

بیمارستان شاهد

دکتر علی شفیعی 😊 در بیمارستان شاهد کار می کند. سیستم مدیریت بیماران دکتر شفیعی به شما سپرده شده است. در مطب دکتر شفیعی، به هر بیماری که وارد می شود بسته به مشکلی که دارد یک عدد نسبت داده می شود.

هر چقدر این عدد **بزرگتر** باشد یعنی کار این بیمار اورژانسی تر است و به سرعت باید به کار آن رسیدگی شود.

می خواهیم داده ساختاری داشته باشیم که در هر لحظه، **بیمار با بیشترین اولویت** را به ما نشان بدهد.

این داده ساختار دارای اعمال زیر است:

- اضافه کردن یک بیمار جدید به صف
- نشان دادن بیمار با بیشترین اولویت
- خارج کردن بیمار با بیشترین اولویت از صف و ویزیت کردن آن

در مطبی که دکتر شفیعی در آن کار می کند دکتر انصاری نیز کار میکند. متأسفانه دکتر انصاری بسیار بی انضباط است و چون شب ها تا دیر وقت مشغول علافی است صبح خواب می ماند.

از آنجایی که بیماران به دکتر انصاری اعتراض کرده اند و در آستانه اخراج است، دکتر انصاری از دکتر شفیعی خواهش کرده که روز هایی که او در محل کارش حاضر نمی شود به دکتر شفیعی زنگ بزند و بیماران او را دکتر شفیعی ویزیت کند. او ممکن است در اواسط روز کاری تماس بگیرد و اطلاع دهد که در محل کار حاضر نمی شود، بنابراین در این روزها نیاز داریم صف دکتر انصاری و صف دکتر شفیعی را ادغام (merge) کنیم.

دکتر انصاری از شما نیز خواستار است این عملیات را به داده ساختار تان اضافه کنید. به این صورت که دو شیء از داده ساختارمان را گرفته، آنها را با هم merge کند و شیء جدیدی از این داده ساختار باز گرداند.

در ازای این لطف شما، دکتر انصاری در روز امتحان ساختمان داده قرار است به همه ما شیرینی بدهد.

بخش امتیازی:

1. فرض کنید در زمان ادغام دو شیء نیاز داریم که عناصر یک شیء اولویت بیشتری نسبت به شیء دیگر داشته باشد و برای اینکار نیاز داریم به تمام عناصر درون شیء ای که اولویت بیشتری دارد k واحد اضافه کنیم. (k میزان بیشتر بودن اهمیت آن را تعیین میکنید). یعنی اگر در صفهان عناصری داریم که اعداد 1 5 2 به آنها نسبت داده شده است، پس از اعمال این عملیات با $k=2$ عناصرمان به 3 7 4 تبدیل می شوند. (به هر عنصر k واحد اضافه شد)

سوال: اگر عمل merge را نداشته باشیم، می توان عمل اضافه کردن k واحد به همه عناصر را بهینه تر کرد یا خیر؟ به این سوال فکر کنید. در زمان ارائه این سوال از شما پرسیده می شود.

2. ساختار کلاستان را محدود به `int` نکنید و `generic` بنویسید (البته پایتون به خودی خود `generic` هست ولی ممکنه بخشی از کدتان طوری نوشته شود که تنها به اعداد محدود شوید). یعنی کلاس `MaxHeap` به ازای هر نوع داده ای که عملکرد های مقایسه ای و همچنین عمل جمع روی آن تعریف شده کار کند.

نمونه ای از کلاسی که باید پیاده سازی کنید در این [لینک](#) قرار دارد. این ساختار برای راهنمایی شماست و اگر از ساختار دیگری استفاده کنید مشکلی ندارد ولی باید نام کلاستان `MaxHeap` بوده و توابع `extract_max`، `find_max`، `merge` و `add_to_all` را با همین signature داشته باشید.

نکات مهم

- تمیز و منطقی بودن کدتان اهمیت دارد.
- پیچیدگی زمانی و فضایی هر عملیات اهمیت دارد.
- نوشتن Document برای پروژه (markdown) و قرار دادن پروژه در Github نمره امتیازی دارد.

امتیازدهی

- اضافه کردن یک بیمار جدید به صف: 100 امتیاز
- خارج کردن بیمار با بیشترین اولویت از صف و ویزیت کردن آن: 100 امتیاز
- ادغام دو شیء: 100 امتیاز
- اضافه کردن k واحد به تمام عناصر یک شیء: 70+ امتیاز
- تمیز بودن کد: 50 امتیاز
- Document: 15+ امتیاز
- Github: 5+ امتیاز

(موفق باشید :)