



ساختمان داده ها - تمرین سری سوم

- استاد درس: دکتر عبدالرضا میرزایی
- مسئول تمرین: مصطفی دریس پور
- از طریق ایمیل زیر می‌توانید با TA مربوط به این تکلیف در ارتباط باشید.

– mostafa.des.por@gmail.com

قسمت اول:

الزامات:

1. در بخش اول این تکلیف شما باید که یک درخت جست‌وجو دودویی (Binary search tree) با استفاده از فایل‌های که در اختیار شما قرار داده شده است پیاده‌سازی کنید. در این درخت جست‌وجو اعضای تکراری قابل قبول نمی‌باشد. با توجه به فایل‌های که در اختیار شما قرار گرفته شده است کلاس که این درخت در آن پیاده‌سازی شده است از جنس `template` می‌باشد و همچنین برای فراخوانی توابع بازگشتی از `wrapper` ها استفاده می‌شود. (یک تابع `wrapper` تابعی است که در داخل آن یک تابع بازگشتی فراخوانی می‌شود.)

2. فایل‌های که برای شما فراهم شده به شرح زیر می‌باشد:

- یک کلاس کامل `EmptyDataCollectionException` شامل:

`EmptyDataCollectionException.h`

`EmptyDataCollectionException.cpp`

- یک کلاس کامل `ElementAlreadyExistsException` شامل:

`ElementAlreadyExistsException.h`

`ElementAlreadyExistsException.cpp`

- یک کلاس کامل `ElementDoesNotExistException` شامل:

`ElementDoesNotExistException.h`

`ElementDoesNotExistException.cpp`

- یک کلاس کامل `BSTNode` شامل:

`BSTNode.h, BSTNode.cpp`

- یک کلاس کامل `WordPair` شامل:

`WordPair.h, WordPair.cpp`

که در کد تست شما استفاده می‌شود.

- یک `makefile` کامل.

- یک فایل ناقص `testDriver.cpp` که برای تست کد های شما استفاده می‌شود.

شما می‌توانید با تکمیل کد های آن جواب خود را تست کنید.

- یک فایل تست به نام `dataFile.txt` که شامل چند جفت دوتایی کلمه می‌باشد که با : (colon) از هم جدا شده‌اند. (کلمه اول یک کلمه انگلیسی و کلمه دوم معادل [Klingon](#) آن می‌باشد.)

- یک کلاس ناقص `BST` که شامل:

`BST.h, BST.cpp`

و شما باید آن را تکمیل کنید توجه کنید که متد های مثل:

`insert(...), retrieve(..), traverseInOrder(...)`

یک `wrapper` هستند که در آن ها توابع بازگشتی زیر فراخوانی می‌شوند:

`insertR(...), retrieveR(..), traverseInOrderR(...)`

شما باید توابع بازگشتی بالا را پیاده‌سازی کنید.

3. کد ها و کامنت های فایل های بالا را بررسی کنید و بخوانید. دستور کاری که برای قسمت اول باید انجام دهید در فایل ها گفته شده است. همچنین در بخش کامنت های کد ها گفته شده است که چه قسمت های را تکمیل کنید.
4. در آخر توجه داشته باشد که پس از هر تغییر در کدتان ابتدا دستور `make clean` و سپس دستور `make` را بزنید.

پس از اتمام قسمت اول به سراغ قسمت دوم بروید.

قسمت دوم:

در این قسمت از تمرین باید مسئله تبدیل کلمات انگلیسی به زبانی دیگر را حل کنیم. این مسئله به صورت زیر تعریف می شود:

با داشتن یک کلید (کلمه انگلیسی) مقدار آن را (ترجمه آن) برگرداند.

یکی از بهترین راه ها برای حل این مسئله ها استفاده از دیکشنری می باشد. دیکشنری یک ساختمان داده است که یک مقدار را به یک کلید مرتبط می کند (همانند یک دیکشنری در دنیای واقعی).

الزامات:

- لیست کلمات انگلیسی و ترجمه آن در فایل `dataFile.txt` موجود می باشد که می توانید محتوای آن را تغییر دهید یا جفت های دیگری به آن اضافه کنید.
 - با توجه به فایل `Dictionary.h` یک فایل `Dictionary.cpp` بسازید که متد های کلاس در آن پیاده سازی شده است.
 - نکته حائز اهمیت این است که در این کلاس از ساختمان داده درخت جست و جو دودویی (همان کلاس BST) استفاده شده است.
 - در قسمت دوم تمرین از یک `makefile` دیگری استفاده می شود که با نام `makefile (1).txt` در لیست فایل های که در اختیار شما می باشد موجود است.
- قبل از زدن دستور `make` ابتدا یک بک آپ از `makefile` قبلی بگیرید سپس `makefile` دوم را (`makefile (1).txt`) به نام `makefile` تغییر نام بدهید. میتوانید از دستور زیر برای تغییر نام فایل استفاده کنید:
- ```
mv "makefile (1).txt" makefile
```
- سپس برای تست کردن کد این بخش یک فایل به نام `Translator.cpp` که یک `instance` از کلاس `dictionary` می سازد و از آن استفاده کند بنویسید. (راهنمایی: برای نوشتن کد های این فایل مشابه کد های `testDriver.cpp` در بخش اول عمل کنید.)

یک مثال از نحوه عملکرد مورد انتظار از فایل Translator.cpp:

```
uname@hostname: ~$./translate
laser
laser:'uD'a'
today
today:jajvam

apple
Not Found!
<CTRL+D>
```

توضیحات: کلمه انگلیسی به عنوان ورودی گرفته می شود و سپس همان کلمه و ترجمه آن در خروجی چاپ می شود اگر کلمه در دیکشنری وجود نداشت پیام مناسبی چاپ می شود.

## نمره دهی:

1. فایل dataFile.txt را به نحوی تغییر دهید که پس از افزودن لیست کلمات در آن پیچیدگی یافتن کلمات در آن از  $O(\log_2(n))$  باشد. در این صورت این فایل باید حداقل شامل 40 جفت باشد. (امتیازی)
2. صحت عملکرد و تمیزی کد همچنین کامنت گذاری. (الزامی)
3. در این سوال نمی توانید از کتابخانه STL استفاده کنید. (الزامی)

## در مورد گروه ها:

شما می توانید این تمرین را یا به صورت انفرادی و یا در گروه های حداکثر دو نفره تحویل دهید. در صورت تحویل این تکلیف به صورت گروهی نام و شماره دانشجویی هر دو عضو گروه در نام فایل zip آپلود شده به فرمت زیر نوشته شود:

HW2\_STU1NUM\_STU2NUM\_STU1NAME\_STU2NAME.zip