

Subject:

ترتیب چهارم ساختن داده

Year. Month. Date.

## جواب سوال ۱:

در این قسمت از `insertionsort` استفاده می‌کنیم که بدین صورت  
که در مرحله  $i$ ام عنصر  $i$ ام را با عنصر قبل از خود مقایسه  
می‌کنیم (۱- عنصر قبلی مرتب شده هستند) و اگر عنصر بزرگتر  
بدر جایش همان جا هست در غیر این صورت جای آن  
رو عوض می‌شود و این الگوریتم ادامه دارد.

حالا می‌دانیم که هر عنصر حداکثر ۲۰ عنصر درست جبه خود  
خورد دارد که از آن بزرگتر هستند و چون تمام آن‌ها مرتب  
هستند (خاصیت `insertionsort`) پس هر عنصر باید ۲۰  
بمقایسه انجام دهد.

پس الگوریتم از مرتبه  $O(20n)$  است که پس  $O(n)$  است.

## جواب سوال ۲:

ما یک `max-heap` می‌توان این کار را انجام داد بدین صورت که با  
بزرگترین عنصر که همان آخرین عنصر موجود در درخت از کایست است  
یک درخت `max-heap` درست می‌کنیم.

این درخت با پیچیدگی زمانی  $O(k)$  ساخته می‌شود چنان این  
`max-heap` دارای  $k$  عنصر هست.

حالا یک روند تکراری را شروع می‌کنیم به این صورت که رأس `max-heap`  
را جدا می‌کنیم و به لیست `Sort` شده اضافه می‌کنیم سپس از لیستی که  
آنها

ریشه جدا شده به آن تعلق داشت می‌گردیم و دوباره بزرگترین را جدا می  
کنیم و دوباره `max-heap` را مرتب می‌کنیم که از  $O(k+1)$  هست.

تا جایی که `max-heap` خالی شود که ادامه می‌دهیم. بدترین حالت

زمانی است که به ازای هر بار جدا شدن رأس `max-heap`

مرتب شود که در این صورت از  $O(n \log k)$  هست.

Subject:

تمرین چهارم ساختمان داده

Year. Month. Date.

۱ جواب سوال ۳: در اینجا مقدار  $\alpha$  :  $\alpha = \frac{h}{m}$

حالت عادی { جست و جو :  $O(1 + \alpha)$   
ناموفق :  $O(1 + \alpha)$   
حذف :  $O(1)$   
رج :  $O(1)$

حالت مرتب شده { جست و جو :  $O(1 + \alpha)$   
ناموفق :  $O(1 + \frac{\alpha}{p})$   
حذف :  $O(1)$   
رج :  $O(n)$

۱۵ تفاوت هار چرج و جست و جو ناموفق هست زیرا وقتی  
۱۶ مرتب باشد ممکن هست که تمام لیست باید جست و جو شوند  
۱۷ برای درج کردن که این از مرتبه  $O(n)$  می شود و برای  
۱۸ جست و جو ناموفق کمترین مقدار کافی هست که  
۱۹ اول هر لیست چک شود و اگر ناموفق بود از آن قسمت بگذریم  
۲۰ چون مرتب هستند و به بعدی برویم پس آن کمترین حالت  
۲۱ عادی می شود.



Subject: تدریس چهارم ساختمان داده Year: Month: Date:

۱ جواب سوال ۴:

۳ برای اثبات از استقرا قوی استفاده می‌کنیم:

۵ پایه: اگر  $A$  دارای یک عنصر باشد، آنگاه خود آن صورت شده است.

۸ فرض: اگر  $k$  عنصر داشته باشد به صورتی که  $1 \leq k \leq 2$  باشد،  $A$  به درستی مرتب شده است.

۱۱ حکم:  $A$  با  $n$  عنصر به درستی مرتب شده باشد.

