مینی پروژه دوم

لیست دوطرفه پویا (Resizable Deque)

بخش پیاده‌سازی

برای پیاده‌سازی این پروژه تنها مجاز به استفاده از کتابخانه‌های زیر می‌باشید:

Functions and Classes	Header File
همه توابع و کلاس‌ها	#include <iostream>
assert(predicate);	#include <cassert>

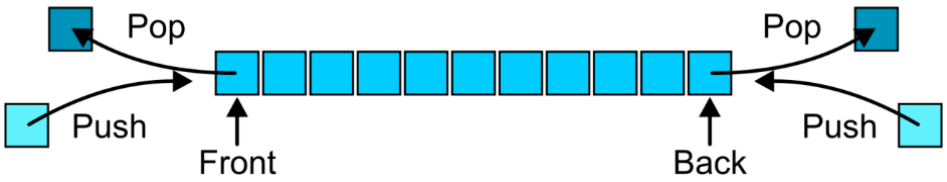
std::min(T, T);
std::max(T, T);
std::swap(T&, T&);
	#include <algorithm>

همه توابع	#include <cmath>
همه ثابت‌ها	#include <cfloat>
همه ثابت‌ها	#include <climits>

برای هر تابع کامنت‌گذاری کرده و موارد زیر را توضیح دهید:

- عملکرد تابع
- کاربرد هر پارامتر
- فرضیات تابع
- خطاهایی که ممکن است رخ دهد و نحوه مدیریت‌شان

نام کلاس شما باید دقیقا ResizableDeque بوده و در فایل resizable_deque.h پیاده‌سازی شود. توابع موردنیاز لیست دوطرفه پویا در ادامه توضیح داده شده‌اند. فایل امضای توابع، ضمیمه تمرین شده است. برخی از توابع پیاده‌سازی شده‌اند و برخی باید توسط شما پیاده‌سازی شوند.



شکل ۱: نمایش لیست دوطرفه پویا (Resizable Deque)

UML Class Diagram

ResizableDeque
- array:int* - iFront:int - iBack:int - dequeSize:int - initialCapacity:int - currCapacity:int
+ create(in capacity:int = 16):ResizableDeque + create(in rd: ResizableDeque): ResizableDeque + size():Integer + capacity():Integer + empty():Boolean + front():Integer + back():Integer + swap(inout list: ResizableDeque) + pushFront(in value: Integer) + pushBack(in value: Integer) + popFront() + popBack() + destroy()

توضیحات فیلدها و توابع

فیلد/تابع	توضیحات
array	آرایه‌ای که عناصر لیست را نگهداری می‌کند
iFront	اندیسی که به ابتدای لیست اشاره می‌کند
iBack	اندیسی که به انتهای لیست اشاره می‌کند
dequeSize	تعداد عناصر موجود در لیست را نگه می‌دارد
initialCapacity	ظرفیت اولیه لیست
currCapacity	ظرفیت فعلی لیست
create(int)	سازنده کلاس که ظرفیت لیست را به عنوان یک پارامتر اختیاری دریافت می‌کند. در صورتی که مقداری در پارامتر قرار داده نشود، مقدار ۱۶ درنظر گرفته می‌شود. اگر ظرفیت تعیین شده کوچکتر از ۱۶ باشد، ظرفیت ۱۶ درنظر گرفته می‌شود.
create(&ResizableDeque)	سازنده کپی کننده. یک لیست دریافت کرده و همه عناصر آن را در خود کپی می‌کند. $O(n)$
size()	تعداد عناصر موجود در لیست را برمی‌گرداند. $O(1)$
empty()	در صورتی که لیست خالی باشد، مقدار true و درغیراینصورت false برمیگرداند. $O(1)$
clear()	همه مقادیر لیست را حذف کرده و ظرفیت را به مقدار اولیه بازمی‌گرداند.
front()	مقداری که در ابتدای لیست قرار دارد را برمی‌گرداند. در صورت خالی بودن لیست خطایی از نوع Underflow پرتاب می‌کند. $O(1)$
back()	مقداری که در انتهای لیست قرار دارد را برمی‌گرداند. در صورت خالی بودن لیست خطایی از نوع Underflow پرتاب می‌کند. $O(1)$

