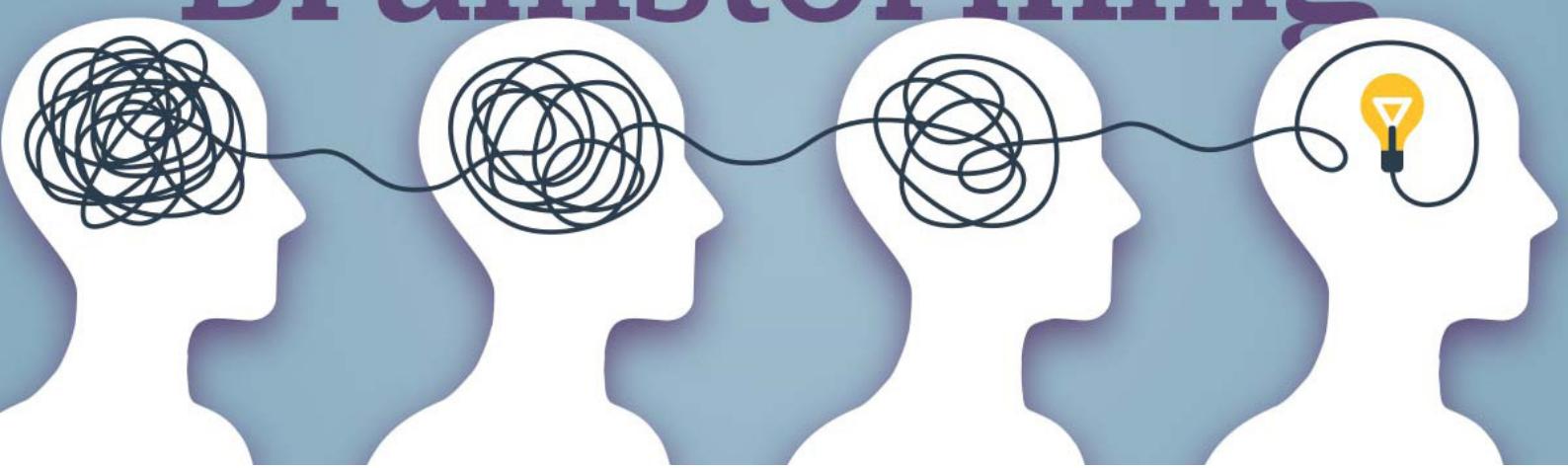
(ب) داستان کاربری: به عنوان یک کارفرما، میخواهم تا یک آگهی با تمام جزئیات کاری ایجاد کنم تا پیمانکار به طور کامل با پروژه آشنا شود.

ضوابط پذیرش:

- شرح کامل پروژه درخواستی همراه با مهارت‌های مورد نیاز.
- ضمیمه کردن تصویر به پروژه.
- ثبت مدت زمان مورد نیاز برای انجام پروژه.
- اضافه کردن بازه قیمت پیشنهادی برای پروژه.
- ثبت حداقل تعداد افرادی که می‌توانند در پروژه شرکت کنند.

(پ) داستان کاربری: به عنوان یک پیمانکار،

Brainstorming



برای کار، و هزینه پیشنهادی در سیستم ذخیره می‌شود. برای این پروژه اشیای مرتبط را می‌توان به صورت زیر دسته‌بندی کرد:

- کارفرما
- پیمانکار
- آگهی‌های کاری به همراه جزئیات
- قراردادهای کاری به همراه جزئیات
- رتبه‌بندی کارفرما و پیمانکار
- تسویه حساب قرارداد
- ثبت شکایت
- ثبت تیکت برای پشتیبانی
- ارسال ایمیل و اعلان‌ها
- ...

تکنیک اسم

این تکنیک برای تشخیص کلاس‌های دامنه با استفاده از لیست اسامی مورد استفاده در مستندات و بحث‌ها طراحی شده است.

"روش‌های تحلیل و طراحی سیستم: از تکنیک بارش فکری تا تکنیک اسم"

تکنیک بارش فکری

در این تکنیک، از چک لیستی از تمامی اشیائی که در سیستم وجود دارد استفاده می‌شود. با استفاده از طوفان فکری، کلاس‌های دامنه هر یک از این اشیاء تشخیص داده می‌شود.

قدم‌های تکنیک طوفان فکری:

۱. تشخیص یک کاربر و مجموعه‌ای از موارد کاربرد.
۲. انجام طوفان فکری با کاربر برای تشخیص اشیاء مورد استفاده در هنگام انجام موارد کاربرد.
۳. استفاده از دسته‌بندی اشیاء برای پرسش سوالات سیستماتیک در مورد اشیاء بالقوه.

طوفان فکری در پروژه معرفی شده:

در این پروژه، زمانی که کارفرما قصد دارد یک آگهی کاری ثبت کند، اطلاعاتی مانند کارفرما، شرح آگهی، مکان و زمان کار، تعداد نیروهای مورد نیاز

۲. برنامه‌ریزی پروژه:

- مصاحبه با کارفرمایان و پیمانکاران: برای جمع‌آوری اطلاعات مورد نیاز.
- طراحی دیتابیس و بخش ظاهري سیستم: بر اساس نیازها و الزامات کاربری.

۳. کشف و درک جزئیات مسئله یا نیاز:

- مورد کاربرد و استفاده از نمودارهای کلاسی: برای بهترین درک مسئله و نیازهای کاربران.

۴. طراحی اجزای سیستم:

- طراحی دیتابیس و کلاس‌های برنامه مطابق با نیازها و اطلاعات مورد نیاز.

۵. ساخت، تست و یکپارچه‌سازی اجزای سیستم:

- پیاده‌سازی بخش‌های مختلف سیستم و انجام تست‌های مورد نیاز.

۶. اتمام تست سیستم و به‌کارگیری:

- انجام تست‌های عملیاتی و به‌کارگیری سیستم در محیط واقعی.



در این مطلب، به بررسی ضرورت ایجاد پلتفرمی جامع برای تسهیل تعاملات میان کارفرما و پیمانکار و نقش روش‌های تحلیل و طراحی سیستم در توسعه آن پرداختیم. با تمرکز بر داستان کاربری به عنوان ابزاری کلیدی برای درک نیازهای کاربران، و به کارگیری تکنیک‌های مختلف تحلیل و طراحی سیستم، می‌توانیم سیستمی کارآمد و کاربرپسند ایجاد کنیم که به طور موثری به نیازهای ذینفعان پاسخ دهد.

چرخه حیات توسعه زیرسیستم عقد قرارداد بین پیمانکار و کارفرما

این چرخه حیات، تمامی فعالیت‌های مورد نیاز برای ساخت، بهره‌برداری، و نگهداری یک سیستم اطلاعاتی را شامل می‌شود. این شامل ۶ فرآیند اصلی زیر است:

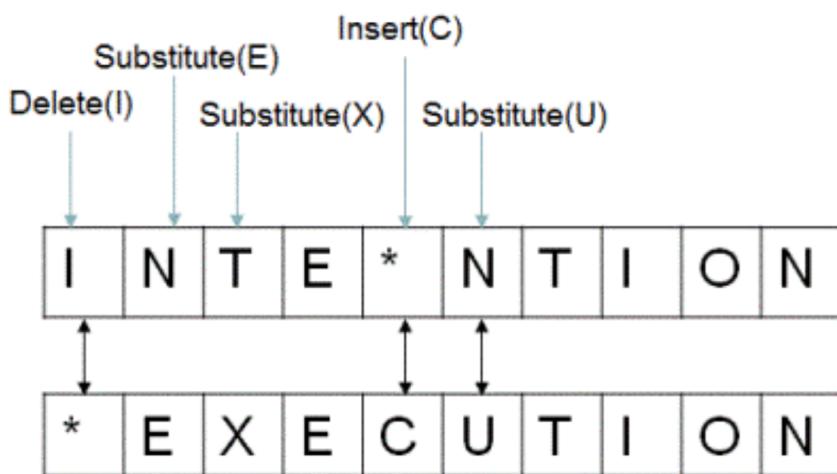
۱. تشخیص مسئله یا نیاز و اخذ موافقت
۲. برنامه‌ریزی و کنترل پروژه
۳. کشف و درک جزئیات مسئله یا نیاز
۴. طراحی اجزای سیستم که مسئله را حل می‌کنند
۵. ساخت، آزمون، و یکپارچه‌سازی اجزای سیستم
۶. اتمام آزمون‌های سیستم و اعمال راهکارها

مثال اجرایی:

۱. تشخیص مسئله یا نیاز و اخذ موافقت:

- **توصیف مسئله:** ساخت یک قرارداد بین پیمانکار و کارفرما برای انجام کار با شرایط مشخص.
- **قابلیت‌های سیستم:** ذخیره تمام جزئیات قرارداد از جمله شرح کار، مشخصات طرفین، تعداد نیروهای مشارکت‌کننده، زمان و هزینه.
- **مزایای تجاری:** ایجاد امنیت و نظم در انجام پروژه‌ها، تسهیل در فرآیند انجام کارها.

EDIT DISTANCE



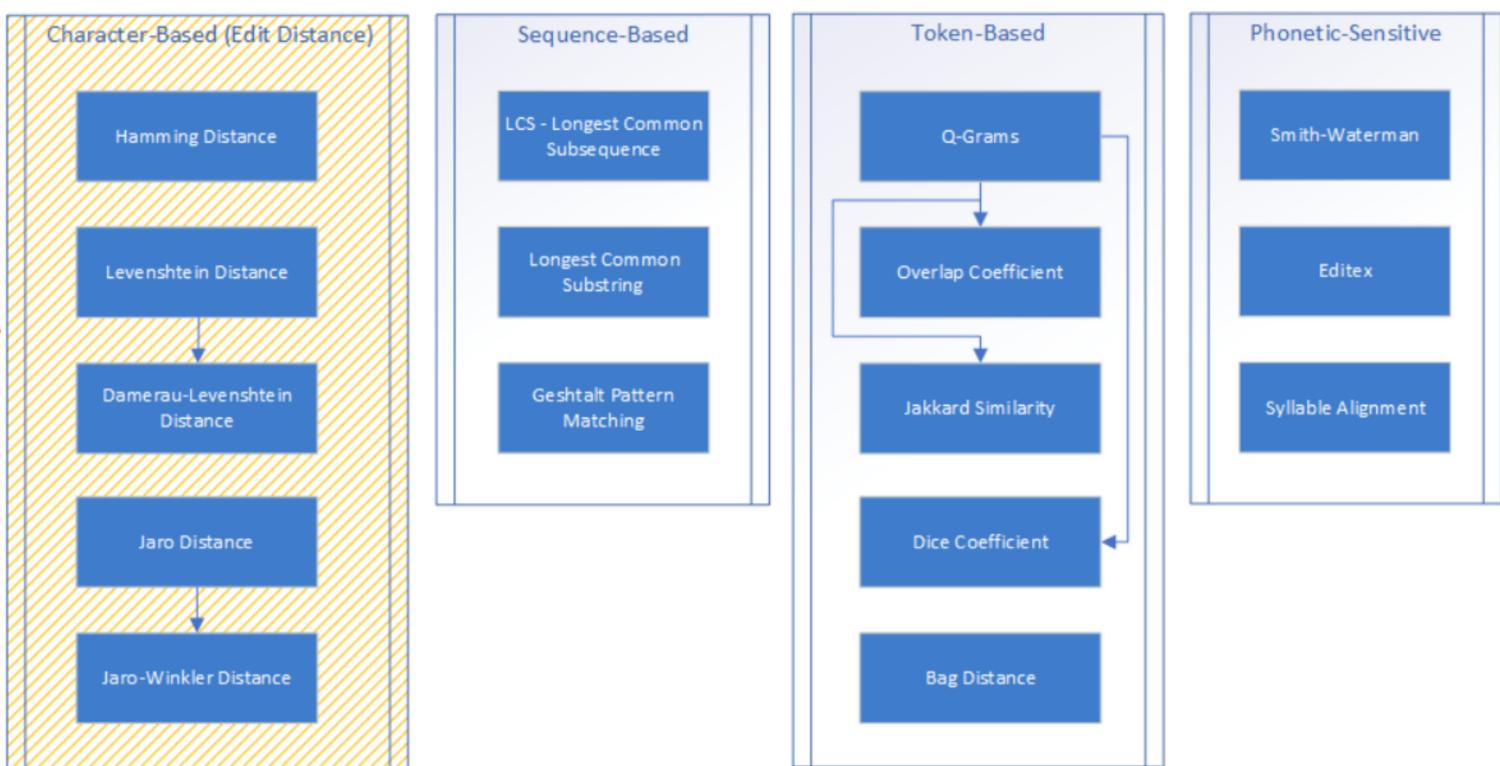
نویسنده: سارا خسروزاده و معین رضازاده

بخش ۱

فاصله ویرایشی

در دنیای زبان‌شناسی محاسباتی و علوم رایانه، مفهومی به نام «فاصله ویرایشی» وجود دارد که به عنوان یک معیار برای سنجش میزان شباهت بین دو رشته (مانند کلمات) عمل می‌کند. این کار با شمارش حداقل تعداد عملیات ویرایشی که برای تبدیل یک رشته به رشته دیگر لازم است، انجام می‌شود.

در این مطلب به بررسی روش‌های مختلف Edit Distance می‌پردازیم. در این بخش، روش‌های مبتنی بر کاراکتر و توالی را شرح می‌دهیم و نحوه عملکرد آنها را توضیح می‌دهیم.



کاراکتر را به عنوان یک موجودیت جداگانه در نظر می‌گیرد.

فاصله Hamming

فاصله همینگ (Hamming Distance) یک معیار بنیادی برای مقایسه رشته‌هایی با طول مساوی است. این معیار، عدم شباهت بین دو رشته را با شمارش تعداد موقعیت‌هایی که نمادهای متناظر در آنها متفاوت هستند، کمیت می‌کند. به عبارت دیگر، فاصله همینگ حداقل تعداد تعویض‌هایی را که برای تبدیل یک رشته به رشته دیگر لازم است، اندازه‌گیری می‌کند.

