

۱ طراحی EZ

داده ساختاری طراحی کنید که اعمال زیر را بهینه انجام دهد.

- یک عدد را به انتهای لیست اضافه کند
- یک عدد را از انتهای لیست کم کند
- k عنصر انتهایی لیست را قرینه کند که k برای داده ساختار همیشه یک عدد ثابت است (در عمل نتیجه مثل این است که k عنصر را به ترتیب بخواند و روی هم بریزد، سپس آن ها را وارونه کند و سپس به لیست برگرداند).
- عناصر را به ترتیبی که در لیست قرار دارند چاپ کند

۲ استریم محمد

MamadTV سرویسی است که به بینندگان این امکان را می دهد که در حین تماشای بازی های ویدیویی که توسط استریمر پخش می شوند، چت کنند. هر بیننده با یک شناسه عدد صحیح منحصر به فرد شناسایی می شود. چت از جریانی از پیام هایی که بینندگان نوشته اند تشکیل شده است. بینندگان می توانند جدیدترین پیام های چت را بر اساس اندازه صفحه نمایش خود مشاهده کنند. اگر بیننده ای در چت بد رفتار کند، ممکن است توسط پخش کننده ممنوع شود. هنگامی که یک بیننده ممنوع است، نه تنها دیگر نمی تواند پیام ارسال کند، بلکه تمام پیام های قبلی او از چت حذف می شود.

یک ساختمان داده داده را برای پیاده سازی کارآمد چت *MamadTv* توصیف کنید، که از عملیات زیر پشتیبانی می کند، جایی که n تعداد همه بینندگان (ممنوع یا غیرممنوع) است (همه عملیات باید در بدترین حالت باشد):

- $build(V)$: این عملیات یک اتاق گفتگوی جدید را با $n = |V|$ بیننده مقداردهی اولیه می کند در V در زمان $O(n \log n)$
- $send(v, m)$: این عملیات به بیننده v اجازه می دهد تا در زمان $O(\log n)$ پیام m را به چت ارسال کند (مگر اینکه ممنوع شده باشد).
- $Recent(k)$: این عملیات k جدیدترین پیام های حذف نشده (یا همه پیام ها اگر کمتر از k باشد) را در زمان $O(k)$ برمی گرداند.
- $ban(v)$: این عملیات بازدید کننده v را ممنوع می کند و تمام پیام های آنها را در زمان $O(nv + \log n)$ حذف می کند.

۳ DeleteMin

داده ساختاری طراحی کنید که بتواند اعمال $Pop, Push, FindMin$ (یافتن و برگرداندن کوچک ترین عنصر) را در $O(1)$ انجام دهد. سپس با فرض اینکه می دانیم نمیتوان یک آرایه را در حالت کلی در بهتر از $O(n \log n)$ مرتب کرد، ثابت کنید اگر این داده ساختار بخواند عمل $DeleteMin$ هم پشتیبانی کند، نمیتواند آن را هم در مرتبه $O(1)$ انجام دهد.

۴ لیست پیوندی دوگانه

یک لیست پیوندی دوتایی یک ساختار داده با $node$ هایی است که $pointer$ به $node$ های قبلی و بعدی در دنباله دارند. لیست با دو نشانگر $L.head$ و $L.tail$ تعریف می شود که به اولین و آخرین گره اشاره می کنند. بر خلاف لیست های دارای پیوند منفرد، لیست های دارای پیوند دوگانه نیازی به پیگیری طول آنها ندارند.

۱. یک لیست پیوندی دوگانه همانطور که در بالا توضیح داده شد را دارید، الگوریتم هایی را برای اجرای عملیات دنباله زیر، هر کدام در زمان $O(1)$ توصیف کنید.

۲. دو $node$ $X1, X2$ را داریم از یک لیست با پیوند دوگانه L ، که در آن $X1$ قبل از $X2$ قرار می گیرد، یک الگوریتم زمان ثابت برای حذف تمام گره ها از $X1$ تا $x2$ از L را توصیف کنید، و آنها را به عنوان یک لیست جدید با پیوند دوگانه برگردانید.