



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

طراحی الگوریتم
(پاییز ۱۴۰۱)

تمرین دوم

مهلت تمرین:

۱۴۰۱/۱۰/۰۲

ساعت ۱۱:۵۹

استاد درس : دکتر مهدی جوانمردی

سوال ۱ :

نشان دهید که چگونه می‌توان quicksort را در زمان $(O(\log(n)))$ در بدترین حالت اجرا کرد، با این فرض که همه عناصر متمایز هستند.

سوال ۲ :

در ارتباط با الگوریتم مرتب‌سازی Quicksort به سوالات زیر پاسخ دهید:

الف) نشان دهید چگونه می‌توان با فرض یکتا بودن همه‌ی اعضای یک آرایه، Quicksort را به شکلی تغییر داد که آرایه را در بدترین حالت در زمان مرتب کند.

ب) احتمال آنکه الگوریتم مرتب‌سازی Randomized Quick Sort n عنصر را در زمان $\Omega(n^2)$ مرتب کند چقدر است؟

الگوریتم QUICKSORT زیر شامل دو فراخوانی بازگشتی با خودش است. پس از فراخوانی PARTITION، زیر آرایه سمت چپ و سپس زیر آرایه راست به صورت بازگشتی مرتب می‌شوند. به دومین فراخوانی بازگشتی در QUICKSORT در واقع نیازی نیست، زیرا می‌توان با استفاده از یک ساختار کنترل‌کننده‌ی تکراری آن را انجام داد. این تکنیک به نام tail recursion به طور خودکار توسط کامپایلرهای خوب ارائه می‌شود. نسخه زیر از quicksort را در نظر بگیرید که شبیه‌سازی tail recursion است:

```
QUICKSORT'(A, p, r)
1  while p < r
2      do ▷ Partition and sort left subarray.
3          q ← PARTITION(A, p, r)
4          QUICKSORT'(A, p, q - 1)
5          p ← q + 1
```

ج) حالتی را توصیف کنید که عمق پشته QUICKSORT' برابر با $\Theta(n)$ است به ازای آرایه ورودی با n عضو.

د) کد QUICKSORT را به گونه‌ای تغییر دهید که با حذف زمان اجرای الگوریتم در مرتبه‌ی $O(n \log n)$ بدترین عمق پشته $\Theta(\log n)$ شود.

سوال ۳ :

ثابت کنید هر الگوریتم مرتب سازی مقایسه ای در بدترین حالت نیاز به

$\Omega(n \lg n)$ مقایسه دارد.

سوال ۴ :

قطعه کدی بنویسید که K امین عنصر کوچک آرایه را با استفاده از درخت max heap بدست آورد. مرتبه پیچیدگی زمانی و حافظه آن را بدست آورید.

سوال ۵ :

الگوریتم Radix Sort را طوری تغییر می دهیم که مرتب سازی رقم ها از رقم های پرارزش شروع شود و به رقم های کم ارزش خاتمه یابد. این الگوریتم تغییر یافته را توصیف کنید و زمان اجرای آن را تحلیل کنید.

سوال ۶ :

الگوریتم Counting Sort را به گونه ای تغییر دهید که مرتب سازی اعداد از اول آرایه به آخر انجام شود و خاصیت پایداری برقرار بماند. شبه کد مربوط به این الگوریتم تغییر یافته را بنویسید.

سوال ۷ :

توضیح دهید که چگونه می توان n عدد صحیح در بازه ی 0 تا $n^k - 1$ ($k \in \mathbb{N}$) را در زمان $O(kn)$ مرتب کرد.

سوال ۸ :

توضیح دهید که چگونه می توان الگوریتم counting sort را موازی سازی کرد و در p فرآیند به صورت موازی انجام داد. (پیچیدگی زمانی این روش را هم توضیح دهید)

سوال ۹ :

به عنوان بخشی از یک بازی جدید، شرکت کنندگان به نوبت چندین عدد را بین ۰ تا ۱۰۰۰ حدس میزنند. در هر دور میزبان برنامه، باید بداند که کدام دو حدس به یکدیگر نزدیکتر هستند. یک الگوریتم بهینه از نظر زمانی ارائه کنید که به این پرسش پاسخ می‌دهد. استدلال کنید که این الگوریتم درست است و پیچیدگی زمانی آن را توضیح دهید.

بخش عملی :

برای دیدن تمرین پیاده سازی دوم به [این لینک](#) مراجعه فرمایید.
رمز ورود کلاس: ۰۰۱۰

نکات تمرین:

- فایل تمرین های خود را صورت یک pdf با فرمت "Stunum_HWnum.pdf" نام گذاری کنید.
- به دلیل فشرده بودن زمان تا پایان ترم ها و شرایط پیش آمده در ترم جاری امکان تمدید تمارین وجود ندارد.
- برای تمرینات در مجموع ۵ روز زمان تاخیر وجود دارد. در صورت تاخیر بیشتر به ازای هر روز ۵ درصد از نمره ی کل تمرینات شما کسر می شود.
- در صورت شبیه بودن پاسخ تمارین دانشجویان، نمره تمرین بین دانشجویان با پاسخ مشابه تقسیم خواهد شد. (معیار برای شباهت تمرین های عملی کوئرا است)
- در صورت داشتن هرگونه ابهام و سوال با یکی از راه های زیر ارتباط برقرار کنید.

ایمیل: mohamadchoupan80@gmail.com

تلگرام: [lostago](#)