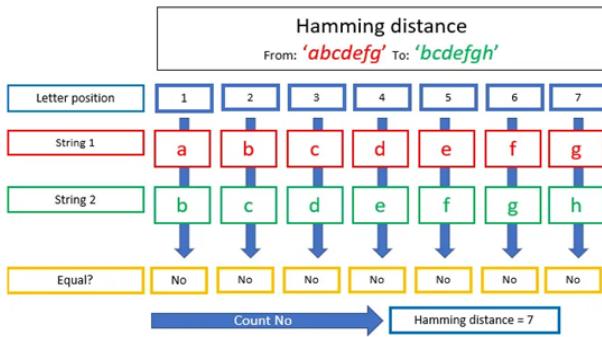
فاصله ویرایشی به ۱۴ قسمت اصلی تقسیم می‌شود:

- Character-Based
- Sequence-Based
- Token-Based
- Phonetic-Sensitive

۱ فاصله ویرایشی

فاصله ویرایش مبتنی بر کاراکتر دو رشته را در سطح کاراکترهای فردی مقایسه می‌کند. تعداد درج‌ها، حذف‌ها یا تعویض‌های مورد نیاز برای تبدیل یک رشته به رشته دیگر را می‌شمارد و هر

مثال:



فاصله Levenshtein

فاصله لونشتاین (Levenshtein Distance) مانند فاصله همینگ، معیاری برای سنجش کمترین تعداد عملیات ویرایشی است که برای تبدیل یک رشته به رشته دیگر لازم است. برخلاف فاصله همینگ که فقط به تعداد نمادهای متفاوت در دو رشته توجه می‌کند و ترتیب قرارگیری آنها را در نظر نمی‌گیرد، فاصله لونشتاین به طور کامل به ترتیب و نوع نمادها نیز توجه می‌کند.

ویژگی‌های فاصله لونشتاین

- همه کاره بودن
- فاصله لونشتاین ابزاری همه کاره است که می‌تواند رشته‌هایی با طول‌های مختلف را مقایسه کند. این قابلیت به دلیل در نظر گرفتن عملیات درج، حذف و جایگزینی کاراکترها در رشته‌ها امکان‌پذیر است. به همین دلیل، فاصله لونشتاین برای طیف وسیعی از کاربردها از جمله پردازش زبان طبیعی، رمزگاری و تصویح غلط املایی مناسب می‌باشد.

- برنامه‌نویسی پویا
- محاسبه فاصله لونشتاین بین دو رشته اغلب از طریق الگوریتم‌های برنامه‌نویسی پویا انجام می‌شود. در این روش، از یک

فرض کنید می‌خواهیم فاصله همینگ بین دو رشته «۱۰۰۰» و «۱۱۰۱» را محاسبه کنیم. در این دو رشته، نمادها در موقعیت‌های دوم و چهارم با یکدیگر متفاوت هستند، بنابراین، فاصله همینگ بین این دو رشته برابر با ۲ خواهد بود.

کاربردهای فاصله همینگ:

مهمترین کاربرد فاصله همینگ در تشخیص و تصویح خطأ است. این معیار به طور گسترده در کدهای تصویح خطأ، مانند کدهای همینگ، مورد استفاده قرار می‌گیرد. در این کدها، از فاصله همینگ برای شناسایی و تصویح خطأ در داده‌های باینری استفاده می‌شود.

مزایای استفاده از فاصله همینگ:

پیچیدگی محاسباتی ساده: فاصله همینگ دارای پیچیدگی محاسباتی $O(n^2)$ است، که در آن n طول رشته‌های مقایسه شده است. این پیچیدگی زمانی خطی، محاسبه فاصله همینگ را برای پردازش حجم عظیمی از داده‌ها به روی کارآمد تبدیل می‌کند. کاربردهای متعدد: فاصله همینگ علاوه بر کاربرد در کدهای تصویح خطأ، در زمینه‌های مختلفی مانند فشرده‌سازی داده‌ها، رمزگاری، و بازیابی اطلاعات نیز مورد استفاده قرار می‌گیرد.

فاصله Hamming چگونه کار می‌کند؟

فاصله همینگ هر حرف از دو رشته را بر اساس موقعیت مقایسه می‌کند. بنابراین حرف اول کلمه ۱ با حرف اول کلمه ۲ و غیره مقایسه می‌شود.

تصحیح خطای برای تصحیح غلطهای املایی و نگارشی در متن.

ترجمه ماشینی: برای ترجمه متن از یک زبان به زبان دیگر.

بازیابی اطلاعات: برای یافتن اسنادی که به یک پرس و جو خاص مرتبط هستند.

پیچیدگی زمانی الگوریتم فاصله لونشتاین

پیچیدگی زمانی الگوریتم فاصله لونشتاین $O(m * n)$ است، که در آن m و n طول دو رشته مورد مقایسه هستند. این به این معنی است که زمان اجرای الگوریتم با طول رشته‌ها به صورت تصاعدی افزایش می‌یابد.

پیچیدگی فضایی الگوریتم فاصله لونشتاین

پیچیدگی فضایی الگوریتم فاصله لونشتاین نیز $O(m * n)$ است. دلیل این امر نیاز به ذخیره کل ماتریس محاسبه شده در طول الگوریتم است. این ماتریس ابعاد m در n دارد و هر سلول آن نشان‌دهنده فاصله بین زیررشته‌های طول‌های مختلف دو رشته اصلی است.

نحوه محاسبه فاصله لونشتاین

با توجه به دو رشته، فاصله لونشتاین بین آنها حداقل تعداد ویرایش‌های تک کاراکتری (درج، حذف، یا جایگزینی) مورد نیاز برای تغییر یک رشته به رشته دیگر است.

درج: اضافه کردن یک کاراکتر به رشته.

حذف: حذف یک کاراکتر از رشته.

جایگزینی: تغییر یک کاراکتر در رشته.

ماتریس برای ذخیره‌سازی هزینه تبدیل هر زیررشته از یک رشته به رشته دیگر استفاده می‌شود. این ماتریس به طور گام به گام با استفاده از معادلات مربوط به عملیات درج، حذف و جایگزینی کاراکترها پر می‌شود.

کاربردهای فاصله لونشتاین

بررسی املای کلمات

فاصله لونشتاین به طور گسترده در الگوریتم‌های بررسی املای کلمات برای پیشنهاد اصلاحات برای کلمات غلط املایی استفاده می‌شود. در این الگوریتم‌ها، فاصله لونشتاین بین کلمه غلط املایی و کلمات موجود در دیکشنری محاسبه می‌شود و کلماتی که کمترین فاصله را با کلمه مورد نظر دارند به عنوان کاندیدای تصحیح ارائه می‌شوند.

تحلیل توالی‌های DNA

در بیوانفورماتیک، از فاصله لونشتاین برای مقایسه توالی‌های DNA و شناسایی شباهت‌ها یا تفاوت‌ها بین آن‌ها استفاده می‌شود. این امر می‌تواند در مطالعات مختلف ژنتیکی مانند یافتن جهش‌ها، مقایسه ژن‌ها بین گونه‌های مختلف و ردیابی تکامل گونه‌ها مفید باشد.

پردازش زبان طبیعی (NLP)

فاصله لونشتاین یک معیار اساسی در وظایف مختلف پردازش زبان طبیعی است. از جمله کاربردهای آن در این حوزه می‌توان به موارد زیر اشاره کرد:

سنجهش شباهت متن: برای مقایسه دو متن و تعیین میزان شباهت آن‌ها به یکدیگر.

فاصله لونشتاین بین دو رشته "kitten" و "sitting" برابر با ۳ است که یک عدد غیر منفی است.

- متقارن: فاصله لونشتاین بین دو رشته A و B برابر با فاصله لونشتاین بین دو رشته B و A است. به عبارت دیگر، اگر میخواهیم بدانیم که رشته A چقدر شبیه به رشته B است، فرقی نمی‌کند که ابتدا کدام رشته را بررسی کنیم.

- ارضای نابرابری مثلث: برای هر سه رشته A، B و C، نابرابری مثلث زیر برقرار است:

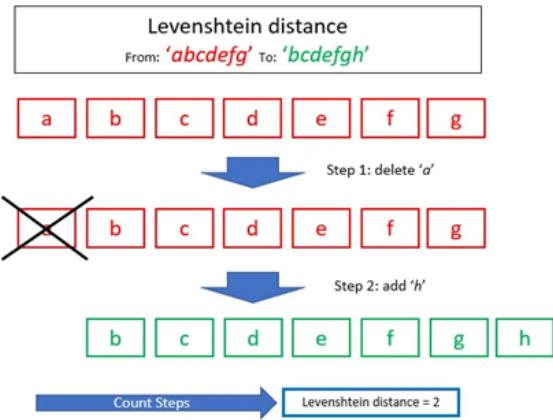
$$d(A, B) \leq d(A, C) + d(C, B)$$

این بدان معناست که اگر A و B تا حدی شبیه به یکدیگر باشند و B و C تا حدی شبیه به یکدیگر باشند، A و C نیز باید تا حدی شبیه به یکدیگر باشند.

محاسبه فاصله لونشتاین، علی‌رغم مزایایی که دارد، می‌تواند پرهزینه باشد. با وجود این چالش، می‌توان از فاصله لونشتاین به عنوان معیار تشابه بین دو رشته استفاده کرد. برای این منظور، می‌توان فاصله لونشتاین را با استفاده از فرمول زیر به یک عدد نرمال شده بین ۰ و ۱ تبدیل کرد:

$$\text{sim}_{ld}(a, b) = 1.0 - \frac{\text{dist}(a, b)}{\max(|a|, |b|)}$$

Levenshtein پر کردن ماتریس در الگوریتم فاصله حلقه دوگانه با استفاده از برنامه نویسی - پویا، ماتریس dp را پر می‌کند.



در حالی که فاصله همینگ فقط میتواند تفاوت‌های نقطه‌ای بین دو رشته را نشان دهد، فاصله لونشتاین به دلیل در نظر گرفتن عملیات ویرایشی مانند درج، حذف و جایگزینی، درک شهودی بهتری از شباخت بین دو رشته ارائه می‌دهد. به عنوان مثال، فاصله لونشتاین بین دو رشته abcdefg و bcdefgh که در فاصله همینگ کاملاً متفاوت هستند، ۱ است.

همچنین، فاصله لونشتاین بین دو رشته GRATE و GIRAFFE برابر با ۳ است، زیرا می‌توان با انجام سه عملیات ویرایشی زیر، یک رشته را به دیگری تبدیل کرد:

	G	R	A	T	E		
Insertion (I)	G	I	R	A	T	E	
Substitution (T->F)	G	I	R	A	F	E	
Insertion (E)	G	I	R	A	F	F	E

فاصله لونشتاین و الزامات متریک مطابقت

فاصله لونشتاین به طور کلی الزامات یک متریک مطابقت را برآورده می‌کند، به این معنی که خواص زیر را دارا است:

- غیر منفی: فاصله لونشتاین بین دو رشته همیشه غیر منفی است. به عنوان مثال،

Damerau-Levenshtein متریک فاصله متریک است، به این معنی که فاصله رشته A تا B با فاصله رشته B تا A برابر است.

کاربردهای فاصله Damerau-Levenshtein

- تشخیص نوری حروف (OCR) : فاصله Damerau-Levenshtein در کاربردهای OCR که جابجایی ها خطاهای رایجی هستند، مفید است.
- تصحیح املایی: از این فاصله در الگوریتم های تصحیح املایی استفاده میشود که علاوه بر درج ها، حذف ها و جایگزین ها، باید حروف جابجا شده را نیز در نظر بگیرند.

پیچیدگی محاسباتی فاصله Damerau-Leven-shtain

مشابه فاصله Levenshtein، پیچیدگی محاسباتی این فاصله $O(m * n)$ است، که در آن m و n طول رشته های مقایسه شده هستند.

نحوه محاسبه فاصله Damerau-Levenshtein

به عنوان مثال، فاصله Levenshtein بین GIFT و FIT ، ۳ است:

	G	I	F	T
Deletion (G)		I	F	T
Substitution (I->F)		F	F	T
Substitution (F->I)	F	I	T	

در این مثال، برای تبدیل IF به FI به دو مرحله نیاز داریم. با این حال، اگر اجازه جابه جایی بین دو کاراکتر مجاور را بدھیم، میتوانیم تنها از یک

- موارد پایه (سطر اول و ستون اول) هزینه تبدیل یک رشته خالی به یک رشته غیر خالی یا بالعکس را نشان می دهد.
- برای هر سلول (j,i)، الگوریتم سه احتمال را در نظر می گیرد: حذف، درج و جایگزینی.
- هزینه هر عملیات بر اساس مقادیر موجود در سلول های همسایه است.

فاصله Levenshtein مقدار ذخیره شده در گوشه پایین سمت راست ماتریس ($dp[m][n]$) است.

Damerau-Levenshtein

مشاهده شده است که اکثر اشتباهات املایی انسان در چهار نوع خطا جای میگیرند: درج، حذف، جایگزینی و جابجایی. بر این اساس، فاصله Damerau-Levenshtein مجموعه عملیات ویرایشی مجاز در فاصله Levenshtein را با اضافه کردن جابجایی دو کاراکتر مجاور گسترش می دهد.

به عبارتی دیگر، میتوان گفت که فاصله Damer-Leven-shtain گسترش متريک فاصله Levenshtein است. اين فاصله حداقل تعداد عملیات ویرایشی (شامل درج، حذف، جایگزینی و جابجایی کاراکترهای مجاور) مورد نیاز برای تبدیل یک رشته به رشته دیگر را اندازه گیری میکند.

