



مدرس:

بسمه تعالی

دانشگاه فردوسی مشهد

درس: ساختمان های داده ها
شیرازی

تمرین Deterministic Selection

هدف از این تمرین پیاده سازی الگوریتم DSelect با استفاده از افراز سه راهه است. لازم به یادآوری است که اندیس آماره ترتیبی در تابع select از یک شروع می شود ولی آرایه ها در کد ما از صفر اندیس گذاری می شوند. در این تمرین از دانشجو خواسته شده است که متوذهای زیر را در فایل DSelect.h پیاده سازی کند:

۱- تابع select:

```
virtual T select (T* A, int p, int r, int i){  
    //Write your code here  
}
```

این تابع با فراخوانی تابع choosePivot محوری مناسب انتخاب می کند و سپس بر اساس آن به کمک threeWayPartition آرایه را از اندیس p تا اندیس r به سه قسمت بزرگ تر، مساوی و کوچکتر از محور تقسیم می کند و در نهایت با توجه به خروجی آن ممکن است به صورت بازگشتی خود را برای یکی از قسمت های کوچکتر فراخوانی کند. پارامتر های این تابع به این شرح است:

A: آرایه داده های ورودی

p: اندیس شروع بررسی (که شامل خود p نیز می شود)

r: اندیس پایان بررسی (که شامل خود r نیز می شود)

i: آماره ترتیبی مورد نظر (با شروع از اندیس یک)

خروجی این تابع مقدار آماره ترتیبی i ام (و یا i-1 امین عنصر در لیست مرتب شده داده ها با شروع از اندیس صفر) می باشد.

۲- تابع choosePivot در فایل:

```
virtual T choosePivot(T* A, int p, int r)
```

A: آرایه داده های ورودی

p: اندیس شروع بررسی (که شامل خود p نیز می شود)

r: اندیس پایان بررسی (که شامل خود r نیز می شود)

شبه کد این تابع در اختیار شما قرار داده شده است و کافی است تا قسمت های ؟ را با توجه به کد و توضیحات داخل اسلاید به نحو صحیحی تکمیل کنید تا الگوریتم به درستی کار کند.

* همچنین توصیه می شود تا نحوه عملکرد کد را برای خود بررسی کنید و اگر تفاوتی نسبت به اسلاید ها در آن دیدید دلیل آن را پیدا کرده و ببینید کدام یک از روش ها و به چه دلیلی می تواند بهتر عمل کند.

۳- تابع sortBlock

```
virtual void sortBlock(T* A, int p, int r){  
    //Write your code here  
}
```

این تابع در حقیقت تابعی داخلی در DSelect است که در choosePivot استفاده می‌شود و برای راحتی و تمیز تر شدن کد آن را جدا کرده‌ایم. در این تابع شما باید با کمک Selection Sort بلاک‌ها را (که در این کد اندازه بلاک‌ها را ۵ گرفته‌ایم) مرتب کنید. پارامترهای این تابع به این شرح است:

A: آرایه داده‌های ورودی

p: اندیس شروع مرتب‌سازی (که شامل خود p نیز می‌شود)

r: اندیس پایان پایان مرتب‌سازی (که شامل خود r نیز می‌شود)

* توجه کنید که در این الگوریتم نوع مرتب‌سازی بلاک اهمیت چندانی ندارد، اما از آنجایی که در تست‌ها تعداد دستور انتصاب جهت بررسی مرتبه زمانی الگوریتم شمرده می‌شود، توصیه می‌کنیم تا از Selection Sort استفاده کنید تا با مشکلی مواجه نشوید.

