

مطالبی که باید گفته بشه:

- معرفی کدفورسز و پلتفرم‌های مشابه
- صفحه‌ی کانتست‌ها
- کانتست‌های ریتد و آن‌ریتد

- <https://codeforces.com/blog/entry/62865>

- معرفی رنک‌ها و خفن‌های ایران و جهان

- ویژگی‌های اسکوربورد و نحوه‌ی دیدن کد سوالات

- مفهوم هک کردن

- معرفی کانتست‌ها (دیوها و اجوکیشنال). با لینک شاید حتی

- معرفی ادیتوریال و تگ‌های سوال، یکی از مهم‌ترین‌هاش تگ difficulty

- مفهوم ریتینگ. با لینک شاید حتی

- صفحه‌ی پروفایل و قسمت‌های مهم آن، برای ما قسمت مهمش from, blog, submissions و کامنت هست

- بخش problemset و ویژگی‌های سرچش

- معرفی بلاگ‌ها

- معرفی کانال تلگرام کدفورسز و بات کانتست‌ها

در پست قبل در مورد برنامه‌نویسی رقابتی و دلایل اهمیت اون صحبت کردیم. در این پست قصد داریم پلتفرمی در این زمینه رو معرفی کنیم که یکی از منابع اصلی آموزش و پیشرفت در این زمینه محسوب میشه.

TL;DR

کدفورسز یکی از بهترین روش‌های یادگیری و پیشرفت در حوزه‌ی برنامه‌نویسی رقابتی هست. به طور خلاصه برای استفاده‌ی مناسب از کدفورسز می‌توان به صورت زیر عمل کرد:

- به طور منظم در کانتست‌های کدفورسز شرکت کنیم.
- سوالات با سختی بالاتر از ریتینگ خود رو که در زمان کانتست حل نکردیم رو با استفاده از ادیتوریال کانتست بررسی کنیم و دلیل حل نکردنش رو متوجه بشیم (به طور مثلا درست فکر نکردن و یا مباحثی رو بلد نبودن).
- مباحثی که با استفاده از ادیتوریال متوجه می‌شویم که بلد نیستیم و یا سطحی بلدیم رو با سرچ کردن در کدفورسز یاد بگیریم.
- در صورت نیاز، کد شرکت‌کننده‌های مطرح که در کانتست شرکت کردند رو ببینیم و از روش‌هایی که استفاده می‌کنند یاد بگیریم.
- پیگیر پست‌های بلاگ کدفورسز باشیم و مباحث جدید یاد بگیریم.

معرفی کدفورسز

[کدفورسز](#) یک پلتفرم برگزاری کانتست‌های الگوریتمی هست (برای آشنایی با کانتست‌های الگوریتمی و برنامه‌نویسی رقابتی، [این لینک](#) رو ببین)

سایت‌های زیادی هستند که کانتست‌های الگوریتمی برگزار می‌کنند. چندتا از معروف‌ترین‌هاش عبارتند از: codechef، atcoder، topcoder و ... ولی همیشه گفت که فعال‌ترین و معروف‌ترینش کدفورسزه.

کدفورسز هر هفته، تقریبا به صورت منظم، کانتست‌های متفاوتی برگزار می‌کنه که با انواع اون‌ها بعدا آشنا می‌شیم. لیست کانتست‌هایی که قراره برگزار بشه و همه‌ی کانتست‌های برگزار شده رو میشه توی بخش [CONTESTS](#) (نوار بالای سایت) دید. این کانتست‌ها به طور معمول ۲ ساعته هستن که همین مدت نسبتا کم باعث میشه برای تمرین و یادگیری گزینه‌ی مناسبی باشند (البته دلایل بیشتر و مهم‌تری هم هستن که می‌رسیم بهشون). یه مزیت خوبی که این کانتست‌ها برای ما ایرانی‌ها داره اینه که زمان برگزاری‌شون معمولا مناسبه.

بد نیس با موسس و مدیر کدفورسز، [Mike Mirzayanov](#) که خیلی آدم کول و باحالی هم هست آشنا بشین. اگه علاقه دارید از [این لینک](#) می‌تونین اطلاعات بیشتری از این آقا بخونین.

