

به نام خدا

تمرین چهارم

تاریخ ارسال 1399/9/17

تاریخ تحویل 1399/9/30

1- برای تمرین های عملی خود می توانید از هر زبانی که مسلط تر هستید استفاده نمایید.

2- تمرین های خود را حتما از طریق hws ارسال نمایید.

3- تمرین های خود را زیپ کرده و نام فایل زیپ خود را به فرمت زیر تغییر دهید.

DS-991-firstname_lastname-name project

تمرین تئوری

مسئله ۱.

[15 نمره]

فرض کنید یک جدول درهم سازی با ظرفیت ۱۰ داریم و می خواهیم مقادیر زیر را به ترتیب از چپ به راست در جدول درهم سازی وارد کنیم.

۱۳, ۳۳, ۳۴, ۲۴, ۲۳, ۷۲, ۱۵

در هر بخش جدول درهم سازی نهایی را رسم کنید و تعداد برخورد را بنویسید. تابع درهم سازی اولیه را رقم دهگان ($h(۱۳) = ۱$) و تابع درهم سازی ثانویه را (در صورت لزوم) رقم یکان ($g(۱۳) = ۳$) در نظر بگیرید.

الف) اگر جدول از کاوش خطی استفاده کند.

ب) اگر جدول از کاوش مربعی با پارامتر $C_1 = 0$ و $C_2 = 2$ استفاده کند.

پ) اگر جدول از درهم‌سازی دوتایی استفاده کند.

۹	۸	۷	۶	۵	۴	۳	۲	۱

مسئله ۲.

[5 نمره]

روش درهم‌سازی کوکو^۱ یک روش درهم‌سازی بر پایه‌ی آدرس‌دهی باز است که با استفاده از دو تابع درهم‌سازی، کلید را به گونه‌ای ذخیره می‌کند که زمان جستجو در جدول در بدترین حالت نیز $O(1)$ است. این روش درهم‌سازی را به صورت مختصر توضیح دهید.

مسئله ۳.

[16 نمره]

می‌خواهیم آرایه‌های زیر را مرتب کنیم. برای هر کدام از آنها از کدام یک از الگوریتم‌های مرتب‌سازی استفاده کنیم تا در سریع‌ترین زمان آرایه را مرتب شده داشته باشیم (n را عدد صحیح بسیار بزرگی فرض کنید و $k < \log n$).

- یک آرایه از n عنصر که به طور کاملاً تصادفی در آن قرار دارند.
- یک آرایه از n عنصر که همه‌ی آنها به جز k عنصر که به طور تصادفی در آرایه پخش شده‌اند و در جای خود قرار ندارند، مرتب شده‌اند (در صورت حذف این k عنصر آرایه مرتب است).
- یک آرایه از n عنصر که همه‌ی آنها به جز k جفت عنصر همسایه که به صورت تصادفی در آرایه انتخاب شده و با هم جابه‌جا (swap) شده‌اند، مرتب‌اند (هر عنصر حداکثر جزئی از یک جفت است).
- آرایه‌ای از n عنصر که همه اعدادی صحیح و تصادفی از ۰ تا k هستند.

مسئله‌ی ۴.

[10 نمره]

در یک نوع نادر از مرتب‌سازی سریع، ابتدا از میان n عنصر، $1 + 2\sqrt{n}$ عنصر اول آن را انتخاب کرده و با الگوریتم مرتب‌سازی درجی آنها را مرتب می‌کنیم. محور را برابر میانه‌ی قسمت مرتب شده در نظر گرفته و در ادامه مشابه مرتب‌سازی سریع عمل می‌کنیم. بدترین زمان اجرای الگوریتم فوق با محور عجیب انتخاب شده را به دست آورید.

مسئله‌ی ۵.

[9 نمره]

یک خیابان k چراغ دارد که با شماره‌های 0 تا $k - 1$ مشخص شده‌اند. آرایه‌ی A شامل n عنصر متمایز است و هر عنصر آن برابر با شماره‌ی یکی از چراغ‌ها است. اگر شماره‌ی چراغی در آرایه‌ی A موجود باشد، آنگاه آن چراغ روشن خواهد بود و در غیر این صورت نقطه‌ی تاریک نامیده می‌شود. هم‌چنین می‌دانیم که در هر لحظه حداقل یک نقطه‌ی تاریک وجود دارد.

الگوریتمی از مرتبه‌ی زمانی $O(n)$ و حافظه‌ی اضافی $O(1)$ ارائه دهید تا یک نقطه‌ی تاریک را پیدا کند. هم‌چنین فرض کنید امکان اضافه کردن اطلاعات اضافی به آرایه را نداریم و تنها می‌توانیم اعضای آن را با هم جابه‌جا کنیم.

مسئله‌ی ۶.

[10 نمره]

روی یک جاده تعدادی روستا قرار دارند که پراکندگی آن‌ها از قاعده‌ی خاصی پیروی نمی‌کند. می‌خواهیم یک پمپ بنزین در یکی از روستاها بسازیم تا مجموع فاصله‌ی همه‌ی روستاها از پمپ بنزین کمینه شود. نشان دهید که بهترین روستا برای این کار روستای میانه است.

مسئله‌ی ۷.

[5 نمره]

نشان دهید که دومین کمینه از میان n عنصر را میتوان در بدترین حالت با $2 - \lceil \log n \rceil + n$ مقایسه یافت.

مسئله‌ی ۸.

[10 نمره]

فرض کنید هزینه‌ی درج و حذف در یک آرایه از مرتبه‌ی $O(1)$ باشد. برای این‌که مطمئن باشیم آرایه‌ی ما همیشه جای خالی به اندازه‌ی کافی دارد و همچنین حافظه‌ی زیادی هدر نمی‌دهد، از قواعد زیر استفاده می‌کنیم:

- بعد از هر درج، اگر بیش از $\frac{3}{4}$ آرایه پر باشد، یک آرایه جدید از حافظه می‌گیریم که اندازه‌ی آن دو برابر بزرگ‌تر از آرایه فعلی باشد. حال تمام عناصر آرایه فعلی را در آرایه جدید درج می‌کنیم و سپس آرایه فعلی را آزاد می‌کنیم.
- بعد از هر حذف، اگر کمتر از $\frac{1}{4}$ آرایه پر باشد، آرایه جدیدی از حافظه می‌گیریم که اندازه‌ی آن نصف اندازه‌ی آرایه فعلی باشد و تمام عناصر آرایه فعلی را در آن درج می‌کنیم و سپس آرایه فعلی را آزاد می‌کنیم.

نشان دهید، به ازای هر رشته از درج و حذف‌ها همچنان، هزینه‌ی زمانی سرشکن هر عملیات از مرتبه‌ی $O(1)$ می‌ماند. (راهنمایی: روش‌هایی ساده‌تر از روش پتانسیل وجود دارد! :)

تمرین عملی

[20 نمره]

. یک آرایه از n عدد طبیعی داریم. می‌خواهیم اعضای آن را با استفاده از الگوریتم *quicksort* مرتب کنیم.

ورودی:

در خط اول ورودی ابتدا عدد n نشان‌گر تعداد اعداد آمده است.

در n خط بعدی در هر کدام یک عدد می‌آید که نشان‌گر یکی از اعداد آرایه است.

خروجی:

در خروجی آرایه‌ی مرتب شده را در یک سطر چاپ کنید.

محدودیت‌ها

$$n \leq 100000$$

همه‌ی اعداد ورودی در *int* جا می‌شوند.

ورودی و خروجی نمونه:

stdout	stdin
2 4 5 5 7 8 9 10 14 100	10
	100 14 10 5 2 7 4 9 5 8