۴- تابع threeWayPartition

```
virtual void threeWayPartition(T* A, T pivot, int p, int r, int& q1, int& q2) {  
    //Write your code here  
}
```

این تابع همان تابع threeWayPartition برنامه Quick Sort است با این تفاوت که در اینجا مقدار محور به عنوان یکی از پارامترهای ورودی دریافت می‌شود. پارامترهای این تابع به این شرح است:

A: آرایه داده‌های ورودی

pivot: مقدار عددی محور است (نه اندیس آن در آرایه A)

p: اندیس عنصر شروع آرایه (که شامل p نیز می‌شود)

r: اندیس عنصر انتهایی آرایه (که شامل r نیز می‌شود)

q1: در آرایه خروجی، اندیس آخرین عنصر کوچکتر از محور را نشان می‌دهد

q2: در آرایه خروجی، اندیس اولین عنصر بزرگتر از محور را نشان می‌دهد

** چون q1 و q2 به صورت reference (با استفاده از علامت &) تعریف شده‌اند و به صورت call by reference ارسال می‌شوند، از آنها به عنوان خروجی‌های تابع استفاده می‌کنیم.

* در این تمرین فرض شده است که شما الگوریتم مشابه این را در تمرین‌های پیشین پیاده‌سازی کرده‌اید و نیازی به تست مستقیم آن نیست. به همین جهت در تستی که در اختیار شما قرار گرفته داده شده است، پیاده‌سازی این قسمت را به صورت مستقیم بررسی نکرده‌ایم. طبیعی است که پیاده‌سازی صحیح این تابع در نحوه عملکرد الگوریتم و همچنین تست‌های سپهر و نمره شما تأثیر مستقیم خواهد داشت.

*** نکته بسیار مهمی که در پیاده‌سازی select باید در نظر بگیرید، این است که الگوریتمی که در اختیار شما قرار داده شده است بر روی آرایه‌های کوچک عمل نمی‌کند. پس باید اندازه‌ای که کوچکتر از آن با مشکل مواجه می‌شود را بیابید و با توجه به آن تابع select را به نحوی پیاده‌سازی کنید تا بر روی آن‌ها نیز درست عمل کند. (سعی کنید دلیل این اشکال را برای خود پیدا کنید و ببینید در صورت درست عمل نکردن select بر روی آرایه‌های کوچک، الگوریتم با چه مشکل جدی‌ای مواجه می‌شود. در نهایت بررسی کنید که با وجود این مشکل و راه حل آن، این الگوریتم چه مزیتی نسبت به پیاده‌سازی عادی می‌تواند داشته باشد)

در این تمرین نحوه ارزیابی به شرح زیر است:

۱. **testOrder**: این تست مهمترین تست این تمرین است و در آن مرتبه زمانی برنامه شما تست می‌شود. برای این کار تعداد دستورات انتصاب شمرده می‌شوند به همین جهت توجه داشته باشید که کدتان بهینه باشد و دستور انتصاب اضافه نداشته باشید. همچنین همانطور که ذکر شد برای مرتب سازی از الگوریتم Selection Sort استفاده کنید تا در این تست به مشکلی نخورید. از آنجایی که هدف اصلی از این تمرین و الگوریتم مرتبه زمانی آن است، این تست ۳۰٪ نمره را به خود اختصاص می‌دهد و پیشنهاد می‌دهم تست‌ها نیز هست. به این معنی که در صورت عدم موفقیت در این تست از تست‌های دیگر نیز نمره‌ای کسب نخواهید کرد.
۲. **testTime**: از آنجا که مهمترین عاملی که به ما کمک کرد تا مرتبه زمانی الگوریتم را به $O(n)$ برسانیم نحوه انتخاب محور بود و بر این اساس می‌دانیم که در بدترین حالت هم این الگوریتم با $O(n)$ اجرا می‌شود. پس در این تست بررسی می‌شود که الگوریتم شما محور خوبی را انتخاب کند و زمان اجرا در هر حالتی به صورت خطی تغییر کند. این تست از اهمیت بالایی برخوردار است، در نتیجه ۳۰٪ از نمره را به خود اختصاص داده است و پیش نیاز تست‌های بعدی نیز هست و در صورتی که از این تست نمره‌ای کسب نکنید، نمره دو تست بعدی را نیز از دست خواهید داد. همچنین دقت کنید که در این تست بررسی می‌شود از کتابخانه دیگری استفاده نشده باشد. پس در صورتی که از کتابخانه‌هایی مانند iostream استفاده کرده‌اید، مطمئن شوید که آن‌ها را حذف کرده باشید و پس از آن تمرین خود را به سیستم سپهر ارسال کنید. در غیر این صورت از این تمرین نمره‌ای دریافت نخواهید کرد.
۳. **testSelect**: در این تست الگوریتم شما بر روی تعدادی آرایه اجرا می‌شود و بررسی می‌شود که آیا عنصر درستی را انتخاب می‌کند یا خیر. این تست ۲۵٪ دیگر از نمره را به خود اختصاص می‌دهد.
۴. **testEqualNumbers**: در نهایت الگوریتم شما را بر روی آرایه‌هایی با مقادیر برابر تست می‌کنیم و بررسی می‌شود تا خطایی وجود نداشته باشد. این تست از اهمیت کمتری برخوردار است و فقط ۱۵٪ نمره را دارد.