ویژگی های فاصله Damerau-Levenshtein

- گسترش جابجایی ها: در مقایسه با فاصله Damerau-Levenshtein، فاصله Damerau-Levenshtein شامل یک عملیات اضافی، جابجایی، است. جابجایی شامل تعویض دو کاراکتر مجاور است.

راه حل های جایگزین

برای غلبه بر این چالش ها، روش های مختلفی برای محاسبه صحیح فاصله Damerau-Leven- shtain ارائه شده است:

- روش جدول: این روش شامل ایجاد یک جدول است که در آن هر سلول هزینه تبدیل پیشوندهای مختلف هر دو رشته را ذخیره می کند. سپس از این جدول برای محاسبه فاصله Damerau-Levenshtein بین دو رشته کامل استفاده می شود.
- روش بازگشتی: این روش از الگوریتم بازگشتی برای محاسبه فاصله Dam- erau-Levenshtein مختلف دو رشته استفاده می کند.
- روش برنامه نویسی پویا اصلاح شده: این روش شامل اصلاح رویکرد برنامه نویسی پویا سنتی برای در نظر گرفتن عملیات جابجایی است.

پر کردن ماتریس در الگوریتم فاصله Damer- au-Levenshtein

مشابه الگوریتم فاصله Levenshtein، الگوریتم فاصله Damerau-Levenshtein نیز از برنامه نویسی پویا برای پر کردن ماتریس dp استفاده می کند. این ماتریس مقادیر مربوط به فاصله های جزئی بین زیرشته های مختلف دو رشته اصلی را ذخیره می کند.

در الگوریتم فاصله Damerau-Levenshtein، علاوه بر عملیات درج، حذف و جایگزینی که در الگوریتم فاصله Levenshtein وجود دارد، عملیات جابجایی کاراکترهای مجاور نیز بررسی می شود.

مرحله برای تبدیل استفاده کنیم. بنابراین، فاصله Damerau-Levenshtein بین GIFT و FIT است:

	G	I	F	T
Deletion (G)		I	F	T
Transposition (IF->FI)	F	I		T

گنجاندن جابجایی در الگوریتم فاصله لونشتاین

گنجاندن عملیات جابجایی در الگوریتم های اصلی محاسبه فاصله لونشتاین میتواند چالش برانگیز باشد. دلیل این امر این است که اصلاح مستقیم رویکرد برنامه نویسی پویا که برای محاسبه فاصله لونشتاین استفاده می شود، با ورودی های مربوط به جابجایی به طور غیرمنتظره ای "فاصله تراز بهینه Rشته" را به جای فاصله Damerau-Levenshtein محاسبه می کند.

فاصله تراز بهینه Rشته، حداقل تعداد عملیات ویرایشی (شامل درج، حذف، جایگزینی و جابجایی) مورد نیاز برای تراز کردن دو رشته به گونه ای است که حداقل تعداد کاراکترهای همسان در ستون ها قرار گیرند. در حالی که این معیار تا حدی شبیه به فاصله Damerau-Levenshtein است، اما دو تفاوت کلیدی وجود دارد:

- محاسبه نادرست: فاصله تراز بهینه Rشته همیشه با فاصله shtain برابر نیست.
- عدم ارضای نابرابری مثلث: فاصله تراز بهینه Rشته خاصیت نابرابری مثلث را که یکی از الزامات اساسی برای یک معیار سنجش شباهت رشته است، نقض می کند.

فرآیندهای حذف سوابق تکراری برای یافتن و ادغام رکوردهای مشابه در یک مجموعه داده استفاده می‌شود.

پیچیدگی محاسباتی فاصله Jaro

پیچیدگی زمانی برای محاسبه فاصله Jaro، $O(n^2)$ است جایی که n طول رشته کوتاهتر است.

نحوه محاسبه Jaro Distance

فاصله‌ی Jaro (J) بین دو رشته s_1 و s_2 با فرمول زیر قابل محاسبه است:

$$J = \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right)$$

در این فرمول:

m تعداد کاراکترهای همسان بین دو رشته است.

t نصف تعداد جابجایی‌ها است.

مراحل محاسبه:

- یافتن کاراکترهای همسان: رشته‌ها را از ابتدا تا انتها اسکن کنید و کاراکترهای مشترک را در فاصله حداقل t از یکدیگر شناسایی کنید. هر کاراکتر همسان فقط یک بار شمارش می‌شود.

- محاسبه تعداد جابجایی‌ها: برای هر کاراکتر همسان در رشته اول، بررسی کنید که آیا در رشته دوم در همان موقعیت قرار دارد یا خیر. اگر در همان موقعیت نباشد، یک جابجایی رخ داده است.
- محاسبه L : مقادیر را در فرمول L جایگزین کنید.

با استفاده از این چهار حالت، می‌توان مقادیر مربوط به تمام سلول‌های ماتریس dp را محاسبه کرد. در نهایت، مقدار موجود در سلول $[dp][m][n]$ که در آن m و n طول رشته‌های اصلی را نشان می‌دهند، فاصله Damerau-Levenshtein بین دو رشته را نشان می‌دهد.

فاصله Jaro

فاصله Jaro یک معیار سنجش شباهت است که برای مقایسه بین دو رشته به کار می‌رود. این معیار شباهت را بر اساس تعداد کاراکترهای همسان و جابجایی کاراکترها در دو رشته محاسبه می‌کند.

فاصله Jaro عددی بین ۰ و ۱ را به عنوان خروجی ارائه می‌دهد. در این مقیاس، ۰ نشان‌دهنده عدم شباهت کامل و ۱ نشان‌دهنده تطابق کامل دو رشته با یکدیگر است.

ویژگی‌های فاصله Jaro

- در نظر گرفتن کاراکترهای همسان: فاصله Jaro تعداد کاراکترهای مشترک و همسان بین دو رشته را شمارش می‌کند.
- جریمه برای جابجایی‌ها: این معیار جابجایی‌ها (تغییر در ترتیب کاراکترهای مجاور) را در رشته‌ها جریمه می‌کند.

Jaro Distance

- پیوند رکوردها: فاصله Jaro به طور معمول در وظایف پیوند رکوردها استفاده می‌شود. این وظایف به دنبال شناسایی رکوردهای مشابه در مجموعه داده‌های مختلف هستند.
- حذف سوابق تکراری: از این معیار در

فاصله Jaro-Winkler

این عامل برای افزایش امتیاز شباهت رشته‌هایی که پیشوند مشترک دارند، به کار می‌رود.

Jaro-Winkler Distance

- پیوند رکورد فاصله Jaro-Winkler معمولاً در کارهای پیوند رکورد که در آن رشته‌ها ممکن است پیشوندهای مشترکی مانند نام و آدرس داشته باشند، استفاده می‌شود.
- پاکسازی داده‌ها فاصله Jaro-Winkler همچنین در فرآیندهای پاکسازی داده‌ها و حذف رکوردهای تکراری کاربرد دارد. این امر به دلیل توانایی آن در شناسایی رشته‌هایی که از نظر معنایی مشابه هستند، حتی اگر تفاوت‌های جزئی در املای یا نگارش داشته باشند، بسیار مفید است.

پیچیدگی محاسباتی فاصله Jaro-Winkler

پیچیدگی زمانی برای محاسبه فاصله Jaro-Winkler است جایی که n طول رشته کوتاه‌تر است.

Jaro-Winkler Distance

فاصله Jaro-Winkler برای اندازه‌گیری شباهت از مقیاس پیشوندی p استفاده می‌کند. این مقیاس امتیازات مطلوب‌تری به رشته‌هایی که از ابتدا با یک طول پیشوند مشترک مطابقت دارند، می‌دهد. برای دو رشته s_1 و s_2 ، شباهت Jaro-Winkler به شکل زیر محاسبه می‌شود:

$$\text{sim}_w = \text{sim}_j + \ell p(1-\text{sim}_j)$$

• sim_j : شباهت Jaro بین دو رشته است.

فاصله Jaro-Winkler یک معیار سنجش شباهت بین دو رشته است که به عنوان یک نسخه اصلاح‌شده از فاصله Jaro معرفی شده است. هدف اصلی این معیار، اعطای وزن بیشتر به پیشوندهای مشترک بین رشته‌ها است.

نحوه عملکرد این معیار به این صورت است که ابتدا با استفاده از معیار Jaro، شباهت کلی بین دو رشته محاسبه می‌شود. سپس، با در نظر گرفتن طول پیشوندهای مشترک، یک امتیاز اضافی به این شباهت اولیه اضافه می‌شود.

در نهایت، فاصله Jaro-Winkler به صورت عددی بین ۰ و ۱ ارائه می‌شود. مقدار ۰ نشان‌دهنده عدم شباهت و مقدار ۱ نشان‌دهنده تطابق کامل بین دو رشته است.

خصوصیات فاصله Jaro-Winkler

- بهبود پیشوند مشترک در فاصله Jaro-Winkler

فاصله Jaro-Winkler برای تشخیص شباهت بین رشته‌ها، از عاملی به نام "بهبود پیشوند مشترک" استفاده می‌کند. این عامل، وزن بیشتری به پیشوندهای مشترک بین دو رشته می‌دهد و در نتیجه، رشته‌هایی که پیشوندهای مشابهی دارند، کمتر جریمه می‌شوند.

عامل مقیاس‌بندی قابل تنظیم

عامل مقیاس‌بندی، که با حرف " p " نشان داده می‌شود، معمولاً روی یک مقدار ثابت کوچک (مانند ۱٪) تنظیم می‌شود.

LCS کاربردهای

- مقایسه متن: برای یافتن شباهت‌ها بین اسناد یا کدهای مختلف
- ترجمه ماشینی: برای یافتن معادلهای کلمات یا عبارات در زبان‌های مختلف
- نسخه پشتیبان‌گیری از داده‌ها: برای یافتن تفاوت‌های بین دو نسخه از یک فایل
- زیست‌شناسی: برای مقایسه توالی‌های DNA یا پروتئین

LCS نحوه عملکرد

محبوب‌ترین الگوریتم برای یافتن LCS، الگوریتم برنامه‌نویسی پویا است. این الگوریتم با استفاده از یک جدول به نام "جدول LCS" که در آن هر سلول (j, i) طول LCS اولین i کاراکتر از رشته اول و اولین j کاراکتر از رشته دوم را نشان می‌دهد، کار می‌کند.

مثال:

LCS برای دو رشته‌ی "ABCD" و "ACDF" برابر است با : ".ACD".

طولانی‌ترین زیررشته مشترک

طولانی‌ترین زیررشته مشترک، طولانی‌ترین دنباله پیوسته از کاراکترها است که در هر دو دنباله داده شده ظاهر می‌شود.

Longest Common Substring نحوه عملکرد

مشابه LCS، اغلب از برنامه نویسی پویا استفاده می‌شود. ایده این است که جدولی بسازیم که در آن هر سلول (j, i) نشان دهنده طول رشته فرعی مشترک باشد که در موقعیت i برای دنباله اول و

- ℓ : طول پیشوند مشترک بین دو رشته است.
 - P : مقیاس پیشوند است.
 - $(1 - sim_j)$: این بخش نشان می‌دهد که دو رشته چقدر از نظر کاراکترهایشان غیرمشابه هستند.
- ملاحظات:
- مقدار p باید بین 0 و $1/25$ باشد.
 - مقدار پیش فرض p برابر با $1/4$ است.
 - طول پیشوند مشترک ℓ حداقل 4 کاراکتر است.

فاصله ویرایشی ۲

فاصله ویرایش مبتنی بر توالی، ترتیب و توالی کاراکترها را در رشته‌ها در نظر می‌گیرد. حداقل تعداد عملیات مورد نیاز برای تبدیل یک دنباله به دنباله دیگر را اندازه‌گیری می‌کند، جایی که یک دنباله می‌تواند هر مجموعه مرتب شده‌ای از آیتم‌ها باشد. به عنوان مثال، فاصله ویرایش مبتنی بر دنباله بین $[1, 2, 3, 4]$ و $[1, 3, 2, 4]$ خواهد بود، زیرا برای تطبیق دنباله‌ها به دو عملیات (تعویض عنصر دوم و سوم) نیاز است.

طولانی‌ترین دنباله مشترک (LCS)

طولانی‌ترین دنباله مشترک (LCS) دنباله‌ای از کاراکترها است که در هر دو رشته (رشته‌ها) به یک ترتیب ظاهر می‌شود و طول آن حداقل بین تمام دنباله‌های مشترک دو رشته است. به عبارت دیگر، LCS طولانی‌ترین رشته‌ای است که می‌توان از هر دو رشته با حذف صفر یا چند کاراکتر از هر رشته به دست آورد.



اصول روانشناسی گشتالت است.

این یک مفهوم انتزاعی تر و ذهنی تر در مقایسه با LCS و زیر رشته رایج است. به جای جستجوی تطابق دقیق، تطبیق الگوی گشتالت ممکن است شامل شناسایی الگوهای سطح بالاتر، مانند اشکال، ترتیبات، یا روابط بین عناصر باشد.

Gestalt Pattern Matching

در داده‌های سری زمانی که قیمت سهام را نشان می‌دهد، یک الگوی گشتالت ممکن است شامل شناسایی روندها، چرخه‌ها یا الگوهای تکرار شونده باشد که لزوماً با یک دنباله ثابت نشان داده نمی‌شوند.

پیچیدگی محاسباتی فاصله تطبیق الگوی گشتالت زمان اجرای الگوریتم تطبیق الگوی گشتالت در

در موقعیت ز برای دنباله دوم ختم می‌شود.

