



## ساختمان های داده

تمرین Counting Sort

مدرس: دکتر سید کمال الدین غیاثی شیرازی

در این تمرین از شما خواسته می شود که بدنه چهار تابع Min، Max، Count و Sort را در فایل CountingSort.h با تفاوتی جزئی نسبت به آنچه در کلاس مورد بررسی قرار گرفته است، تکمیل کنید.

در این تمرین برای استفاده از مرتب سازی شمارشی از نوع داده ای که آن را IntegerKey نامیده ایم استفاده می کنیم. با بررسی فایل IntegerKey.h می توانید جزئیات پیاده سازی این نوع داده را بررسی کنید. بنابراین برای استفاده از این مرتب سازی می بایست از IntegerKey یا نوع داده ای که از IntegerKey ارث می برد استفاده کنیم. شما می توانید با بررسی تستی که به عنوان نمونه داده شده (CountingSortTest.cpp) نحوه استفاده از این کلاس را ملاحظه فرمایید.

توابعی که باید پیاده سازی شوند:

```
int Min(IntegerKey* arr, int n){  
    //Write your code here  
}
```

این تابع آرایه ای را از ورودی دریافت می کند و مقدار کمترین عنصر آرایه را به عنوان خروجی بازمی گرداند.

```
int Max(IntegerKey* arr, int n){  
    //Write your code here  
}
```

این تابع آرایه ای را از ورودی دریافت می کند و مقدار بیشترین عنصر آرایه را به عنوان خروجی بازمی گرداند.

پارامترهای این دو تابع به شرح زیر است:

n: تعداد داده های ورودی

arr: آرایه داده های ورودی

```
void Count(IntegerKey* arr, int n, int* C, int k, int min, int max){  
    //Write your code here  
}
```

در این تابع می بایست تعداد هر یک از اشیاء موجود در آرایه ای arr را بشمارید و در آرایه ای C ذخیره کنید. سپس عناصر آرایه ای C را به گونه ای تغییر دهید که در C[i] تعداد عناصر آرایه ای arr که کوچکتر از i هستند ذخیره شود. (توجه کنید که در الگوریتم بررسی شده در کلاس در این مرحله، تعداد عناصر آرایه ای arr که کوچکتر یا مساوی i بودند ذخیره می شد).

```
void Sort(IntegerKey* arr, int n, int* C, int k, int min, int max){  
    //Write your code here  
}
```

در این تابع با استفاده از آرایه ای C که در تابع Count پر شده است، می بایست آرایه ای arr از اشیاء را مرتب کنید. دقت کنید که با توجه به نحوه ی پر شدن آرایه ای C، عملیات مرتب سازی را به گونه ای انجام دهید که پایداری آن حفظ شود.

پارامترهای این دو تابع به شرح زیر است:

n: تعداد داده های ورودی

arr: آرایه ی داده های ورودی

k: اندازه آرایه ای C

C: آرایه کمکی برای مرتب سازی شمارشی

max: مقدار بیشترین عنصر آرایه ای arr

min: مقدار کمترین عنصر آرایه ای arr

نحوه ارزیابی در این تمرین به شرح زیر است:

- (۱) `TestCountingSortMin`: این تست تنها `key` عنصری که `key` با کمترین مقدار را دارد بازمیگرداند و تنها ۵ نمره از ۱۰۰ نمره تمرین را به خود اختصاص داده.
- (۲) `TestCountingSortMax`: این تست نیز همانطور که از اسمش پیداست، عنصری که `key` با بیشترین مقدار را دارد را برمی گرداند و ۵ نمره از ۱۰۰ نمره را به خود اختصاص میدهد.
- (۳) `TestCountingSortCount`: این تست درستی آرایه `count` را می آزماید و ۱۰ نمره از ۱۰۰ نمره را به خود اختصاص میدهد.
- (۴) `TestCountingSortRandomNumbers`: در این تست، درستی مرتب سازی الگوریتم شما با یکی از توابع مرتب سازی زبان `C++` مقایسه میشود و ۱۵ نمره از ۱۰۰ نمره به این تست اختصاص داده شده است.
- (۵) `TestCountingSortSort`: در این تست، مرتب سازی صحیح الگوریتم بررسی میشود، این تست نیز ۱۵ نمره از ۱۰۰ نمره را به خود اختصاص میدهد.
- (۶) `stableTest`: در این تست پایداری الگوریتم مرتب سازی شما بررسی میشود. این تست پیش نیاز دو تست پیشین میباشد و در صورت عدم کسب نمره این تست، از دو تست بالا نیز نمره ای دریافت نخواهید کرد این تست ۲۰ نمره از ۱۰۰ نمره را به خود اختصاص میدهد.
- (۷) `testOrder`: در این تست، خطی بودن زمان مرتب سازی کد شما بررسی میشود. این تست نیز همانند تست بالا پیش نیاز دو تست `TestCountingSortRandomNumbers` و `TestCountingSortSort` میباشد و در صورت عدم کسب نمره از این تست، از دو تست مذکور نیز نمره ای دریافت نخواهید کرد. این تست ۳۰ نمره از ۱۰۰ نمره را به خود اختصاص میدهد.

برای انجام این تمرین کارهای زیر را انجام دهید:

- 1 ابتدا در این پوشه فایل `info.txt` را با مشخصات خود پر کنید.
- 2 سپس به پوشه `src` بروید و پروژه را در `Visual Studio` باز کنید.
- 3 کد برنامه خود را در فایل `CountingSort.h` تکمیل کنید.
- 4 برنامه را بر روی تست(های) داده شده تست نمایید.
- 5 در صورت تمایل تست های بیشتری بنویسید تا از عملکرد برنامه خود اطمینان بیشتری حاصل نمایید.
- 6 برنامه را اجرا کنید و پس از اطمینان از صحت عملکرد آن، با استفاده از کلید `PrtScr` از خروجی برنامه عکس بگیرید.
- 7 عکس را با استفاده از `mspaint` در پوشه `img` ذخیره نمایید.
- 8 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).
- 9 محتویات کل پوشه را به صورت یک فایل `zip` در آورید.
- 10 مطمئن شوید که وقتی فایل `zip` را باز می کنید پوشه های `src`، `img` و `test` و همچنین فایل `info.txt` را می بینید.
- 11 نام این فایل `zip` را به «شماره تمرین-شماره دانشجویی» تغییر دهید.
- 12 ابتدا این فایل را به سیستم «سپهر» ایمیل کنید تا از نحوه عملکرد برنامه خود بر روی تست های تکمیلی آگاه شوید.
- 13 اشکالاتی را که سیستم «سپهر» مشخص کرده است برطرف نمایید و مجددا تمرین را به سیستم «سپهر» تحویل دهید.
- 14 مرحله قبل را آن قدر ادامه دهید که از صحت عملکرد برنامه خود اطمینان حاصل نمایید.
- 15 نسخه نهایی فایل `zip` خود را تهیه نمایید.
- 16 این فایل را از طریق سیستم `vu` تحویل دهید.

با آرزوی موفقیت