برای انجام این تمرین کارهای زیر را انجام دهید:

- ۱- ابتدا در این پوشه فایل info.txt را با مشخصات خود پر کنید.
- ۲- سپس به پوشه src بروید و پروژه را در Visual Studio باز کنید.

- ۳- کد های برنامه را تکمیل کنید.
- ۴- برای تست برنامه cmake را دانلود و نصب کنید. (در هنگام نصب مطمئن شوید cmake را به PATH ویندوز اضافه کرده باشید).
- برای دانلود می‌توانید از این لینک استفاده کنید: <https://cmake.org/download/>
- ۵- در ترمینال مطمئن شوید cmake و gcc نصب شده باشند. (می‌توانید از دو دستور `cmake --version` و `gcc -v` استفاده کنید).
- ۶- افزونه CMake Tools را به vscode اضافه کنید.
- ۷- پوشه test را به تنهایی در vscode باز کنید.
- ۸- با `Ctrl+Shift+P` گزینه CMake: Select a Kit را انتخاب کرده و از آنجا کامپایلر gcc را انتخاب کنید.
- ۹- از سمت چپ، منو CMake را انتخاب کنید و از قسمت بالایی آن Build All Projects را انتخاب کنید. (همچنین می‌توانید با `Ctrl+Shift+P` و انتخاب گزینه CMake: Build نیز این کار را انجام دهید)
- ۱۰- حال با راست کلیک بر روی تست کامپایل شده گزینه Run in Terminal را انتخاب کنید. (می‌توانید این مراحل را از این لینک نیز دنبال کنید: <https://code.visualstudio.com/docs/cpp/cmake-linux>)
- ۱۱- در صورت تمایل تست های بیشتری بنویسید تا از عملکرد برنامه خود اطمینان بیشتری حاصل نمایید.
- ۱۲- برنامه را اجرا کنید و پس از اطمینان از صحت عملکرد آن، با استفاده از کلید `PrtScr` از خروجی برنامه عکس بگیرید.
- ۱۳- عکس را با استفاده از mspaint در پوشه img ذخیره نمایید.
- ۱۴- از پوشه src همه فایل های اضافی که به دلیل کامپایل برنامه بوجود آمده اند را پاک نمایید. (پوشه Debug و فایل با پسوند sdf را حتما پاک کنید زیرا حجم زیادی می گیرند).
- ۱۵- محتویات کل پوشه را به صورت یک فایل zip در آورید.
- ۱۶- مطمئن شوید که وقتی فایل zip را باز می کنید پوشه های src, img و test و همچنین فایل info.txt را می بینید.
- ۱۷- نام این فایل zip را به «شماره تمرین-شماره دانشجویی» تغییر دهید.
- ۱۸- ابتدا این فایل را به سیستم «سپهر» ایمیل کنید تا از نحوه عملکرد برنامه خود بر روی تست های تکمیلی آگاه شوید.
- ۱۹- اشکالاتی را که سیستم «سپهر» مشخص کرده است برطرف نمایید و مجدداً تمرین را به سیستم «سپهر» تحویل دهید.
- ۲۰- مرحله قبل را آن قدر ادامه دهید که از صحت عملکرد برنامه خود اطمینان حاصل نمایید.
- ۲۱- نسخه نهایی فایل zip خود را تهیه نمایید.
- ۲۲- این فایل را از طریق سیستم vu تحویل دهید.

با آرزوی موفقیت