مثال:

Longest Common Substring برای رشته های "ABCBA" و "BABCA" برابر است با: ABC

ABABC
BABCA
ABCBA

Gestalt Pattern Match (ing)

تطبیق الگوی گشتالت شامل شناسایی الگوها یا ساختارهایی در یک توالی است که از نظر ادراکی یا مفهومی معنادار هستند، که اغلب بر اساس



در این بخش، به بررسی روش‌های مبتنی بر کاراکتر و دنباله برای محاسبه فاصله ویرایشی پرداختیم. در بخش بعدی، به سراغ روش‌های مبتنی بر توکن Phonetic-Sensitive می‌رویم و جزئیات این روش‌ها را بررسی خواهیم کرد. برای دسترسی به کدهای هر یک از روش‌های فاصله‌ی ویرایشی کد QR زیر را اسکن کنید:



<https://github.com/UUCESSC/Edit-Distance>

بدترین حالت برابر با $O(n^3)$ و در حالت متوسط برابر با $O(n^2)$ است (n طول یکی از رشته‌های ورودی است).

این پیچیدگی به این معنا است که با افزایش طول رشته‌های ورودی، زمان اجرای الگوریتم به سرعت (در حالت بدترین سناریو به صورت توان سومی) افزایش می‌یابد.

نحوه عملکرد Gestalt Pattern Matching

شماحت بین دو رشته S_1 و S_2 با استفاده از فرمول زیر محاسبه می‌شود. این فرمول دو برابر تعداد کاراکترهای همسان (Km) را بر مجموع تعداد کاراکترهای هر دو رشته تقسیم می‌کند.

$$D_{r_n} = \frac{2k_m}{|s_1| + |s_2|}$$

در این فرمول، معیار شماحت میتواند بین صفر و یک باشد که مقدار ۱ نشان دهنده تطابق کامل دو رشته و مقدار ۰ به معنی عدم وجود تطابق است.

فرض کنید میخواهیم شماحت بین دو رشته WI-KIMANIA و WIKIMANIA را بررسی کنیم:

S_1	W	I	K	I	M	E	D	I	A
S_2	W	I	K	I	M	A	N	I	A

$$\frac{2 \cdot (5 + 2)}{9 + 9} = \frac{14}{18} = 0.7$$

امتیاز ۰.۷ نشان دهنده شماحت قابل توجهی بین دو رشته است.

حافظه‌ها

نویسنده: آریا رافع

بخش ۱





حافظه چیست؟

در علوم کامپیوتر به هر واحدی که بتواند عمل ذخیره سازی را انجام دهد، حافظه می‌گویند. یک حافظه میباشد قابلیت نوشتن، حفظ کردن و خواندن را داشته باشد. در کامپیوتر چندین نوع حافظه وجود دارند که هر یک به منظور خاصی طراحی شده‌اند. با توجه به این اصل که کامپیوتراهای امروزی بر اساس معماری فون نیومان (Von Neumann Architecture) طراحی و ساخته می‌شوند، میتوان به اهمیت آن در معماری کامپیوترها نیز پی‌برد.

اولین سری حافظه‌ها پانچ کارت‌ها بودند. طرز کار آن‌ها بدین صورت بود که سوراخ‌هایی را بر روی آن ایجاد می‌کردند که همان نمایش صفر و یک بود. با افزایش حافظه مورد نیاز و این مهم که هر پانچ کارت میتوانست در حدود هشتاد بایت اطلاعات را ذخیره کند، پس تولید و استفاده از آن‌ها به طور کلی منسوخ شد. امروزه به صورت کلی حافظه‌ها به چهار دسته مغناطیسی، نوری، نیمه‌هادی و حافظه‌های مداری دسته بندی می‌شوند.

با توجه به اینکه همه ما روزانه داده‌هایی با حجم قابل تولید و ذخیره می‌کنیم، نقش و اهمیت حافظه‌های ارزان و دردسترس چندین برابر می‌شود. در این مطلب قصد داریم طرز کار و چگونگی ساخت ارزان‌ترین

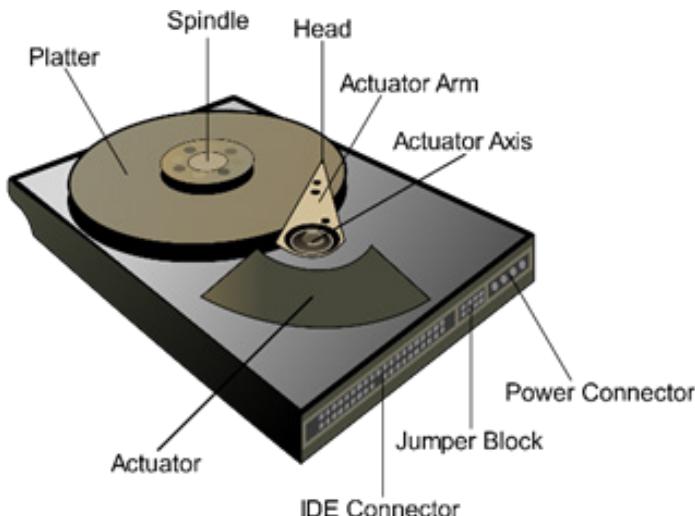
نوع حافظه یعنی هارد دیسک‌ها را بیان کنیم که جزو دسته حافظه‌های مغناطیسی محسوب می‌شوند. در شماره‌های بعدی نیز به انواع دیگر حافظه می‌پردازیم.

ابتدا فرایند کلی تکامل حافظه‌های مغناطیسی را دنبال می‌کنیم. یکی از اولین حافظه‌های مغناطیسی، نوارهای مغناطیسی بودند که در دهه پنجاه میلادی معرفی شدند و تا به امروز نیز مورد استفاده قرار می‌گیرند. این حافظه‌ها از یک نوار پلاستیکی نازک و بلند به عنوان بستر مواد فرو مغناطیسی استفاده می‌کنند. طرز کار آنها بدین صورت است که این نوارهای منعطف را رول می‌کنند و با استفاده از یک سوزن بر روی جهت‌های مغناطیسی را مشخص می‌کنند و از همین طریق آن‌ها را نوشته و می‌خوانند. یکی از بزرگترین ایرادهای نوارهای مغناطیسی بحث دسترسی به آنهاست. بدین‌گونه که برای خواندن اطلاعات در یک نقطه خاص میباشد از ابتدا تا آن نقطه قرائت شود.

هارد دیسک‌ها (Hard Disk Driver) که معمولاً در بازار با نام HDD شناخته می‌شوند؛ از پر طرفدارترین حافظه‌های موجود هستند. جدایی از قیمت آن‌ها فاکتورهای دیگری مانند حجم زیاد حافظه، راحتی حمل و نقل، مانایی زیاد اطلاعات در آن و همچنین سرعت قابل قبول نسبت به حجم و قیمت از دلایل پرطرفداری آن هاست. حال بیایید نگاهی به ساختار و طرز کار آن‌ها بیاندازیم.

مهم‌ترین قسمت‌های هارد دیسک‌ها شامل هد (سوزن) و صفحه مغناطیسی آن می‌شود؛ برروی صفحات مغناطیسی داده‌ها ذخیره می‌شوند و هد هم وظیفه خواندن و نوشتن (بازنویسی) را برعهده دارد.

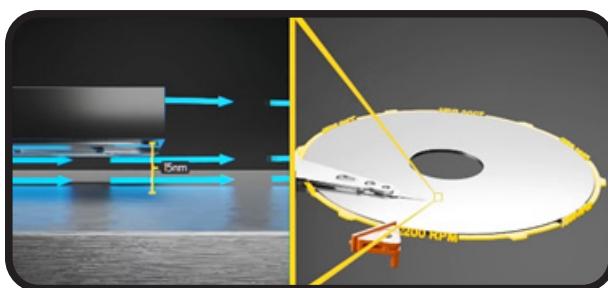
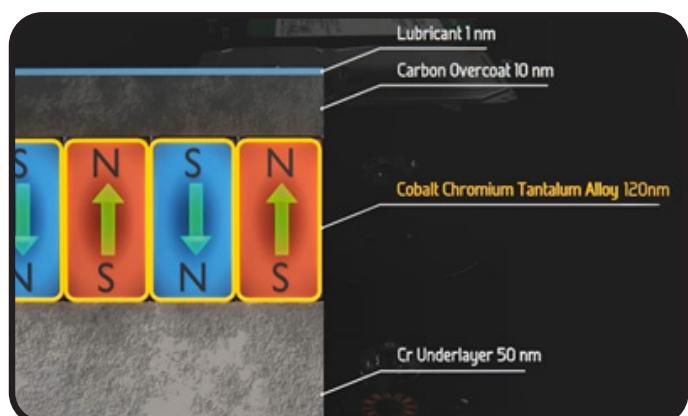
اجزای داخلی



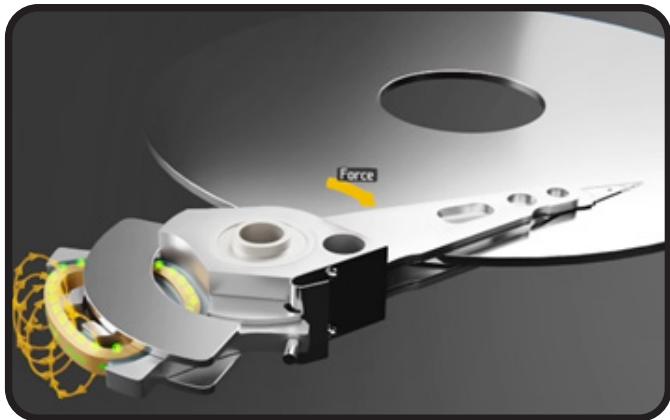
در قسمت داخلی این درایو، قطعات مختلفی را میبینیم. در اینجا دیسک یا صفحه‌ای است که همه دادهها را ذخیره می‌کند، و بسته به ظرفیت ذخیره‌سازی درایو، ممکن است چند صفحه بلند باشد. این دیسک از یک آلیاژ آلومینیوم منیزیم با پوشش‌های متعدد از آلیاژهای دیگر تشکیل شده است، اما لایه مغناطیسی و عملکردی این لایه

نازک ۱۲۰ نانومتری از آلیاژ کبالت کروم تانتال است که دارای حوزه‌های مغناطیسی کوچک یا مناطقی است که جهت آنها را می‌توان از طریق میدان‌های مغناطیسی خارجی دستکاری کرد. پلاٹر روی یک دوک نصب شده است که با سرعت ۷۲۰۰ دور در دقیقه با استفاده از یک موتور DC بدون جاروبک (Brushless) در مرکز آن می‌چرخد. بعد، یک مجموعه پشته سر، با یک بازو در بالا و یک بازو در زیر هر دیسک، و با یک نوار لغزنه و یک سر خواندن/نوشتن در انتهای هر بازو وجود دارد.

این نوار لغزنه به گونه‌ای منحصر به فرد طراحی شده است که جریان هوای تولید شده توسط دیسک سریع چرخش را جذب می‌کند و از جریان هوای برای صاف کردن یا به حرکت درآوردن هد خواندن/نوشتن استفاده می‌کند به طوری که تنها ۱۵ نانومتر یا حدود ۱۰۰ اتم از سطح فاصله دارد. از آنجایی



که مجموعه بازو در بالای دیسک چرخان پرواز می‌کند، تنها زمانی که چرخش دیسک با سرعت کامل باشد روی سطح آورده می‌شود، و زمانی که دیسک نمی‌چرخد، مجموعه بازو روی یک قطعه کوچک پلاستیکی در کنار پارک می‌شود. در اینجا یک موتور سیم پیچ صوتی مت Shank از یک سیم پیچ و دو منگنت نیودیمیوم قوی در بالا و پایین برای حرکت دادن کل مجموعه دسته بازو استفاده می‌شود. هنگامی که جریان الکتریکی از سیم پیچ عبور می‌کند، یک الکترومغناطیس ایجاد می‌کند که تحت تأثیر آهنرباهای نئودیمیوم است، بنابراین نیرویی ایجاد می‌کند که



باعث می شود بازو در سراسر دیسک حرکت کند. هنگامی که یک جریان معکوس از طریق سیم پیچ صدا ارسال می شود، بازو در جهت مخالف به حرکت در می آید، در نتیجه کنترل موقعیت دقیق هد خواندن/نوشتن در 30 نانومتر یا بیشتر امکان پذیر می شود. علاوه بر این، آهنرباها و سیم پیچ صدا یک موتور نسبتاً قوی ایجاد می کنند که به مجموعه

دسته بازوهای سبک وزن و سر خواندن/نوشتن این امکان را می دهد که تا 20 بار در ثانیه به عقب و جلو برود و روی صفحات مختلف حرکت کند و تنظیمات کوچک را با سرعت باورنکردنی انجام دهد. برای اتصال به هد خواندن/نوشتن، یک نوار منعطف از سیمها در امتداد کنار بازو و پایین به این کانکتور هدایت می شود که سیگنالها را به بیرون محفظه هارد دیسک و به برد مدار چاپی یا PCB می رساند. در ما پردازنده اصلی و همچنین یک تراشه DRAM داریم که به عنوان یک صفحه خراش برای پردازنده و یک بافر برای داده های ورودی و خروجی استفاده می شود. بر روی PCB تراشه ای برای کنترل سیم پیچ صدا و موتور اسپیندل DC بدون جاروبک نصب شده و سپس روشن می شود.

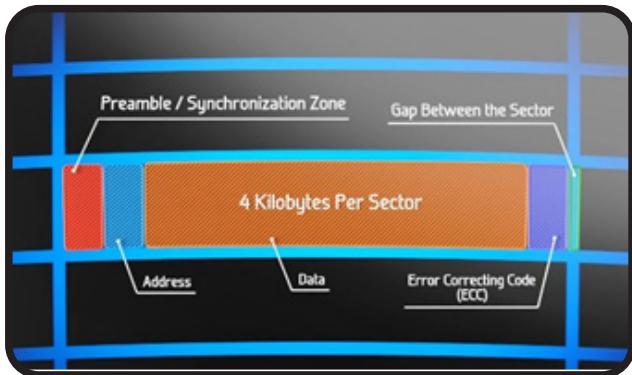
لبه PCB کانکتور SATA است که برای ارتباط به مادربرد وصل می شود و کانکتور جداگانه ای که به منبع تغذیه می رود.



