Lab Report

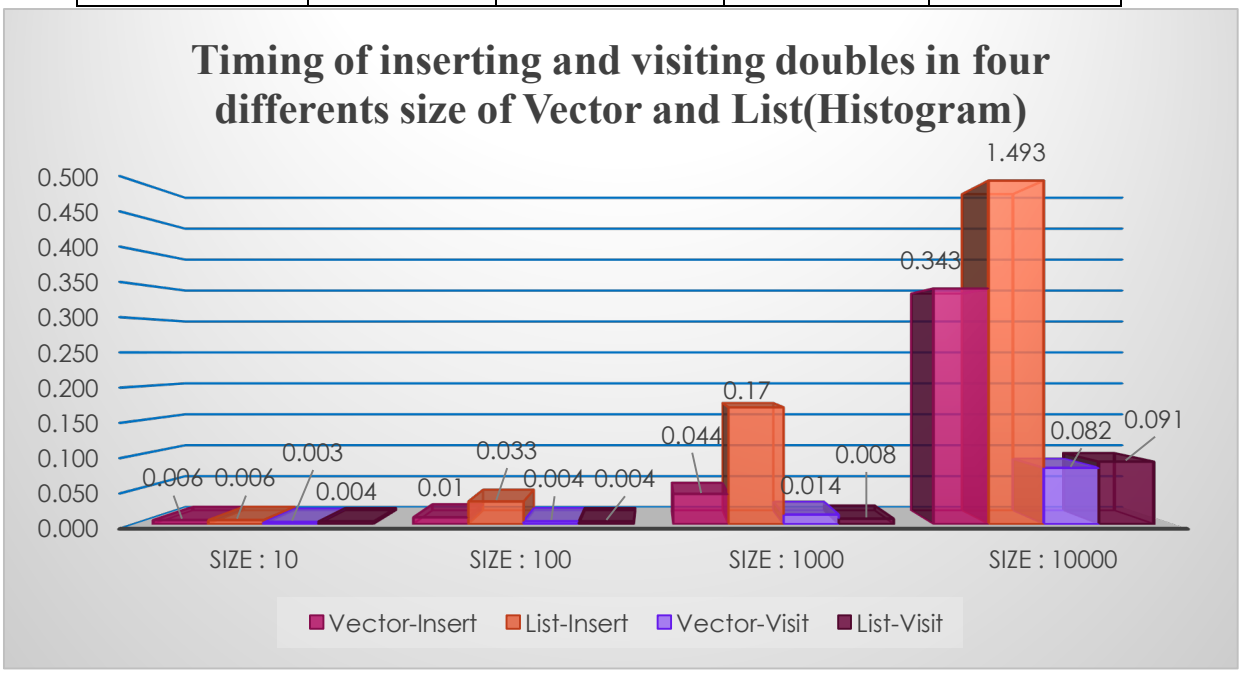
CMPT 225

Junchen Li

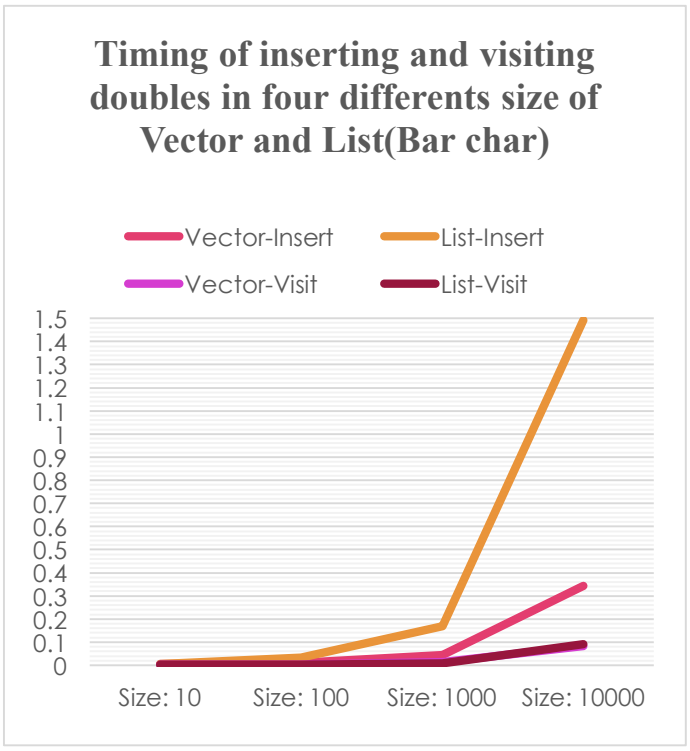
2020.1.25

First part is talking about inserting and visiting the vector and list between different size (10,100,1000,10000) with **DOUBLE and using push_back().**

Number of size	The result of timing in Vector (Unit : milliseconds)		The result of timing in List (Unit : milliseconds)	
	Inserting	Visiting	Inserting	Visiting
10	0.006 ms	0.003 ms	0.006 ms	0.004 ms
100	0.01 ms	0.004 ms	0.033 ms	0.004 ms
1000	0.044 ms	0.014 ms	0.017 ms	0.008 ms