رتینگ

هر کدوم از سایت‌هایی که معرفی کردیم، روشی برای امتیازدهی به کاربران دارند. در کدفورسز این امتیاز رو میشه توی پروفایل هر شخص جلوی قسمت rating دید. این رتینگ بر اساس عملکرد شخص توی کانتست‌های rated (که بعدا معرفی می‌کنیم) تغییر می‌کنه. برای آشنایی بیشتر با نحوه‌ی محاسبه‌ی رتینگ می‌تونین [این لینک](#) رو ببینین.

علاوه‌بر رتینگ، هر شخص یک لقب توی کدفورسز داره که بر اساس محدوده‌ی رتینگ اون شخص مشخص می‌شه. به طور مثال لقب expert متعلق به کاربرانی هست که رتینگ بین ۱۶۰۰ تا ۱۸۹۹ دارن. برای آشنایی بیشتر با این القاب می‌تونین [این لینک](#) رو ببینین. لقب‌ها توی community برنامه‌نویسان رقابتی معمولا موضوع مهمی محسوب میشه و نشون‌دهنده‌ی مهارت اون فرد. با توجه به اینکه تغییر رتینگ توی کدفورسز تا حد خوبی متناسب با قابلیت‌های شرکت‌کننده‌هاست و fair محسوب میشه، این رتینگ و لقب‌ها تا قابل استناد هستن. مثلا کسی که لقب master رو داره واقعا خبره محسوب میشه و کمپانی‌ها و اساتید دانشگاهی که این موضوع براشون اهمیت داره می‌تونند به این رتینگ اعتماد کنند.

نکته‌ی مهمی که وجود داره اینه که این رتینگ تقریبا هیچ نسبیتی با تعداد کانتست‌هایی که توش شرکت می‌کنید نداره! یه نفر ممکنه ۱۰۰ تا کانتست شرکت کنه ولی در نهایت از کسی که ۱ کانتست شرکت کرده رتینگ کمتری داشته باشه. همین یکی از دلایل fair بودن رتینگ هست. تنها در صورتی می‌تونین رتینگ‌تون رو افزایش بدین که الگوریتم‌های جدید و روش‌های جدید یاد بگیرید و با حل‌کردن سوال‌های جدید و بالاتر از سطح خودتون (سوالاتی که در زمان کانتست نتونستید حل کنید)، پیشرفت کنید. هیچ راه میانبری وجود نداره.

کانتست‌ها

کانتست‌های مختلفی توی کدفورسز برگزار میشه. در حالت کلی کانتست‌هایی که برگزار میشن دو دسته‌اند، کانتست‌های rated و کانتست‌های unrated. همون‌طور که از اسمشون پیداست کانتست‌های rated اون‌هایی هستن که باعث تغییر در رتینگ ما میشن. این کانتست‌ها برای همه rated نیستن و این بستگی به نوع کانتست داره. به طور مثال کانتست‌های educational برای کاربرانی rated هست که رتینگ اون‌ها از ۲۱۰۰ بیشتر نباشه. این باعث میشه که هر نوع کانتست سختی خاص خودش رو داشته باشه.

کدفورسز ۴ نوع کانتست اصلی برگزار می‌کنه: [Div1](#)، [Div2](#)، [Div3](#) و کانتست‌های [educational](#).

مهم‌ترین تفاوت این کانتست‌ها در قوانین اون‌هاست و اینکه برای چه محدوده‌ای از رتینگ rated محسوب میشن.

کانتست‌های Div قوانین مشابهی دارن و تنها تفاوت اون‌ها در محدوده‌ی rated بودنشونه.

- کانتست‌های Div3 برای کاربران با رتینگ زیر ۱۶۰۰ rated هستن. این به این معنی نیست که بقیه نمی‌تونند توی این

کانتست‌ها شرکت کنند، بلکه به این معنی هست که بعد از کانتست فقط رتینگ شرکت‌کننده‌هایی که در این محدوده

باشن تغییر می‌کنه و برای بقیه اتفاقی نمیفته و حتی توی اسکوربورد official هم نشون داده نمیشن.

- کانتست‌های Div2 برای کاربران با رتینگ زیر ۲۱۰۰ rated هستن.