اجزای مهم دیگر عبارتند از واشر که دیسک را از محیط بیرونی آب بندی می کند و دو فیلتر که ذرات گرد و غبار سرگردان را می گیرد. این فیلترها از آنجایی که هدهای خواندن/نوشتن فقط 15 نانومتر با صفحه فاصله دارند، ضروری هستند؛ یک لایه گرد و غبار می تواند تا 10000 نانومتر بزرگ باشد و در صورت برخورد با دیسک 7200 دور در دقیقه میتواند آسیب بزرگی به باز بیاورد.



اکنون که بسیاری از این اجزا را بررسی کردیم، بباید ببینیم که چگونه کار میکنند. برای شروع، دیسک به دایره هایی متحدمالمرکز از تراک ها تقسیم می شوند (برای درک بهتر میتوان خطوط موازی خط استوا را نسبت به یکی از قطب های جغرافیایی در نظر گرفت). جدیدترین هارد دیسکها می توانند $500,000$ تراک را فقط در یک طرف داشته باشند. این تراکها سپس به بخش هایی تقسیم می شوند و در هر بخش یک مقدمه یا منطقه همگام سازی وجود دارد که دیسک در حال چرخش و طول هر بیت داده را به هد خواندن/نوشتن می گوید. بخش بعدی سکتور آدرسی است که به هد خواندن/نوشتن کمک می کند تا بداند در حال حاضر روی کدام تراکها و بخش قرار گرفته است.

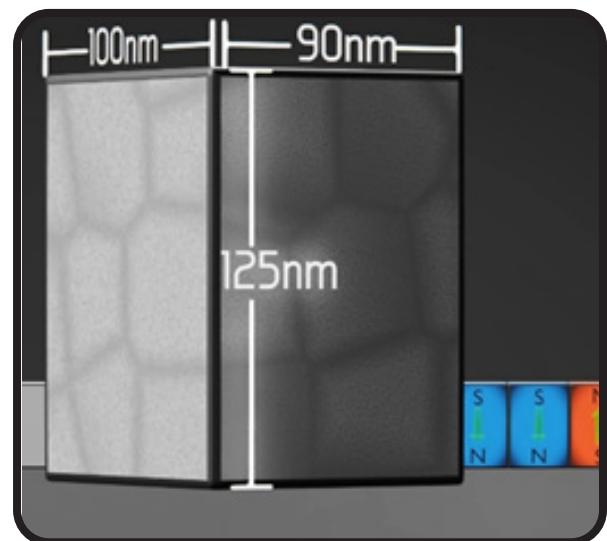


پس از آن، ما داده‌های واقعی ذخیره شده را داریم، معمولاً هر بخش شامل چهار کیلوبایت است. بعد از آن ناحیه‌ای برای کد تصحیح خطای ECC است که برای این که اگر داده‌ها را ذخیره کنیم در بلوک به درستی نوشته شده و به درستی

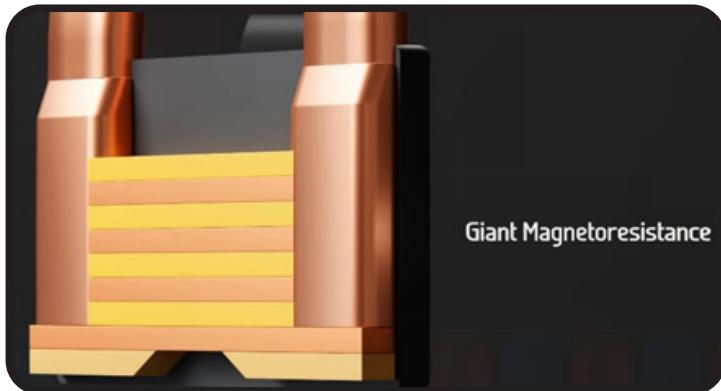
خوانده شده است استفاده می‌شود و در نهایت فاصله‌ای بین سکتور و سکتور بعدی وجود دارد که به سر خواندن/نوشتن، هنگام نوشتن محتويات یک بلوک مقداری تلوارنس می‌دهد. حالا بباید روی هد خواندن/نوشتن و دیسک زوم کنیم تا ببینیم داده‌ها دقیقاً چگونه نوشته و خوانده می‌شوند. نوشتن داده‌ها روی دیسک با دستکاری جهت مغناطیسی یک ناحیه موضعی یا دامنه لایه کبالت-کروم-تانتال در دیسک و فوکوس کردن ناحیه‌ای که قرار است در جهت بالا یا پایین مغناطیسی شود انجام می‌شود.

این حوزه یا ناحیه مغناطیسی کوچک به شکل مکعب مستطیل و در حدود ۹۰ در ۱۰۰ در ۱۲۵ نانومتر است و وقتی مغناطیسی می‌شوند، تمام اتم‌ها قطب‌های مغناطیسی شمالی و جنوبی خود را در یک جهت خواهند داشت.

به منظور مغناطیس کردن یک دامنه، که معادل نوشتن یک بیت داده است، یک جریان به سیم



پیچی در پشت سر نوشتن اعمال می‌شود، بنابراین یک میدان مغناطیسی قوی در اینجا ایجاد می‌شود. میدان مغناطیسی از طریق سر نوشتن هدایت و به نقطه کوچکی در نوک متمرکز می‌شود و سپس از شکاف هوای ۱۵ نانومتری و به داخل دیسک می‌پرد. هنگامی که میدان مغناطیسی متمرکز به یک حوزه منفرد از کبالت-کروم-تانتالوم عبور می‌کند، همه این اتم‌ها مجبور می‌شوند میدان‌های مغناطیسی اتمی کوچک خود را با میدان مغناطیسی اعمال شده از سر نوشتن همتراز کنند، بنابراین حوزه به یک آهنربای دائمی تبدیل می‌شود. نکته کلیدی این است که حتی زمانی که هد نوشتن دور می‌شود، جهت حوزه‌های مغناطیسی در این لایه از دیسک برای سال‌ها حفظ می‌شود و آنها یک میدان مغناطیسی دائمی ساطع می‌کنند که می‌تواند بارها و بارها توسط هد خوانده شده در هر بار خواندن حس شود.

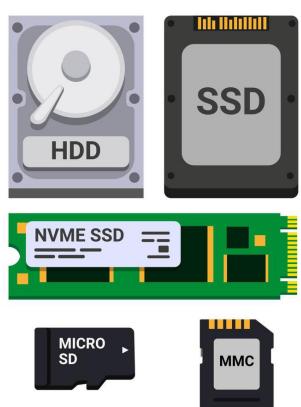
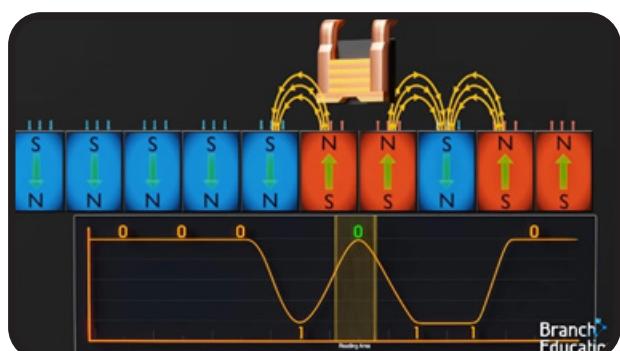


Giant Magnetoresistance

حال به بررسی نحوه خواندن دیسک میپردازیم. تا اینجا ما دامنهای رو به بالا را به صورت باینری ۰ و رو به پایین را به عنوان ۰ نشان میدهیم. اگرچه این از نظر مفهومی ساده است، اما در واقع اینطور نیست؛ هدف خواندن برای تشخیص تغییرات جهت گیری از حوزه‌های

مغناطیسی که در یک جهت و سپس دامنه مجاور در جهت مخالف هستند طراحی شده است. این به این دلیل است که میدان‌های مغناطیسی ساطع شده از مناطق مجاور که جهت‌های خود را تغییر می‌دهند، بسیار قوی‌تر از میدان تابیده شده از یک حوزه واحد است که به یک جهت یا جهت دیگر اشاره می‌کند. بنابراین، به هر کanal در حوزه مغناطیسی که در یک جهت به جهت مخالف اشاره می‌کند، ۱ اختصاص داده می‌شود، و به عدم انتقال از یک دامنه به دامنه بعدی، ۰ اختصاص می‌یابد. بنابراین، هد نوشتن یک دنباله باینری را می‌تواند ثبت کند.

چه چیزی در داخل سر خواندن وجود دارد که این میدان‌های مغناطیسی را تشخیص می‌دهد؟ داخل آن یک ماده رسانای چند لایه است که از لایه‌های متناوب مواد فرومغناطیسی و غیر مغناطیسی تشکیل شده است. این ماده چند لایه خاصیتی به نام مقاومت مغناطیسی غول پیکر یا GMR دارد و به زبان ساده، ماده‌ای است که بسته به قدرت میدان‌های مغناطیسی که از آن عبور می‌کنند، مقاومت خود را تغییر می‌دهد. بنابراین با استفاده از GMR، فقط اندازه‌گیری مقاومت بسیار ساده است، پس زمانی که مقاومت زیاد است نشان دهنده عدد یک و زمانی که مقاومت کم است نشان دهنده عدد صفر است.



در این مطلب سعی شد با زبانی ساده طرز کار کلی یک هارد دیسک بررسی شود. در شماره‌های بعدی به بررسی انواع حافظه‌های دیگر و همچنین انواع آدرس دهی پرداخته خواهد شد.





BYTE-SIZED Experiments

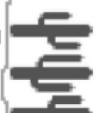


نویسنده: امید پیشرو

بخش ۱

سلام به همگی! به اولین شماره از Byte-sized experiments خوش آمدین. این در این قسمت قراره با هم یاد بگیریم که چطوری می‌تونیم با پایتون، یک گیم خیلی ساده طراحی بکنیم.

طمئنا همتون تا الان موقع قطعی اینترنت با دایناسور گوگل بازی کردین. من در اعتراض به اینکه گربه به این نازی هستش چرا دایناسور رو انتخاب کردین او مدم و نسخه گربه ای رو ساختم! هرکی هم دوست نداره جمع کنه بره.



کلاس‌ها:

در این قسمت باید برای تمامی اشیاء ای که در بازی وجود خواهند داشت، کلاس تعریف کنیم. cloud، catosaur، cactus، ptero اشیاء مورد استفاده این چهار تا هستند: . جلوتر دونه دونه این‌ها رو توضیح خواهیم داد. با cloud شروع می‌کنیم:

```
class Cloud(pygame.sprite.Sprite):
    def __init__(self, image, x_pos, y_pos):
        super().__init__()
        self.image = image
        self.x_pos = x_pos
        self.y_pos = y_pos
        self.rect = self.image.get_rect(center=(self.x_pos, self.y_pos))
    def update(self):
        self.rect.x -= 1
```

تقریباً تمامی المان‌های گیم که قراره باهашون تعامل داشته باشیم، باید sprite بشن. عموماً بخشی از گیمه که نیاز به تعامل با بقیه محیط گیم دارد؛ مثلًا وقتی دو تا توپ به هم می‌خورن، ماشین با دیوار تصادف می‌کنه، وقتی که بازیکن باید یه ستاره رو جمع بکنه و... . برای همین باید این‌ها رو sprite بکنیم.

داخل کلاس image داریم که عکس ابرمون هست، موقعیت عرضی و موقعیت طولی رو داریم اون عکس رو با موقعیت عرضی و طولی تطبیق می‌ده. فانکشن update هم هر بار یه دونه از موقعیت عرضی کم می‌کنه که به نظر برسه ابر داره از سمت راست به سمت چپ میره.

برنامه رو با import pygame کتابخونه‌های sys, random شروع می‌کنیم.

```
import pygame
import sys
import random
```

حالا با استفاده از مازول init، تمام چیزهایی که از pygame نیاز داریم رو استارت می‌زنیم و بلافاصله آماده کارش می‌کنیم.

```
pygame.init()
```

حالا نیاز داریم که screen رو ایجاد کنیم، وقت رو تعریف کنیم و کیشن بازی رو بنویسیم. کیشن همون اسم پنجره بازی هستش که در گوشه بالا نمایش داده می‌شده. در آخر هم فونتی که می‌خواهیم برای بازی استفاده کنیم رو وارد می‌کنیم. تمامی فایل‌هایی که نیاز دارید، در پوشه assets موجود هستش که از لینک گیت هاب می‌توانیں برش دارین.

```
screen = pygame.display.set_mode((1280, 720))
clock = pygame.time.Clock()
pygame.display.set_caption("Cat Run")
game_font = pygame.font.Font("assets/PressStart2P-Regular.ttf", 24)
```

ابعاد screen رو بر روی ۱۲۸۰ و ۷۲۰ پیکسل تنظیم کردیم که با تغییر دادن این مقادیر، اندازه پنجره هم تغییر می‌کنه. متغیر اول width هست (که این جا روی ۱۲۸۰ پیکسل تنظیم شده) و متغیر دوم height هست (که بر روی ۷۲۰ پیکسل تنظیم شده).



```

class Catosaur(pygame.sprite.Sprite):
    def __init__(self, x_pos, y_pos):
        super().__init__()
        self.running_sprites = []
        self.ducking_sprites = []

        self.running_sprites.append(pygame.transform.scale(
            pygame.image.load("assets/frame00.png"), (80, 100)))
        self.running_sprites.append(pygame.transform.scale(
            pygame.image.load("assets/frame01.png"), (80, 100)))
        self.running_sprites.append(pygame.transform.scale(
            pygame.image.load("assets/frame02.png"), (80, 100)))
        self.running_sprites.append(pygame.transform.scale(
            pygame.image.load("assets/frame03.png"), (80, 100)))
        self.running_sprites.append(pygame.transform.scale(
            pygame.image.load("assets/frame04.png"), (80, 100)))
        self.running_sprites.append(pygame.transform.scale(
            pygame.image.load("assets/frame05.png"), (80, 100)))
        self.running_sprites.append(pygame.transform.scale(
            pygame.image.load("assets/frame06.png"), (80, 100)))
        self.running_sprites.append(pygame.transform.scale(
            pygame.image.load("assets/frame07.png"), (80, 100)))

        self.ducking_sprites.append(pygame.transform.scale(
            pygame.image.load(f"assets/duckframe00.png"), (100, 60)))
        self.ducking_sprites.append(pygame.transform.scale(
            pygame.image.load(f"assets/duckframe01.png"), (100, 60)))
        self.ducking_sprites.append(pygame.transform.scale(
            pygame.image.load(f"assets/duckframe02.png"), (100, 60)))
        self.ducking_sprites.append(pygame.transform.scale(
            pygame.image.load(f"assets/duckframe03.png"), (100, 60)))