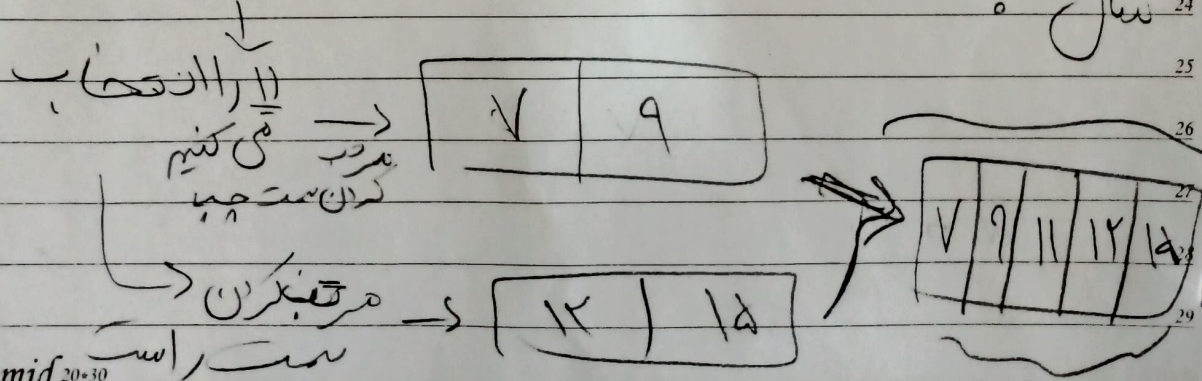
۱۳  $A$  دارای  $n$  عنصر دارد، وقتی که برای اولین بار وارد تابع می‌شویم،  $k$  برابر  $first$  است و پس از دوباره صدا زدن تابع مانند فرض استقرا،  $k-1$  عنصر قبل از  $k$  مرتب شده‌اند. حالا تابع را با مقادیر جدید حذرده  $A$  و  $n$  و  $k+1$  هست صداهای زیر که با فرض استقرا

۱۹ بعد از  $k$  هم مرتب می‌شود (به روش بازگشتی) و در آخر

۲۱ از تابع بیرون می‌آید و کل  $A$  مرتب شده در دست ما

۹	۷	۱۱	۱۵	۱۲
---	---	----	----	----

۲۴ سوال ۵:



Subject:

تمرین چهارم سافتمان داده

Year. Month. Date.

## ۱ جواب سوال ۱۰

۲  
۳ برای حل این قسمت از دو آرایه گلی با طول  $n+1$  کد  
۴ می گیریم.

۵  
۶ در آرایه  $A$  مقادیر این صورت ذخیره می کنیم که  
۷ در خانه  $i$  ام تعداد تکرار  $A$  را داشته باشیم.

۸  
۹ چون هر یکس را باید چک کنیم پس  $O(n)$  می شود.

۱۰  
۱۱ در آرایه  $B$  مقادیر این صورت ذخیره می کنیم که  
۱۲ در خانه  $i$  ام مقدار خانه  $i$  ام  $A$  به علاوه مقدار  $i$  ام  $B$  ذخیره شود.  
۱۳ به صورت اولیه خانه اول را با ۰ جمع می کنیم.

۱۴  
۱۵ چون طول  $k$  باید چک شود پس  $O(k+1)$  نهان  $O(k)$  هست.

۱۶  
۱۷ برای خروجی عدد متناظر  $B$  ام در آرایه  $B$  را می عدد  
۱۸ ذخیره شده در خانه  $i-1$  ام  $A$  می کنیم که از مرتبه  $O(1)$   
۱۹ هست در مجموع الگوریتم  $O(n+k)$  هست.

۲۰  
۲۱  
۲۲  $O(n+k)$   
۲۳  
۲۴  
۲۵  
۲۶  
۲۷  
۲۸  
۲۹



Subject:

Year. Month. Date.

## جواب سوال ۶ و ۸

از الگوریتم زیر برای این سوال استفاده می کنیم که به این صورت که یک عنصر را از انتخاب می کنیم و تمامی اعداد بزرگتر از آن را درست راست می و کوچکتر از آن را درست چپ آن قرار می دهیم. فرض می کنیم بعد از مرتب شدن در جایگاه  $i$  اتم قرار بگیرد که اگر  $i$  از  $\frac{n}{2}$  کوچکتر بود، درست چپ جایگاه  $i$  ادامه می دهیم و اگر از  $\frac{n}{2}$  بزرگتر بود، درست راست جایگاه  $i$  ادامه می دهیم و اگر برابر  $\frac{n}{2}$  بود که میانه یافت شده است.

از الگوریتم quicksort و شباهت آن با این الگوریتم می توان گفت که از میانگین مرتبه از  $O(n)$  است.

$$E[X] = O(n)$$

حال که میانه را پیدا کردیم یک آرایه دیگر در نظر می گیریم که در این آرایه

اختلاف همه اعضا را با میانه بدست می آوریم و در آن قرار می دهیم. (قدر مطلق این مقادیر را باید در آرایه بزرگتریم). این کار هم  $O(n)$  است چون هر عدد یکبار دیده شده است.

حالا همان الگوریتم بالا را در این آرایه پیاده سازی می کنیم که باز هم به طور میانگین از  $O(n)$  می شود. در این قسمت به جای میانه کما کما این عدد کوچکتر از بدست می آوریم.

حالا باید حلقه در آرایه دوم هر عدد که از  $k$  گذشت مسأله بود سؤال جواب نمی شود که باز هم  $O(n)$  هست.

همه قسمت ها  $O(n)$  بود.