- کانتست‌های Div1 برای کاربرانی که رتینگ بالاتر یا مساوی ۱۹۰۰ دارن rated محسوب میشن که خب یعنی

سخت‌ترین کانتست در بین کانتست‌هاست.

برای آشنایی بیشتر با قوانین کانتست‌های Div و نحوه امتیازدهی به سوالاتی که در این کانتست‌ها حل می‌کنید [این لینک](#) و [این لینک](#) رو ببینید.

- کانتست‌های educational هم برای کاربران با ریتینگ زیر ۲۱۰۰ rated هستن. این کانتست‌ها قوانین متفاوتی دارند.

نحوه رنگ‌شدن توی اسکوربورد و امتیازدهی به سوالات حل‌شده در این کانتست‌ها بر اساس قوانین ACM-ICPC هست که می‌تونید [این لینک](#) رو برای اطلاعات بیشتر ببینین.

کانتست‌های Div و educational دو تفاوت مهم دارن:

1. نحوه امتیازدهی به سوالات: در کانتست‌های Div هر سوال با توجه به سختی که داره امتیاز مخصوص خودش رو داره و امتیاز هر سوال چند دقیقه قبل از شروع کانتست اعلام می‌شه ولی در کانتست‌های educational به سبک مسابقات ACM-ICPC وزن سوالات یکسانه.
2. تفاوت در هک کردن در حین کانتست: در کانتست‌های Div شرکت‌کننده‌ها در طول کانتست میتونند راه‌حل‌های همدیگه رو هک کنن که این باعث گرفتن امتیاز میشه. در حالی که در کانتست‌های educational در حین کانتست همچین قابلیتیه وجود نداره ولی بعد از اتمام کانتست به مدت ۲۴ ساعت شرکت‌کننده‌ها میتونند راه‌حل‌های همدیگه رو هک کنند که اینجا هیچ امتیاز مثبتی به همراه نخواهد داشت. (اگر متوجه نشدید قسمت هک رو بخونین)

مفهوم هک

هک یکی از اساسی‌ترین بخش‌های کانتست‌های کدفورسز هست. در کانتست‌های Div زمانی که شما یک سوال رو حل کردید و Accept گرفتید، میتونید سوال رو lock کنین (توی صفحه‌ی problems کانتست با استفاده از آیکون قفل کنار سوال) که این به این معنیه که شما تا آخر کانتست نمی‌تونید کد دیگه‌ای برای این سوال بفرستید. پس از lock کردن سوال، در بخش room می‌تونید کد اون سوال یک‌سری شرکت‌کننده‌ی دیگه رو ببینید و اگه باگی توش پیدا می‌کنید با ساختن ورودی-خروجی‌ای که برای اون کد جواب غلط می‌ده، سوال اون شرکت‌کننده رو هک کنید.

در این پروسه ما نمی‌تونیم کدی که می‌خواهیم هک کنیم رو کپی کنیم؛ بلکه کدفورسز عکس کد رو به ما نشون می‌ده و برای هک کردن مجبوریم که کد رو بخونیم، بفهمیم و سعی کنیم توش باگ پیدا کنیم؛ که خود این پروسه میتونه خیلی آموزنده باشه. این که بتونیم یه کد رو بخونیم، بفهمیم و توش باگ پیدا کنیم قابلیت مهمی محسوب میشه.

اگه باگی توی یک کد پیدا کردید و تصمیم گرفتید برای هک اون اقدام کنید دو حالت پیش میاد؛ اگه شما با موفقیت کد رو هک کردید ۱۰۰ امتیاز مثبت می‌گیرید که امتیاز زیادی محسوب میشه (فرض کنین سوال A کانتست ۵۰۰ امتیاز داره، پس ۱/۵ برابر حل سوال A امتیاز می‌گیرین) در این صورت شرکت‌کننده‌ای که کدش هک شده بهش اعلام می‌شه و اگه اون سوال رو lock نکرده باشه می‌تونه دوباره کد دیگه‌ای ارسال کنه. نکته‌ی مهم اینه که ورودی-خروجی‌ای که کد باهاش هک شده به شرکت‌کننده اعلام نمیشه و خودش باید باگ راه‌حلش رو متوجه بشه، ولی اگه هک شما ناموفق باشه، ۵۰ امتیاز منفی می‌گیرید.