10000	0.343 ms	0.082 ms	1.493 ms	0.091 ms



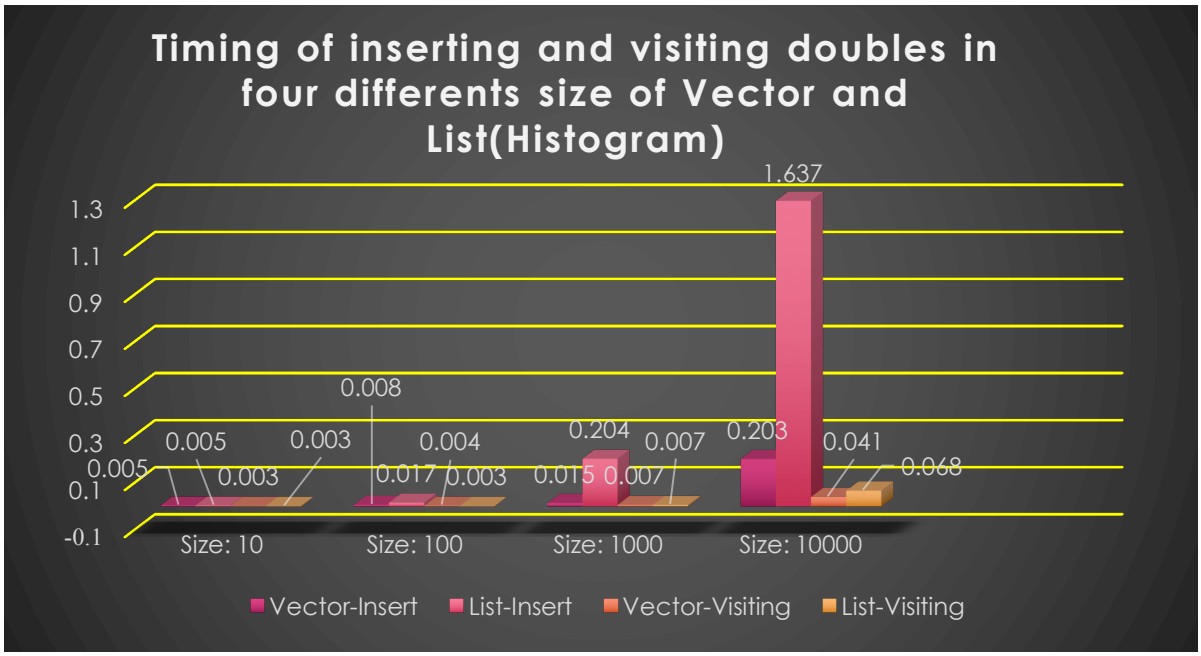
Unit: millisecond



From the reading data and graphs, we can notice that the larger size of containers the more time it needs to run. The most tedious operation is inserting elements to a double data type of list. Generally, either vector or list, the insert elements need more time than visit element. Also from the bar char, for the visiting elements in vector and list. Although it has the different element size, they look need almost same time to run in each interval of size.