```



```

self.ducking_sprites.append(pygame.transform.scale(
    pygame.image.load(f"assets/duckframe04.png"), (100, 60)))
self.ducking_sprites.append(pygame.transform.scale(
    pygame.image.load(f"assets/duckframe05.png"), (100, 60)))
self.ducking_sprites.append(pygame.transform.scale(
    pygame.image.load(f"assets/duckframe06.png"), (100, 60)))
self.ducking_sprites.append(pygame.transform.scale(
    pygame.image.load(f"assets/duckframe07.png"), (100, 60)))

self.x_pos = x_pos
self.y_pos = y_pos
self.current_image = 0
self.image = self.running_sprites[self.current_image]
self.rect = self.image.get_rect(center=(self.x_pos, self.y_pos))
self.velocity = 50
self.gravity = 4.5
self.ducking = False

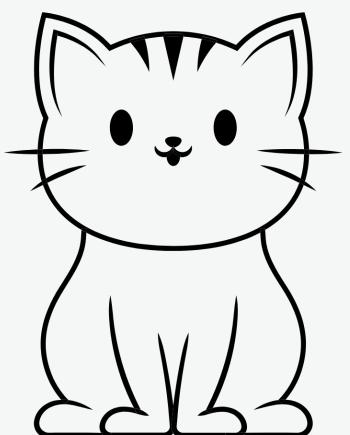
def jump(self):
    jump_sfx.play()
    if self.rect.centery >= 360:
        while self.rect.centery - self.velocity > 40:
            self.rect.centery -= 1

def duck(self):
    self.ducking = True
    self.rect.centery = 380

def unduck(self):
    self.ducking = False
    self.rect.centery = 360

def apply_gravity(self):

```



```

if self.rect.centery <= 360:
    self.rect.centery += self.gravity

def update(self):
    self.animate()
    self.apply_gravity()

def animate(self):
    self.current_image += 0.08
    if self.current_image >= 8:
        self.current_image = 0

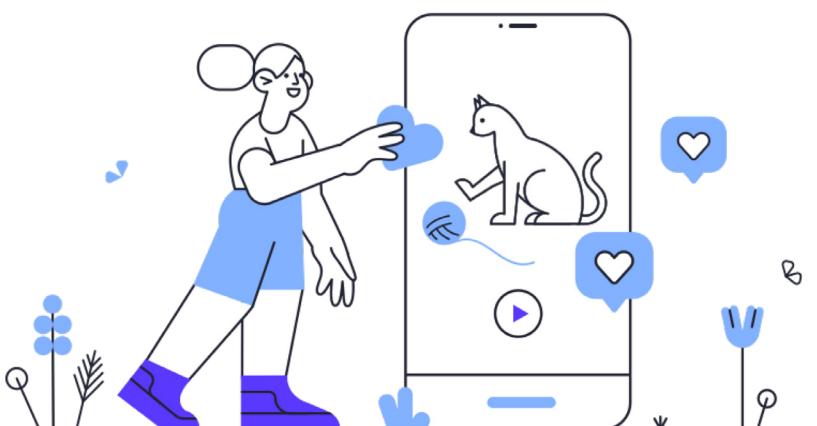
    if self.ducking:
        self.image = self.ducking_sprites[int(self.current_image)]
    else:
        self.image = self.running_sprites[int(self.current_image)]

```

این کلاس همون بازیکن اصلیمونه و دو متغیر موقعیت عرضی و طولی می‌گیره. دو تا عمل خیلی مهم هم انجام می‌ده: راه میره (running) و خم می‌شه (ducking). برای این دو تا حرکت، هر کدام هشت تا تصویر و فریم مختلف (در جمع ۱۶ تا تصویر) دارن که از ۰ تا ۷ شماره‌گذاری شدن و می‌تونید تو پوشه assets پیدا شون کنید. بعدش باید سرعت (velocity) و گرانش (gravity) رو تنظیم کنیم که ما به ترتیب روی ۵۰ و ۱۴/۵ تنظیمشون کردیم. می‌تونید با این اعداد بازی کنید و تغییرشون بدید تا ببینید چه تاثیری روی نحوه بازی به جا می‌داره.

متغیر مهم دیگه هم self.ducking هست که باید این رو روی False قرار بدم چون به صورت عادی گربه ما به صورت ایستاده اس و نیاز داریم که یکی از دو حالت رو به صورت حالت پیش فرض تعریف کنیم.

در تابع jump، پرش رو تعریف می‌کنیم. بعد از این که پرش کرد، باید برگردنه به موقعیت قبلی (درواقع سقوط کنه). برای این حلقه while تعریف می‌کنیم و می‌گوییم اگر centery بالاتر از ۳۶۰ بود، تا وقتی که self.rect.centery - self.velocity > 40 هستش، یه دونه از self.rect.centery کم کن تا بیاد پایین. بعد تابع خم شدن رو داریم که ارتفاع رو کم می‌کنه و برعکسش که ارتفاع رو دوباره به حالت عادی برگردونه. تابع animate هم وظیفه چرخیدن بین عکس‌ها رو داره به صورتی که خیلی سریع از یه عکس به عکس دیگه میره و فریم به فریم باعث می‌شه تصور حرکت کردن گربه بهتون دست بده.





```

class Cactus(pygame.sprite.Sprite):
    def __init__(self, x_pos, y_pos):
        super().__init__()
        self.x_pos = x_pos
        self.y_pos = y_pos
        self.sprites = []
        for i in range(1, 7):
            current_sprite = pygame.transform.scale(
                pygame.image.load(f"assets/cacti/cactus{i}.png"), (70, 80))
            self.sprites.append(current_sprite)
        self.image = random.choice(self.sprites)
        self.rect = self.image.get_rect(center=(self.x_pos, self.y_pos))

    def update(self):
        self.x_pos -= game_speed
        self.rect = self.image.get_rect(center=(self.x_pos, self.y_pos))

```



این قسمت برای موانعی هستن (کاکتوس‌ها) که پلیر باید از روشنون بپره. این‌ها به صورت شانسی تولید می‌شون. باز هم این تصاویر و فایل‌ها در پوشه assets هستن و می‌تونین از اون جا برشون دارین. این عرض و طولی که می‌بینید، در واقع اون قسمتی هست که اگر بازیکن بهشون بخوره بازی رو می‌بازه. ما این رو hitbox خطاب می‌کنیم که اندازه واقعی یک شیء رو در بازی ها مشخص می‌کنه. بعد از اینکه مشخص کردیم عکس‌های کاکتوس‌ها رو شانسی از پوشه برداره، الان باید کاری کنیم که به نظر بیاد دارن به سمت ما میان. با متغیر game_speed که پایینتر تعریف شده و سرعت بازی رو مشخص می‌کنه (برای متغیر های مختلفی ازش استفاده می‌کنیم) به صورت مداوم از موقعیت عرضی کاکتوس کم میشه تا به نظر بیاد داره به سمت ما میاد.



```

class Ptero(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.x_pos = 1300
        self.y_pos = random.choice([280, 295, 350])
        self.sprites = []
        self.sprites.append(
            pygame.transform.scale(
                pygame.image.load("assets/Ptero1.png"), (84, 62)))
        self.sprites.append(
            pygame.transform.scale(
                pygame.image.load("assets/Ptero2.png"), (84, 62)))
        self.current_image = 0
        self.image = self.sprites[self.current_image]
        self.rect = self.image.get_rect(center=(self.x_pos, self.y_pos))

    def update(self):
        self.animate()
        self.x_pos -= game_speed
        self.rect = self.image.get_rect(center=(self.x_pos, self.y_pos))

    def animate(self):
        self.current_image += 0.025
        if self.current_image >= 2:
            self.current_image = 0
        self.image = self.sprites[int(self.current_image)]
```

شانسی تولید می‌شن. دوباره با روشی که برای حرکت گربه به کار بردیم کاری می‌کنیم که به نظر بیاد پرنده‌ها دارن بالشون رو تکون میدن. با تابع `animate` بین دو عکس جابجا می‌شیم و به این سادگی کد پرنده هامون هم تموم شدن.

پرنده‌هایی (پتروداکتیل) هستن که در بازی هستن. بعضی‌هاشون اون قدری بالان که نیازی نیست هیچ کاری بکنیم و خودشون عبور می‌کنن ازمن؛ ولی اگر بپریم، بهشون برخورد می‌کنیم. یک سری رو می‌شه هم با پریدن و هم با خم شدن رد کرد و یک سری رو فقط با پریدن. این‌ها هم به طور

```

def end_game():

    global player_score, game_speed

    game_over_text = game_font.render("Game Over!", True, "black")
    game_over_rect = game_over_text.get_rect(center=(640, 300))
    score_text = game_font.render(f"Score: {int(player_score)}", True, "black")
    score_rect = score_text.get_rect(center=(640, 340))

    screen.blit(game_over_text, game_over_rect)
    screen.blit(score_text, score_rect)

    game_speed = 6

    cloud_group.empty()
    obstacle_group.empty()

```



در قسمت بعدی کد، متغیرهایمان رو تعریف می‌کنیم. متغیر game_speed که معرف حضورتون هست و سرعت بازی رو مشخص می‌کنه. امتیاز رو تعریف میک نیم و متغیر game_over رو روی false تنظیم می‌کنیم. بعدها در تابع پایان بازی ازش استفاده می‌کنیم. نکته‌ای که باید بهش دقت کنیم، اینه که ممکنه بعضی وقتاً این رندوم تولید کردن موائع باعث ایجاد یک سری موائعی بشه که رد شدن ازشون با توجه به قابلیت‌هایی که برای بازیکن تعریف کردیم ناممکنه. مثلًا ممکنه ۱۰ تا مانع پشت سر هم بدون فاصله بیان و چون مقدار پرش ما، یا قابلیت‌های بازیکن ما به قدری نیست که بتونه این ۱۰ تا رو پشت سر بذاره، رد کردنشون ناممکن می‌شه و طراحی بازی ما اشتباهه. به این نوع اشتباه در طراحی gameplay design گفته میشه و به خاطر عدم پیش‌بینی سناریوهای ممکن توسط برنامه نویس به وجود می‌آید. برای همین باید بینشون فاصله‌ای تعریف کنیم که حداقل این قدر فاصله باید وجود داشته باشه تا بازی قابل ادامه باشه. متغیر obstacle_cooldown رو تعریف می‌کنیم که بین موائع فاصله وجود داشته باشه و مشکلمون به سادگی حل میشه.

الان باید زمین بازی رو تعریف کنیم. اندازه اش رو با عرض پنجره یکی می‌کنیم تا کل پنجره بازی رو در بر بگیره. الان نیازه که برای کلاس cloud که بالاتر نوشتم عکس هم تعریف کنیم. متغیر رو تعریف می‌کنیم و عکس زمین و ابر رو از پوشه assets بارگیری می‌کنیم.

الان برای هر کدام از کلاس‌ها گروهی ایجاد می‌کنیم تا sprite بشن. سپس برای گربه شیء تعریف می‌کنیم تا به گروهش اضافه کنیم.

الان باید صدای بازی رو مشخص کنیم. فعلًا برای پریدن، رسیدن به مضارب ۱۰۰ و تمام شدن بازی صدا درست کردیم که در پوشه assets موجودن. اونها رو با یک متغیر جدا مشخص می‌کنیم تا در تابعهایمان فراخوانی بشن.

```

while True:

    keys = pygame.key.get_pressed()

    if keys[pygame.K_DOWN]:
        cat.duck()

    else:
        if cat.ducking:
            cat.unduck()

for event in pygame.event.get():

    if event.type == pygame.QUIT:
        pygame.quit()
        sys.exit()

    if event.type == CLOUD_EVENT:
        current_cloud_y = random.randint(50, 300)
        current_cloud = Cloud(cloud, 1380, current_cloud_y)
        cloud_group.add(current_cloud)

    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_SPACE or event.key == pygame.K_UP:
            cat.jump()

        if game_over:
            game_over = False
            game_speed = 6
            player_score = 0

screen.fill("white")

```



در قسمت بعدی باید توابع‌مون رو تعریف کنیم. تابع مهمی که این جا داریم، تابع end_game است که تعیین می‌کنه چطوری بازی‌مون به اتمام می‌رسه. همین اولش می‌گیم اگر بازی تموم شد عبارت Game Over! رو پرینت کنه. بعدهش امتیاز کاربر رو بهش نشون بده و بگه کجا باید این امتیاز نشون داده بشه. بعدهش باید صفحه رو پاک کنه که این کار رو با استفاده از دو تابع obstacle_group.empty() و cloud_group.empty() انجام می‌دهیم. اینکه این تابع چطوری فعال می‌شه؛ باید بازی‌کن با موانع برخورد داشته باشه.

```

if pygame.sprite.spritecollide(catosaur_group.sprite, obstacle_group, False)
game_over = True

death_sfx.play()

if game_over:
    end_game()

if not game_over:

    game_speed += 0.0020

    if round(player_score, 1) % 100 == 0 and int(player_score) > 0:
        points_sfx.play()

    if pygame.time.get_ticks() - obstacle_timer >= obstacle_cooldown:
        obstacle_spawn = True

    if obstacle_spawn:
        obstacle_random = random.randint(1, 50)

        if obstacle_random in range(1, 7):
            new_obstacle = Cactus(1260, 340)
            obstacle_group.add(new_obstacle)
            obstacle_timer = pygame.time.get_ticks()
            obstacle_spawn = False

        elif obstacle_random in range(7, 10):
            new_obstacle = Ptero()
            obstacle_group.add(new_obstacle)
            obstacle_timer = pygame.time.get_ticks()
            obstacle_spawn = False

```



بعد یک حلقه طراحی می‌کنیم تا بازیکن بتوانه دستورات رو وارد کنه. کلید arrow down باعث تغییر حالت گربه از ایستاده به خم شده اس و رها کردنش هم دوباره باعث ایستادنش میشه. سپس برای بستن پنجره بازی و همچنین فراخوانی تابع prieden چند if تعریف می‌کنیم و در آخر برای شروع دوباره بازی بعد از پایان و نشان دادن صفحه game over ، یک حلقه if دیگر اضافه می‌کنیم.