برای این که متوجه بشین چرا هک کردن نکته‌ی خیلی مهمی محسوب می‌شه بخش بعدی رو بخونید.

اتفاقاتی که بعد از هر کانتست میفته

توی کانتست‌های کدفورسز حتی اگه در حین کانتست سوالی رو حل کنیم و Accept بگیریم، نمی‌تونیم مطمئن باشیم که بعد از کانتست هم سوال ما Accept بشه. کدفورسز دو نوع تست‌کیس داره، یکی pretests که در حین کانتست اگر شما اون‌ها رو pass

کنین Accept می‌گیرید، یکی هم تست‌های نهایی که پس از پایان کانتست همه‌ی راه‌حل‌ها روی اون‌ها امتحان میشن و اگر اون‌ها هم pass بشن در نهایت کد Accept میشه، در غیر این‌صورت امتیاز اون سوال کسر میشه. Pretest ها معمولا تست کیس‌های خوبی هستند ولی با توجه به این مکانیسم، باید دقت دوجندانی کرد که بعد از کانتست امتیازی از ما کسر نشه. تموم تست کیس‌هایی که شرکت‌کننده‌ها با استفاده از اون‌ها هک کردند در نهایت در بین تست‌های نهایی وجود داره. پس اگه کد ما باگ داشته باشه، بهترین حالت اینه که در حین کانتست هک بشه تا متوجه باگ راه‌حل بشیم و برطرفش کنیم تا این که بعد از کانتست متوجه باگ راه‌حل بشیم و امتیاز از دست بدیم. دلیل مهم بودن قابلیت هک کردن این هست که باعث میشه کاربران زیادی تست‌های tricky مختلفی برای هک تولید کنند که در نهایت پس از کانتست روی همه‌ی راه‌حل‌ها تست میشه و باعث میشه فقط راه‌حلی که درست است Accept بشه.

خب بعد از توضیح بالا، الان می‌تونیم اتفاقاتی که بعد از اتمام کانتست اتفاق میفته رو بررسی کنیم:

- بعد از اتمام کانتست وضعیت کانتست به testing تغییر پیدا می‌کنه که این به این معنیه که راه‌حل‌ها در حال تست روی تست‌های نهایی هستن. در این زمان میشه در صفحه‌ی کانتست درصد پیشرفت این وضعیت رو دید.
- پس از تموم شدن testing، نتایج نهایی مشخص میشه و وضعیت کانتست به final standing تغییر پیدا می‌کنه. در این حالت دقایقی طول می‌کشه که بر اساس عملکرد شرکت‌کننده‌ها تغییرات ریتینگ مشخص بشه.
- میشه گفت مهم‌ترین اتفاقی که در فاصله‌ی چند ساعت بعد از کانتست میفته، انتشار ادیتوریل (راه‌حل سوالات کانتست) هست. پس از انتشار، لینک اون توی پست announcement کانتست و قسمت contest materials صفحه‌ی کانتست قرار گرفته می‌شه. به طور مثلا [این پست announcement](#) یکی از کانتست‌ها رو ببینید و به بخش [Editorial is](#) UPD: [out](#) توجه کنید.
- به فاصله‌ی چند ساعت از پایان کانتست سختی سوالات کانتست محاسبه میشن (با توجه به میزان ریتینگ کسانی که در کانتست سوال رو حل کردند) و از توی بخش tag اون سوال قابل مشاهده هست. برای نمونه [این سوال](#) که سختی ۲۲۰۰ داره رو می‌تونید ببینید. سختی سوالات ارتباط مهمی با ریتینگ داره. به طور مثال اگر سختی یک سوال ۱۸۰۰ باشد به این معنی‌ست که یک کاربر با ریتینگ حدود ۱۸۰۰ به احتمالاً ۱/۲ در کانتست میتونه این سوال رو حل کنه (عملا به این معناست که اگر ریتینگ شما ۱۸۰۰ هست سوال سختی نیست براتون). برای آشنایی بیشتر با نحوه‌ی محاسبه سختی سوال و ارتباط اون با ریتینگ [این لینک](#) رو ببینید.