swap(ResizableDeque)	لیست پیوندی ورودی را با خود (this) جابجا می‌کند. $O(1)$
pushFront(Integer)	مقدار جدیدی با مقدار داده شده در ورودی ایجاد کرده و در ابتدای لیست درج می‌کند $O(1)$. در صورتی که فضا برای درج مقدار جدید وجود نداشته باشد، ظرفیت لیست دوبرابر می‌شود $O(n)$.
pushBack(Integer)	مقدار جدیدی با مقدار داده شده در ورودی ایجاد کرده و در انتهای لیست درج می‌کند $O(1)$. در صورتی که فضا برای درج مقدار جدید وجود نداشته باشد، ظرفیت لیست دوبرابر می‌شود $O(n)$.
popFront()	مقدار آغازین لیست را حذف می‌کند. در صورت خالی بودن لیست خطایی از نوع Underflow پرتاب می‌کند $O(1)$. در صورتی که بعد از حذف مقدار، تنها یک چهارم ظرفیت آرایه پر باشد (یا کمتر از یک چهارم) و ظرفیت فعلی آرایه بزرگتر از ظرفیت اولیه باشد، باید ظرفیت آرایه را به نصف کاهش دهید $O(n)$.
popBack()	مقدار پایانی لیست را حذف می‌کند. در صورت خالی بودن لیست خطایی از نوع Underflow پرتاب می‌کند $O(n)$. در صورتی که بعد از حذف مقدار، تنها یک چهارم ظرفیت آرایه پر باشد (یا کمتر از یک چهارم) و ظرفیت فعلی آرایه بزرگتر از ظرفیت اولیه باشد، باید ظرفیت آرایه را به نصف کاهش دهید $O(n)$.

فایل tester.cpp برای آزمون پیاده‌سازی شما نوشته شده است. این کلاس، دستوراتی را به شکل خلاصه شده از ورودی دریافت کرده و توابع متناظر از لیست پیوندی پیاده‌سازی شده توسط شما را فراخوانی می‌کند. صحت پیاده‌سازی شما براساس مقادیری که این فایل در خروجی استاندارد چاپ می‌کند، بررسی خواهد شد. شما نیز می‌توانید با استفاده از همین فایل به تست پیاده‌سازی خود بپردازید.

دستور	توضیحات
new	یک لیست جدید ایجاد می‌کند
copy	یک لیست جدید ایجاد کرده و لیست قبلی را به سازنده آن ارسال می‌کند
end	آخرین لیست ایجاد شده را حذف می‌کند
exit	تست را خاتمه می‌دهد
size	اندازه آخرین لیست ایجاد شده را چاپ می‌کند
cap	ظرفیت آخرین لیست ایجاد شده را چاپ می‌کند
print	مقدار تمامی خانه‌های آخرین لیست ایجاد شده را چاپ می‌کند
empty	خالی بودن/نبودن آخرین لیست ایجاد شده را بررسی می‌کند
swap	تابع swap برای آخرین لیست ایجاد شده و لیست قبل از آن فراخوانی می‌کند
push_front x	تابع pushFront را با پارامتر ورودی x برای آخرین لیست ایجاد شده فراخوانی می‌کند
push_back x	تابع pushBack را با پارامتر ورودی x برای آخرین لیست ایجاد شده فراخوانی می‌کند
pop_front	تابع popFront را برای آخرین لیست ایجاد شده فراخوانی می‌کند
pop_back	تابع popBack را برای آخرین لیست ایجاد شده فراخوانی می‌کند
clear	تابع clear را برای آخرین لیست ایجاد شده فراخوانی می‌کند

یک نمونه ورودی

```
new
 push_front 3
 push_front 2
 push_front 1
 print
 size
 cap
 push_back 4
 push_back 5
 print
end
```

خروجی نمونه بالا

```
START->1->2->3->END
3
16
START->1->2->3->4->5->END
```