و هم اکنون تنها چیزی که نیاز داریم تعریف کردن برخورد گربه با موانع و نحوه ادامه بازی هستش. اگر sprite-collide بین گروه گربه با گروه موانع باشه، این تابع متغیر game_over را از true به false تغییر میده و به این ترتیب تابع end_game وارد عمل میشه و بازی رو خاتمه میده. سپس تمامی تابع های اشیاء را فراخوانی می‌کنیم و با update کردن گروه ها بازی رو به صورت کامل اجرا می‌کنیم.

برای دسترسی به کدهای این قسمت کد QR روبه رو را اسکن کنید.



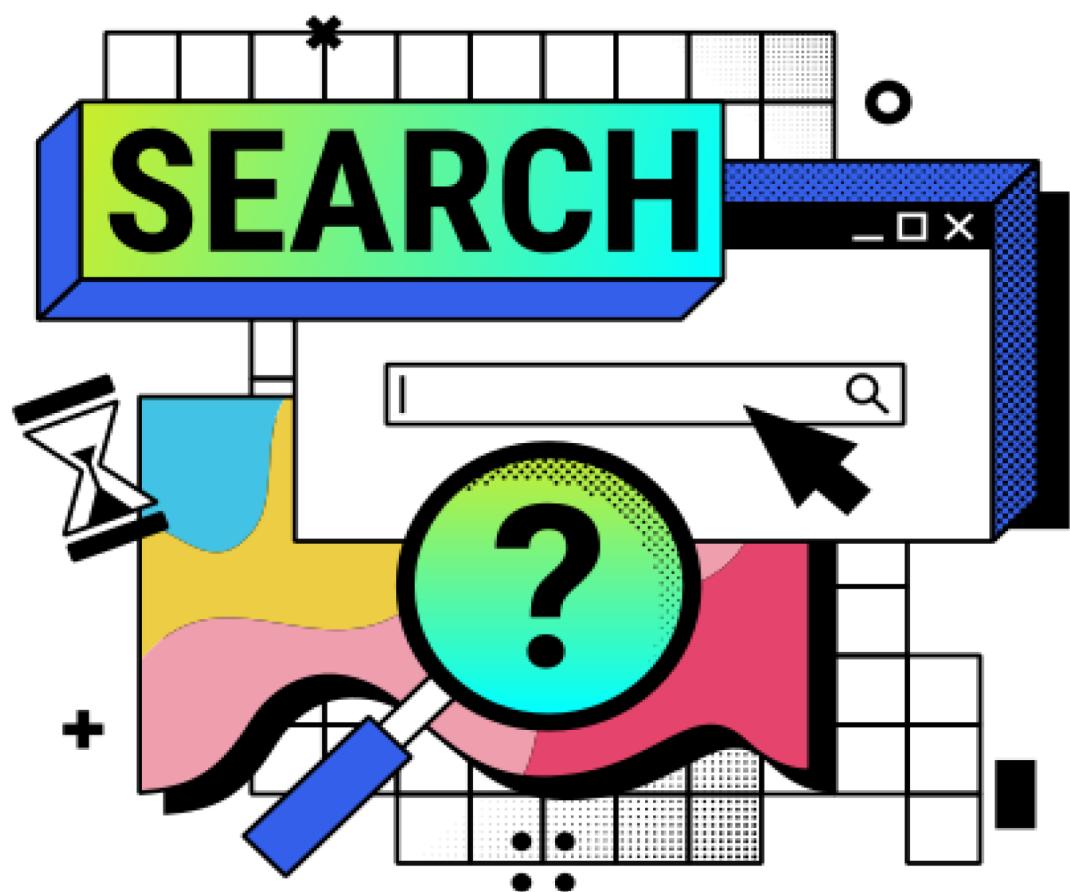


سفری به باغ وحش گوگل!



نویسنده: حدیثه خراسانی و شبتم رنجبری

بخش ۱



اگر گوگل را مثل غول چراغ جادویی در نظر بگیریم که قرار هست کاربرها رو به خواسته‌شان برساند، الگوریتم‌های گوگل به عنوان بچه غول‌های گوگل، حواسشان هست که چه طور کاربرها را به مسیر درست هدایت کنند. درست مانند شهری که شما در آن زندگی می‌کنید، اینترنت هم به یک سری قوانین نیاز دارد. برای اینکه اگر قانونی در کار نباشد، هرج و مرچ و سرگردانی بیداد می‌کند و هیچ چیز تحت کنترل نخواهد بود. به همین صورت اگر موتورهای جستجو هم هیچ قانونی نداشته باشد، همه جا پر می‌شود از سایتها اسپم و بی‌کیفیتی که تنها به دنبال منافع شخصی هستند و حاضرند هر کاری بکنند تا کاربر را به سایت خود بکشانند! ولی چاره چیست؟



موتورهای جستجوگر برای شناخت و درجه بندی وب سایتها الگوریتم‌های متنوعی دارند که همه صاحبان وب سایت که می‌خواهند در اندازه گیری‌های این الگوریتم‌ها سهمی داشته باشند و شناخته شوند.

در این سری مطالب ابتدا تعریف مختصی از الگوریتم‌ها خواهیم داشت و سپس به معرفی ۲ الگوریتم ابتدایی گوگل خواهیم پرداخت. در شماره‌های بعدی به ترتیب شمارا با الگوریتم‌های جدیدتر و کاراتر گوگل آشنا خواهیم کرد. پس با ما همراه باشید!

حتما اسم الگوریتم‌های گوگل به گوستان خورده! الگوریتم پاندا، پنگوئن، مرغ مگس‌خوار، رنکبرین، کبوتر، الگوریتم گورخر و پیج لایوت. اگر می‌خواهید سایت شما جزو پیشنهادهای اول گوگل به کاربرها باشد خوب است این مطالب را دنبال کنید تا با شناخت ویژگی‌های شخصیتی الگوریتم‌ها به شناخت غول گوگل، نزدیکتر شوید. صاحبان وب سایت دوست دارند وب سایتشان در صف اول موتورهای جستجو قرار بگیرد. بنابراین آن‌ها همیشه به دنبال راه‌هایی هستند که وب سایتشان به لحاظ فنی قابل فهم برای موتورهای جستجو باشد و با منطق‌های گوگل هم راستا باشد. البته این کار کاملاً فنی و تخصصی است.

الگوریتم چیست؟



در یک تعریف عمومی به مجموعه ای از دستور العمل ها و فرمول هایی که با زبان رسا و دقیق به همراه جزئیات لازم و به صورت مرحله به مرحله به گونه اجرا شده باشند که هدف خاصی را دنبال کنند و شروع و پایان آنها نیز مشخص باشد، الگوریتم گفته می شود.

الگوریتم های گوگل، راهی برای درک و رفع نیاز کاربر

زمان را چند سال به عقب برگردانید. در این زمان کافی بود تا بر روی یکی از نتایج گوگل کلیک می کردید تا با دریابی از تبلیغات نامرتبط و پاپ آپ های آزار دهنده روبرو می شدید. در این زمان نتایج گوگل بیشتر شبیه به ویرانه بود تا نتایج مورد نیاز کاربر!

خوشبختانه گوگل صدای فریاد رنج کاربران را می شنود و دست به طراحی و توسعه فرشته های نجات خودش می زند.

الگوریتم پاندا، پنگوئن و مرغ مگس خوار اولین موج و نیروی کمکی گوگل برای مقابله با سایتها ای اسپم بود.

الگوریتم های گوگل به کاربران یاد دادند که چطور باید دست به تولید محتوا بزنند. این الگوریتم ها معتقد بودند که تولید محتوا تنها باید در راستای نیاز کاربران باشد.

این الگوریتم ها دو هدف سرنوشت ساز و مهم داشتند:

- کاهش رتبه سایتها بی کفیت و اسپم
- افزایش رتبه سایتها با ارزش و کاربر محور

زمان زیادی طول کشید تا سایتها جایگاه واقعی خود را به دست گرفتند و در این زمان میدان جنگ سئو دیگر آن رنگ و بوی اسپم را نداشت.

Spam Score



این الگوریتم، نمایشی خطی از مقیاس لگاریتمی بین ۰ تا ۱۰ بود که عدد نهایی در نوار ابزار knaRegaP نمایش داده می‌شد. هنگامی که الگوریتم اولیه گوگل توسعه یافت، بر اساس این تئوری بود که ایجاد لینک از یک وبسایت به وبسایت دیگر به عنوان عاملی برای بالا رفتن اعتبار آن سایت عمل می‌کند. بنابراین، هرچه لینک (رأی)‌های بیشتری به یک صفحه وجود داشته باشد، آن صفحه اعتماد بیشتری جلب کرده و بنابراین، از رتبه بالاتری برخوردار می‌شود.

امتیاز در پیج رنک، به ما می‌گوید که کیفیت یک سایت پایین است، در حالی که امتیاز ۱۰، معتبرترین صفحات وب را به ما معرفی می‌کند. این نوار ابزار گوگل کاری کرد که متخصصان محتوا روی آن حساسیت بیشتری پیدا کنند و همه تلاششان صرف به دست آوردن امتیاز بالاتر بشود. این حساسیت تا جایی پیش رفت که این متخصصان، دیگر اهمیت چندانی به تولید محتوای خوب یا بهبود تجربه کاربر نمی‌دادند.

مشکل این بود که به اشتراک گذاشتن عمومی امتیاز اعتبار صفحات، این امکان را برای متخصصان سئو فراهم می‌کرد تا در یک رقابت پایان‌نایذیر بیافتدند و با دستکاری متن انکر تکست‌ها، لینک‌های wolloFoN ... کاری کنند تا پیج رنکشان نسبت به سایر رقبا، بالاتر بروند. گوگل در حال نیز از این ویژگی به عنوان بخشی از الگوریتم خود استفاده می‌کند. البته چیزی که در حال حاضر می‌بینیم با نمونه اصلی و اولیه آن تفاوت زیادی دارد و بسیار پیچیده‌تر است.

در اواخر قرن بیستم، گوگل تصمیم گرفت الگوریتم جدیدی را معرفی کند که توسط آن امکان رتبه‌بندی سایتها وجود داشته باشد. پیج رنک جزو اولین الگوریتم‌های گوگل است در ابتدا این الگوریتم لینک‌هایی که به هر دامنه یا صفحه داده می‌شد را بررسی می‌کرد و بر همین اساس هر سایتی که لینک‌های بیشتر و معتبرتری داشت، در رتبه‌های برتر قرار می‌گرفت.

رفته رفته گوگل الگوریتم پیج رنک را با هدف جلوگیری از ساخت بک لینک‌های انبوه و پولی توسعه داد.

اگر از زمان قدیم با گوگل و الگوریتم‌هایش سرو کار دارید، این تصویر احتمالاً همان چیزی است که وقتی به اعتبار یک وبسایت فکر می‌کنید، در ذهن شما خطور می‌کند.

این همان نوار ابزار معروف گوگل است که بهش اشاره کردیم. اما این الگوریتم معروف چیزی فراتر از یک نوار ابزار است. در سال ۲۰۰۰، گوگل نوار ابزاری را معرفی کرد که اکنون همه ما آن را به عنوان روشی که می‌توانیم امتیاز PageRank سایت خود (و رقبایمان) را مشاهده کنیم، به یاد می‌آوریم. (این نوار ابزار در حال حاضر دیگر موجود نیست و تنها در آن سال‌ها شانس استفاده از این قابلیت را داشتیم).





۲- الگوریتم دوم گوگل - سال ۲۰۰۴ (Google Dance Algorithm)

گوگل، محتواهای داخلی را بررسی نمی‌کند و فقط به نوعی لینک‌های مختلف در صفحات جستجوی گوگل (خصوصاً صفحه اول) را کمی به دوم در می‌آورد!

در ادامه با ما همراه باشید تا در مورد تمام چالش‌ها و سوالات در مورد الگوریتم دوم گوگل (Google Dance) صحبت کنیم.

الگوریتم دوم گوگل (Google Dance) چیست؟

الگوریتم دوم گوگل پدیده‌ای است که جایگاه کلمات کلیدی سایت شما، در گوگل جا به جا می‌شوند و به اصطلاح دچار "دوم" می‌شوند. به این معنا که کلمات کلیدی سایت شما یا در جایگاه بالاتر و یا در جایگاه پایین تری نسبت به همیشه قرار می‌گیرند. مثلًا دیروز، کلمات کلیدی شما جایگاه ۱، ۲ و ۳ بود اما امروز از ۱۰۰ تای اول خارج شده و فقط یک روز بعد، به سرعت در ۵ تای اول قرار می‌گیرد. بنابراین وقتی پدیده دوم گوگل اتفاق می‌افتد، برخی سئوکاران بسیار

در سال‌هایی که گوگل هر ساله سعی می‌کرد برای بهینه سازی بهتر و قوی‌تر موتورهای جستجوی خود و نمایش بهترین نتیجه به کاربر، خود را آپدیت کند، الگوریتم دوم گوگل وارد میدان شد. این الگوریتم از بین تمام الگوریتم‌های گوگل از نظر متخصصان سئو یکی از آزاردهنده‌ترین‌ها به شما می‌آید (حتماً حسابی با دومش تنشون رو به لرزه در میاره).