نکته‌هایی در مورد ادیتوریل

اول اینکه باز هم تاکید می‌کنم که یادگیری و پیشرفت زمانی اتفاق میفته که سوالاتی که توی کانتست نتونستید حل کنید و سطحش از سطح کنونی شما بالاتر هست رو بررسی کنید و حل کنید که خب در این جا به اهمیت ادیتوریل پی می‌بریم. ادیتوریل سوالات معمولا کوتاه هست و ۰ تا ۱۰۰ رو توضیح نداده که طبیعی و منطقی هم هست. به طور مثال راه‌حل سوال D یکی از کانتست‌ها رو از [این لینک](#) ببینید. راه حل، بدون هیچ مقدمه‌ای گفته که این سوال dp هست و recurrence اون رو هم معرفی کرده و یک توضیح مختصری درباره‌ی رسیدن به این recurrence داده. ولی به طور مثال اینکه dp چی هست و recurrence توی dp به چه معناست رو توضیح نداده. در حالت کلی توی ادیتوریل سوال ممکنه با چندین موضوع جدید که بلد

نیستیم و توضیح هم داده نشده رویه‌رو بشیم (که این چیز خوبیه). در این صورت باید یک منبع خوب برای این موضوعات پیدا کنیم و یاد بگیریم‌شون. که خب چون یک سوال از اون مبحث داریم که می‌خوایم حلش کنیم این پروسه هدف‌مندتر میشه. حالا سوالی که پیش میاد این هست که منبع چجوری پیدا کنیم. یکی از بهترین راه‌ها سرچ کردن توی خود کدفورسز هست. معمولاً توضیحات خیلی خوبی به عنوان پست بلاگ درباره‌ی موضوعی که می‌خوایم توی کدفورسز پیدا می‌کنیم. حتی اگر به صورت مستقیم پستی در این باره منتشر نشده باشه، لینک به منابع مفید در سایت‌های دیگه پیدا میشه. به طور مثلاً فرض کنید من می‌خوام dp یاد بگیرم. اول [توی کدفورسز سرچ می‌کنم](#). از این سرچ به [این پست بلاگ](#) می‌رسیم که توی کامنت‌هاش لینک‌های زیادی به سوالات و منابع dp وجود داره، به طور مثال [این لینک](#).

پروفایل

پروفایل هر شخص شامل اطلاعاتی مثل rating، نموداری از پیشرفت او در طول کانتست‌ها، محل تحصیل، ملیت و کامنت‌های اوست. می‌تونین این بخش‌ها رو در [پروفایل tourist](#) مشاهده کنین. هم‌چنین در بخش [blog](#) پست‌های این کاربر رو میتونین ببینین و در بخش [submissions](#) تمام کدهایی که سابمیت کرده قابل دسترسیه.

اسکوربرد

در صفحه‌ی اسکوربرد هر کانتست می‌شه امتیازات شرکت‌کنندگان رو مشاهده کرد. هم‌چنین پس از اتمام هر کانتست می‌شه با کلیک بر روی امتیاز هر فرد از هر سوال به سابمیت‌ها و کدهای اون فرد دسترسی پیدا کرد. به طور مثال می‌تونین اسکوربرد [Codeforces Round #545 - Div1](#) رو مشاهده کنین.

problemset

در بخش [problemset](#) تمامی سوالات کانتست‌های برگزار شده به ترتیب از آخرین کانتست قابل دسترسی‌اند. هم‌چنین در نوار بالای لیست سوالات یک فیلتر برای مرتب‌کردن بر اساس درجه‌ی سختی و یک فیلتر برای مرتب‌کردن براساس تعداد اکسپت‌های سوالات وجود داره. سمت راست صفحه هم یک فیلتر برای مشخص کردن محدوده‌ی سختی سوال وجود داره. همون‌طور که احتمالاً دیدین هر سوال چند تگ بر اساس مباحث مربوط به اون سوال داره. لیست تمامی تگ‌های موجود رو در این [لینک](#) می‌تونین ببینین. بنابراین سوالات هر تگ رو هم به صورت جداگونه می‌تونین دسترسی داشته باشین.