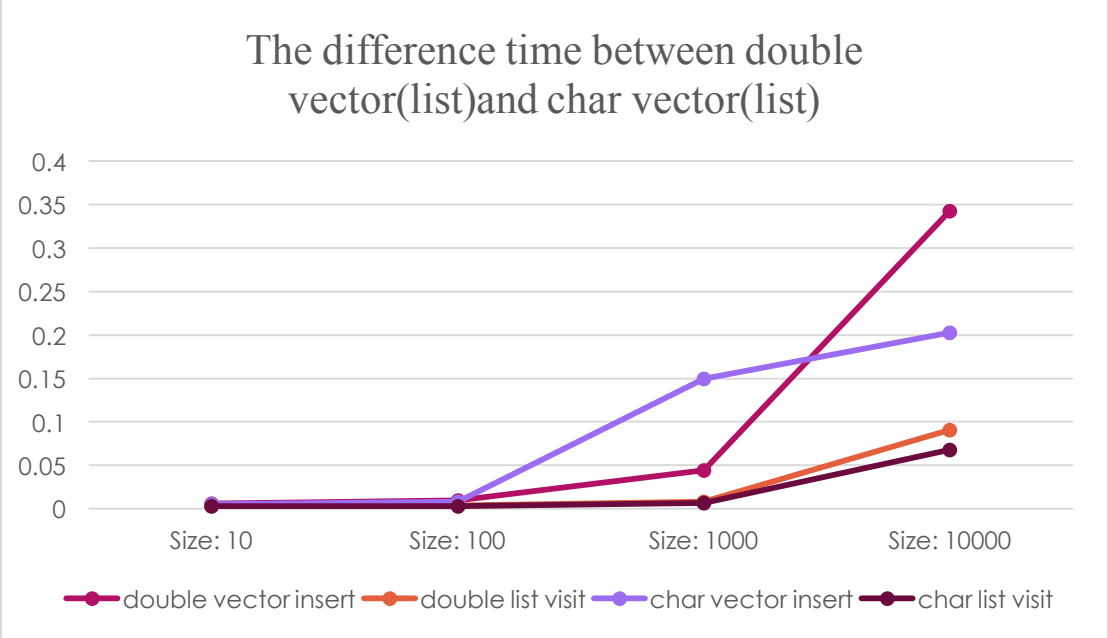
Then talking about inserting and visiting the vector and list between different size (10,100,1000,10000) with **CHAR and using the push_back().**

Number of size	The result of timing in Vector (Unit : milliseconds)		The result of timing in List (Unit : milliseconds)	
	Insert	Visit	Insert	Visit
10	0.005 ms	0.003 ms	0.005 ms	0.003 ms
100	0.008 ms	0.004 ms	0.017 ms	0.003 ms
1000	0.015 ms	0.007 ms	0.204 ms	0.007 ms
10000	0.203 ms	0.041 ms	1.637 ms	0.068 ms



Unit: nillisecond

Now, let us to compare data between the double and char and I select four groups of data.



As we can see that the double data type will need more time to perform operations than char data type. The funny thing is for inserting, before the size of 4000, the char will be quickly than double, however, after around 4000 the double will exceed the char one.

Before all the experiment, we just use the "push_back" operation to insert the element. Now let's try to figure what happen if we use the "push_front".

Number of size	The result of timing in Vector (Unit : milliseconds)		The result of timing in List (Unit : milliseconds)	
Data of the **DOUBLE** type				
	Insert	Visit	Insert	Visit
10	0.001 ms	0.003 ms	0.006 ms	0.004 ms
100	0.168 ms	0.004 ms	0.04 ms	0.005 ms
1000	12.065 ms	0.017 ms	0.194 ms	0.009 ms
10000	795.903 ms	0.033 ms	1.411 ms	0.069 ms
Data of the **CHAR** type				
	Insert	Visit	Insert	Visit
10	0.009 ms	0.004 ms	0.009 ms	0.003 ms
100	0.162 ms	0.004 ms	0.019 ms	0.014 ms
1000	9.382 ms	0.006 ms	0.211 ms	0.01 ms
10000	785.676 ms	0.034 ms	1.432 ms	0.06 ms