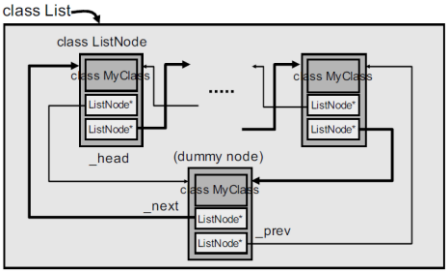
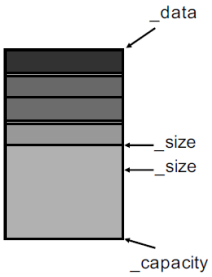
Data Structure and Programming Hw5 Report

B05901020 郭彥廷

一、實作

ADT	DList	Array	Bst (AVL)
Structure	Consist of nodes, which contains two pointers to previous and next nodes' address. 	Stored in consecutive memory _data[]. 	Just like Dlist but the pointers point to a smaller and a larger node. I also keep a integer height in node for auto balance purpose.
Access	Access from head sequentially. Take O(n) time.	Can be randomly accessed by index.	Access from root. Take O(log n) time.
Push back	New a node and insert it between Tail and End by changing the "_next" of Tail and "_prev" of End into the address of new node.	Append _data[size] with data. Need to adjust capacity if too many data added.	Insert from _root recursively. Need or need not rotation.
Pop front	Let _head be _head -> _next and delete original _head.	Let data[0] = data [size-1] and --size	Delete the leftmost node. Need or need not rotation.
Pop back	Connect Tail's _prev and End then delete Tail.	--size	Delete the rightmost node. Need or need not rotation.
Erase	Connect the node before and after the node deleted.	Replace the deleted data by _data[size-1] --size	Find the node recursively in O(log n) time.
Sort	Insertion sort	::sort()	X

二、Performance comparison

1. Add

(1) Design:

Add a million data to ADTs, compare their runtime and space usage.

(2) Expectation:

Runtime: Dlist: O(1) < Array: O(1) < Bst: O(log n)

(dynamic array need to change size from time to time)

Space: Array < Dlist < Bst

(Bst have an additional integer data member to Dlist)

(3) Outcome: (Averaged of 3 times)

ADTs	Dlist	Array	Bst
Runtime (s)	0.17	0.31	2.98
Space (MB)	62.12	49.14	62.21

2. Delete

(1) Design:

Delete a thousand random data in 100000 data.

(2) Expectation:

Runtime: Array: O(1) < Dlist: O(1) < Bst: O(log n)

(3) Outcome: (Averaged of 3 times)

ADTs	Dlist	Array	Bst
Runtime (s)	0.63	~0	6.53

3. Sort

(1) Design:

Sort 50000 data.

(2) Expectation:

Runtime: Bst: 0 < Array: O(nlog n) < Dlist: O(n^2)

(3) Outcome: (Averaged of 3 times)

ADTs	Dlist	Array	Bst
Runtime (s)	39.17	0.02	0

4. Search
   (1) Design:
   Delete a non-existent object (poipoi) in 5000000 data
   (2) Expectation:
   Runtime: Bst: O(log n) < Array: O(n) ~ Dlist: O(n)
   (3) Outcome: (Averaged of 3 times)

ADTs	Dlist	Array	Bst
Runtime (s)	0.07	0.05	0