الگوریتم گوگل به این صورت کار می‌کند که هر چند وقت یک بار رتبه صفحات مختلف در کلمات کلیدی را در SERP تغییر می‌دهد. این روند به دلیل دادن شانس به سایت‌های تازه تاسیس و یا پایین آوردن سایت‌های خیلی معروف و بزرگ انجام می‌دهد. الگوریتم دوم گوگل، یک پدیده طبیعی در نتایج جستجوی گوگل است که باعث می‌شود رتبه‌بندی سایتها به طور مکرر تغییر کند. این تغییرات معمولاً در بازه‌های زمانی کوتاه، مانند چند روز یا چند هفته، رخ می‌دهد. البته این موضوع را فراموش نکنید که الگوریتم دوم

موارد نیز، ممکن است باعث نوسانات رتبه‌بندی سایت شود.

دلایلی که چرا سایت شما الگوریتم دوم گوگل را تجربه می‌کند؟

مهم نیست که موضوع وب سایت شما چیست، هر سایتی با هر موضوعی می‌تواند دچار الگوریتم دوم گوگل شود.

با این حال، یک سایت با شرایط زیر، می‌تواند بیشتر دچار گوگل شود:

سایت‌هایی که روند سئو سایتشان را به درستی انجام نمی‌دهند و یا با معیارهای گوگل، مغایرت دارند.

برای مثال:

- لینک سازی با سرعت بالا، کپی کردن محتوا از سایتها دیگر یا ساخت لینک‌های اسپم
- وب سایتها که به تازگی تاسیس شده‌اند.
- وب سایتها که کلمات کلیدی آنها به تازگی سئو شده باشد و هنوز رتبه آنها پایدار نیست.

الگوریتم‌های گوگل آپدیت شده‌اند.

آپدیت رتبه‌ی میلیون‌ها وب سایت همزمان، کار بسیار دشواری است. بنابراین، این الگوریتم گوگل (google dance) علاوه‌یک پدیده نرمال است. در ادامه هر یک از این موارد را به صورت تخصصی تر شرح می‌دهیم.

هیجان زده شده و سایرین ممکن است بسیار ناامید شوند.

به عبارت دیگر، الگوریتم دوم گوگل زمانی است که گوگل بتواند یک الگوریتم جدید را بنا کند یا یک آپدیت جدید از یک الگوریتم قدیمی ارائه دهد که به طور معمول، حدود یک هفته به طول می‌انجامد و بعد دوباره به روای عادی بر می‌گردد.

تأثیر الگوریتم دوم گوگل بر سئو سایت چیست؟

برای درک تأثیر دوم گوگل بر سئو سایت، باید بدانید که برای رتبه بندی کلمات کلیدی، گوگل بیش از ۳۰۰ معیار دارد و امروزه، هزاران وب سایت است که با یکدیگر در یک کلمه کلیدی رقابت می‌کنند (چه رقابت تنگاتنگی می‌آفریند این دو میان الگوریتم جنجالی گوگل).

برای هر معیار، گوگل هر سایت را مورد بررسی قرار می‌دهد. زمانی که یک سایت با معیار A مطابقت داشته باشد، توسط گوگل به آن امتیاز داده می‌شود و رتبه‌اش افزایش می‌یابد. این بدان معناست که وب سایتها دیگر افت خواهند کرد. همین روای برای تمامی سایتها ادامه پیدا می‌کند تا تمام این ۳۰۰ فاکتور برای تک تک سایتها بررسی شود. بنابراین زمانی که پدیده دوم گوگل اتفاق می‌افتد، تغییر رتبه در سایت چیزی بسیار معمول است و خیلی دشوار است که بتوان از آن اجتناب کرد.

الگوریتم دوم گوگل می‌تواند بر سئو سایتها تأثیرات مختلفی داشته باشد. در برخی موارد، ممکن است باعث افزایش رتبه‌بندی سایت شود و یا در برخی موارد کاملاً بر عکس عمل کند. در برخی

خواهد بسنجد آن سایت چند مرده حل ج است! و گرنه چرا باید مسیر سئوی سایت جدید را با چالش (آن هم همچین چالشی) مواجه کند؟ من که میگم میخواود زهر چشم بگیره).

۴. شروع روند آپدیت برای صفحات مختلف سایت

سومین عاملی که می تواند باعث اعمال دوم گوگل بر صفحه ای از سایت شما(یا کل سایت) شود، این است که به تازگی صفحات مختلف سایت را آپدیت کرده اید. دقیقا هیچ جواب قطعی برای این اتفاق وجود ندارد و کسی نمی تواند بگوید چرا گاهی اوقات بعد از آپدیت مشاهده می شود که سایت از سرپ خارج می شود، اما بعضی مواقع اتفاق می افتد.

بهترین راه حل این است که آپدیت صفحات را در بهینه ترین حالت ممکن انجام دهید. تمام قوانین الگوریتمها را رعایت کنید و اگر صفحات در کلمات کلیدی مختص خودشان رتبه های ۱ تا ۳ دارند، آپدیت نکنید.

چه وب سایتهايی از پدیده دوم گوگل در امان میمانند؟

سایتهايی که از پدیده گوگل در امان هستند، معمولا، سئو داخلی آنها به خوبی انجام شده است و در کنار آن بک لینکهای با کیفیت دارد. اول از همه، این سایتها به جای استفاده از ترفندهای گوناگون برای دور زدن الگوریتمهای گوگل، سعی می کنند با تولید محتوای با کیفیت، سوالات و نیازهای کاربران را برآورده کنند.

دوم، لینکهای داخلی و خارجی، به طور دقیق و منظم برای مقالات آن سایت ساخته می شود.

۱. تغییرات در الگوریتمهای گوگل

همانطور که می دانید، گوگل دائمًا الگوریتمهای خود را برای بهبود نتایج جستجوی خود بهروزرسانی می کند. این بهروزرسانیها می توانند باعث تغییراتی در رتبه بندی سایتها شوند.

یعنی زمانی که آپدیت جدید ارائه شده و می خواهد کار خود را شروع کند، گوگل نیز شروع به کار کرده و در یک تایم مشخص، صفحات را از سرچ خارج کرده یا رتبه ها را جابجا می کند. معمولا این روند بعد از ۲۴ ساعت به حالت اول بر می گردد.

۲. عدم استفاده از سئو کلاه سفید

زمانی که شما سعی کنید با راهکارهای سئو کلاه سیاه و یا خاکستری، الگوریتم های مختلف را دور بزنید و به درستی قوانین گوگل را رعایت نکنید، اتفاقات خوبی پیش روی شما وجود ندارد. در واقع همیشه توصیه بر این بوده است که برای سئو سایت خود و رتبه گرفتن در کلمات کلیدی مختلف حتما از سئو کلاه سفید بهره ببرید و روند سئو را کاملا طبیعی و طبق قوانین انجام دهید.

در غیر اینصورت الگوریتم دوم گوگل شما را گیر انداخته و در یک چشم به هم زدن می تواند سایت را از ۱۰۰ نتیجه اول خارج کند (این الگوریتم ما زبر و زرنگه اگه کسی بخواهد زرنگ بازی دربیاره زود مچشو میگیره!!)

۳. وبسایت تازه تاسیس، طعمه گوگل

معمولا در روند عملکرد گوگل ، سایت های تازه تاسیس بیشتر دچار نوسانات مختلف رتبه بندی خواهند شد. به شخصه تصور می کنم گوگل می

خطایی را فرستاده یا نه. هم چنین گوگل آنالیتیکس را بررسی کنید و ببینید که آیا شاخص هایی مانند بانس ریت (نرخ پرش از سایت)، مدت زمان ماندن در سایت، بازدید از صفحه و ... در وضعیت قابل قبولی هست یا نه، اگر وضعیت مناسب نبود، باید آنها را بهبود دهید.

بعد از بررسی سرج کنسول و گوگل آنالیتیکس، باید حجم کلمات کلیدی در صفحه را بررسی کنید. سعی کنید حجم کلمات کلیدی بین ۵ تا ۷ درصد باشد (یا حتی کمتر اگر امکانش هست)، برای اینکه گاهی اوقات، به کار بردن کلمات کلیدی بیش از حد، می تواند باعث افت رتبه سایت شما شود.

بعد از بهینه سازی حجم کلمات کلیدی، شما باید لینک های داخلی را بهینه کنید و محتوای جدیدی بگذارید که برای بازدید کنندگان مفید باشد. علاوه بر این، شما باید مقالات خوب و کاربردی در شبکه های اجتماعی به اشتراک بگذارید تا بازدید کنندگان را به سایت شما هدایت کند. سیگنال هایی که از شبکه های اجتماعی می گیرید، مانند لایک، به اشتراک گذاری و تعداد بالایی از بازدید کنندگان، می تواند بر سئو سایت شما تاثیر مثبت بگذارد.

شما باید منبع بک لینک های سایتتان را بررسی کنید. برای مثال بک لینک هایی که از فروم ها می گیرید، ناپایدارند (برای اینکه در فروم ممکن است پست ها پاک شوند و یا لینک های شما توسط مدیران انجمن حذف شوند یا تغییر داده شوند). این کار معمولاً باعث فعالیت های غیر معمول می شود و باعث می شود کلمات کلیدی شما دچار دوم گوگل شوند.

علاوه بر آن، این وب سایت ها معمولاً در مدت زمان طولانی دارای سیستم های بک لینک با کیفیت بالا و پایدار هستند.

زمانی که رتبه سایتتان افت می کند، چه کار می توانید انجام دهید؟

گاهی اوقات، پدیده گوگل (google dance) به جای سند باکس (google sandbox) اشتباہ گرفته می شود. بنابراین شما باید دلیل اینکه سایت شما از رتبه های بالا افت می کند را پیدا کنید و یک راه حل مناسب برای آن ارائه دهید ساده ترین راه برای نجات از خطرات الگوریتم گوگل در جنگل بزرگ گوگل، رصد کردن رقبا و غیر عادی بودن رفتار موتورهای جستجو است. اگر رتبه سایت شما افت کرد، باید سایتتان را بررسی کنید و ببینید که آیا محتوای کپی دارد یا نه. واضح است که اگر محتوای کپی شده داشته باشد، دچار پنالتی گوگل می شوید.

از سوی دیگر، اگر سایت شما دچار اختلالات شدید در رتبه بندی شد، و رتبه های بالای خود را از دست داد، نشان می دهد که سایت شما دچار الگوریتم دوم گوگل شده است. اولین کاری که باید انجام بدهید این است که بررسی کنید در یک هفته اخیر، چه کارهایی با وبسایتتان کردید. دوباره بررسی کنید که برای سئو داخلی و سئو خارجی سایتتان در این یک هفته چه کارهایی انجام دادید، اگر چیزی به نظر غیر منطقی رسید، آن را به حالت اولیه برگردانید.

در این شرایط، شما باید وبمستر تولز گوگل (سرچ کنسول) را مورد بررسی قرار دهید و ببینید که آیا برای آن صفحه ای که دچار افت رتبه شده،

کار ما هنوز به پایان نرسیده در سری های بعدی
شما را با الگوریتم های دیگر که در راستای بهبود
عملکرد قوانین جنگل گوگل کار میکنند آشنا
خواهیم کرد...

اگر می خواهید به یک جایگاه پایدار در گوگل برسید،
باید تجربه کاربری خود را به حداکثر برسانید. به
یاد داشته باشید: سایتها به این دلیل ساخته
می شوند که نیاز کاربران را برآورده کنند نه گوگل
را. بنابراین، هرچه کاربران بیشتر در صفحه شما
وقت صرف کنند، احتمال در امان بودن سایتتان
از الگوریتم دوم گوگل بیشتر خواهد شد.

چگونه پدیده google dance را تشخیص دهیم؟

رتبه گرفتن گوگل به فاکتورهای زیادی بستگی دارد.
یکی از آنها بستگی به پنالتی شدن سایت شما
توسط گوگل دارد. خب حالا از کجا تشخیص دهیم
که افت رتبه سایتمان به خاطر الگوریتم google
dance (گوگل) هست یا دچار پنالتی گوگل شده
است؟ برای اینکه جواب این سوال را بیابید، حتما
پنالتی گوگل را مطالعه نمایید. اما اگر بخواهم به
طور خلاصه به این سوال پاسخ دهم، باید بگوییم
ما دو نوع پنالتی داریم، پنالتی دستی و پنالتی
الگوریتم. اگر پنالتی به صورت دستی باشد، با وارد
شدن به سرج کنسول (وبمستر تولز گوگل)، می
توانید از بخش "Manual action" ببینید که آیا
پیغام خطای دارید یا نه. اگر به این صورت پنالتی
نشده باشید، پیغام "No issues detected" را
مشاهده می کنید یعنی "هیچ مشکلی تشخیص
داده نشده".

اما اگر دچار پنالتی الگوریتمی شدید، مثلًا همین
الگوریتم گوگل ، آن وقت دیگر هیچ پیغام خطای
را در سرج کنسول گوگل مشاهده نمی کنید و باید
تمامی الگوریتم های گوگل و فاکتورهای سئو را
بررسی کنید و ببینید که آیا تمام موارد را درست
رعایت کرده اید یا نه.



ما را در صفحات زیر دنبال کنید ...



www.youtube.com/@uucessc



<https://www.instagram.com/uucessc>



<https://github.com/UUCESSC>



t.me/uucessc