Testing Report

Weiqi Wang

301417203

There are two testing files, DQtest and DQarreytest. DQ test is designed to testing the basic functions of the deque. There are two parts of the DQtest. In the first part, the program will store basic data structures into the deque, includes int, float, string and pointers. In the second part, the program will store objects into the deque, includes Deque, Vector, Stack, IVector and TVector. The program will test 63 examples in total.

Testing 1-21 uses Deque<int> to test Deque. Testing 1-3 tests enqueue and clear function. Testing 4-7 tests enqueue, jump, eject and dequeue functions. Testing 8-9 re-enqueue and re-jump an element into the deque, tests the data coverage function of the Deque, also verifying the correctness of front, back and theSize. Testing 10-11 tests reserve function by increase and decrease the capacity of the deque. Testing 12-13 directly dequeue an element from an empty deque, while the program returns unexpected results. Testing 14-15 directly eject an element from an empty deque. Testing 16-21 tests sores multiple int data during a loop.

Testing 22-26 tests storing Double data structure into Deque, and testing 27-29 tests storing String data structure into Deque. Due to the detailly testing on Int data structure, these two parts only contains the basic usage of Double and String data structure. Testing 30-32 tests storing Int Pointers into the Deque. Testing part 1 finishes here.

Testing 33-38 tests the nested usage of Deque, storing a Deque into a Deque. A Deque can’t stores directly into a Deque, so I use the Deque Pointer instead of stores it directly. Testing 39-45 includes the Vector class, tests storing Vectors into Deque. Vectors can be stored directly into the Deque, however, saving a huge number of Vectors into the Deque causes memory error, due to the limitation of the maximum capacity of one block of the heap. The program successfully stores 8 vectors with 8 elements in each Vector, but fail to store 16 vectors with 8 elements in each Vector. So, it is recommended to use pointers to stores an object into the deque, whenever it can be stored directly or not. Testing 46-51 includes TVector, tests storing TVector into the Deque though pointers. Testing 52-57 includes Stack, tests storing Stack into the Deque though pointers. Testing 58-63 includes IVector, tests storing IVector into the Deque though pointers. All of TVector, Stack and IVector can’t be stored directly into the Deque.

DQarreytest uses 3 tests to show the Arrey-style operations of Deque. The program uses d[i] to get the ith element of the Deque d, and uses d[i] = obj to change the ith element to an object or a data structure like int.