ASSIGNMENT 1

Mengqi Wang, 301405066

I have organized the returned data into the 2 tables and 4 plots (see below).

By observation, it can be clearly found that:

1. As the size grows, the running time grows;

2. For the same element type and the same size, the running time for float is similar with that of integer;

3. Time for vector visiting and list visiting is similar ,and is minimal compared to the running time of insertion;

4. List insertion by push_back is very time-consuming, is much more time-consuming than the list's push_front function, and is much more time-consuming than vector's push_back function;

5. Vector insertion by push_front is very time-consuming, is much more time-consuming than the vector's push_back function, and is much more time-consuming than the list's push_front function, and even more time consuming than

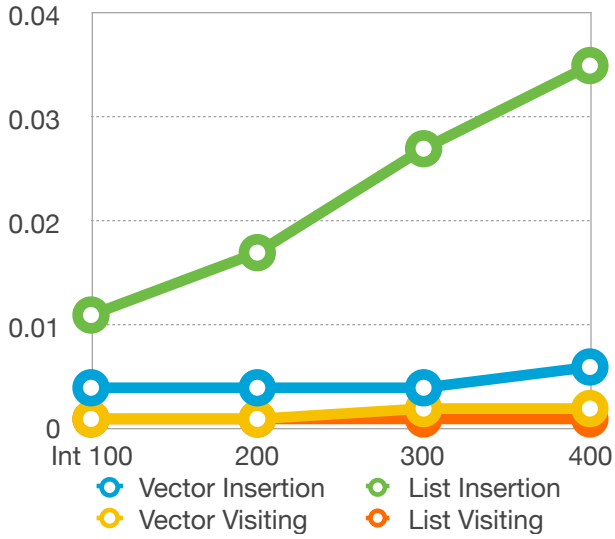
Table 1: Data of Part1

Type of Elements	Int				Float			
Size / Time (milliseconds)	100	200	300	400	100	200	300	400
Vector Insertion	0.004	0.004	0.004	0.006	0.002	0.003	0.004	0.007
List Insertion	0.011	0.017	0.027	0.035	0.011	0.019	0.03	0.038
Vector Visiting	0.001	0.001	0.002	0.002	0.001	0.001	0.001	0.003
List Visiting	0.001	0.001	0.001	0.001	0.001	0.001	0.002	0.005

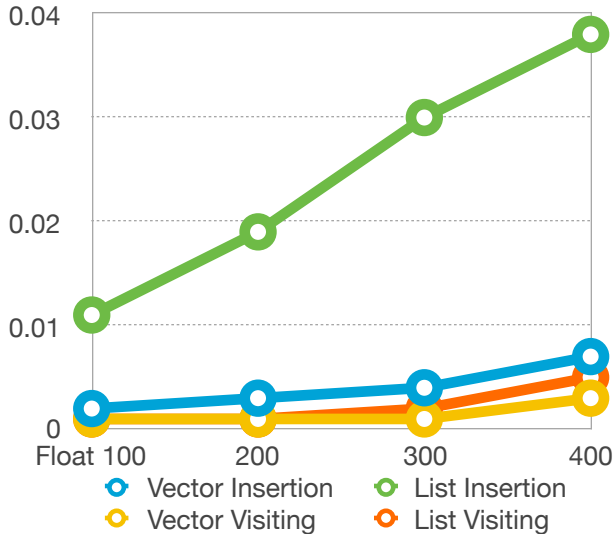
Table 2: Data of Part2-1

Type of Elements	Int				Float			
Size / Time (milliseconds)	100	200	300	400	100	200	300	400
Vector Insertion	0.032	0.215	0.588	1.349	0.079	0.182	0.568	1.153
List Insertion	0.024	0.074	0.058	0.072	0.02	0.044	0.073	0.081
Vector Visiting	0.002	0.003	0.003	0.004	0.001	0.002	0.004	0.005
List Visiting	0.001	0.004	0.005	0.009	0.001	0.003	0.005	0.008

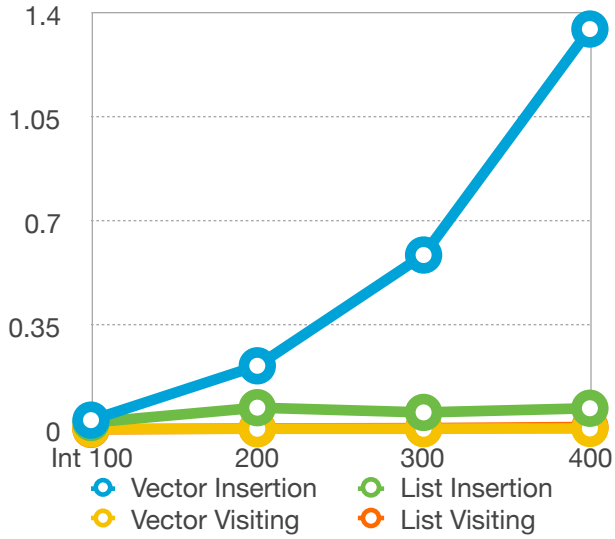
Plot1: part1-running time for int



Plot2: part1-running time for float



Plot3: part2-running time for int



Plot4: part2-running time for float