Data Structure HW7 Report

Name: **洪理川**

Student ID: **B113040056**

Environment:

- CPU Speed : Base Speed 3.00 GHz, 6 Core, 12 Thread, Max. Boost ClockUp to 4.0GHz
- Memory : 24.0 GB, SODIMM 3200MHz
- Operating System : Windows 10, Version 22H2
- Compiler : GCC (GNU Compiler Collection) (Rev10, Built by MSYS2 project) 12.2.0

Insertion Sort

(times measured in seconds)

Data Size	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**	**比例**
100	0.000011	0.000014	0.000015	0.000014	0.000011	0.000011	0.000012	0.000012	0.000015	0.000013	0.0000128	17.25
500	0.000208	0.000225	0.000212	0.00022	0.000208	0.00021	0.000273	0.000239	0.000204	0.000209	0.0002208	3.8233696
1,000	0.000762	0.001212	0.000773	0.000764	0.000758	0.000757	0.000771	0.001088	0.000776	0.000781	0.0008442	21.739161
5,000	0.017921	0.017941	0.018404	0.018024	0.017985	0.018321	0.021045	0.017947	0.018008	0.017926	0.0183522	3.9426935
10,000	0.073084	0.07095	0.072208	0.071602	0.071063	0.072931	0.073906	0.071147	0.071743	0.074937	0.0723571	25.088062
50,000	1.810823	1.841141	1.860447	1.831847	1.813122	1.768417	1.892514	1.797681	1.766055	1.770947	1.8152994	3.953445
100,000	7.340925	7.096091	7.127624	7.120893	7.14197	7.206478	7.152979	7.269409	7.136992	7.173503	7.1766864	27.001828
500,000	203.80031	200.19266	194.79836	191.48429	192.51721	192.62732	191.84613	188.69487	190.51499	191.3604	193.78365	
1,000,000	786.129252 (TLE)	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	#DIV/0!	
5,000,000	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	#DIV/0!	

Merge Sort

(times measured in seconds)

Data Size	1	2	3	4	5	6	7	8	9	10	Avg.	比例
100	0.000038	0.000047	0.000042	0.000041	0.000056	0.000057	0.000024	0.000037	0.000049	0.00004	**0.0000431**	0.00004361
500	0.000159	0.000181	0.000171	0.000138	0.000163	0.000175	0.000178	0.000185	0.000168	0.000221	**0.0001739**	1.884991374
1,000	0.000281	0.000311	0.000262	0.000407	0.000308	0.00032	0.000383	0.000262	0.000348	0.000396	**0.0003278**	3.474679683
5,000	0.001276	0.001235	0.001154	0.001114	0.001122	0.001067	0.001074	0.001137	0.001117	0.001094	**0.001139**	2.048902546
10,000	0.002467	0.002383	0.002343	0.002378	0.002379	0.002245	0.002345	0.002209	0.002336	0.002252	**0.0023337**	5.721386639
50,000	0.012922	0.013079	0.014637	0.014487	0.012357	0.014334	0.013011	0.013456	0.012177	0.01306	**0.013352**	1.929239065
100,000	0.026729	0.025808	0.025926	0.025821	0.026427	0.025369	0.025465	0.025289	0.025123	0.025635	**0.0257592**	5.653312215
500,000	0.140942	0.153884	0.138153	0.139711	0.151455	0.148266	0.141065	0.142667	0.146692	0.153413	**0.1456248**	2.041803319
1,000,000	0.292096	0.314748	0.287696	0.297352	0.293913	0.303903	0.299413	0.29467	0.293862	0.295719	**0.2973372**	5.625959685
5,000,000	1.680714	1.684642	1.676799	1.627135	1.665881	1.790005	1.753453	1.592149	1.593247	1.664046	**1.6728071**	

Radix Sort

(times measured in seconds)

Data Size	1	2	3	4	5	6	7	8	9	10	Avg.	比例
100	0.000003	0.000003	0.000003	0.000003	0.000003	0.000003	0.000003	0.000003	0.000004	0.000003	0.0000031	11.06451613
500	0.000025	0.000051	0.000033	0.000036	0.000045	0.000051	0.000024	0.000024	0.000029	0.000025	0.0000343	1.778425656
1,000	0.000064	0.000059	0.000059	0.000065	0.000059	0.000059	0.000059	0.000063	0.000063	0.00006	0.000061	5.708196721
5,000	0.000396	0.000573	0.000312	0.000318	0.00031	0.000309	0.000317	0.000313	0.000325	0.000309	0.0003482	2.302412407
10,000	0.000785	0.000876	0.000769	0.000783	0.000771	0.000796	0.000915	0.000776	0.000775	0.000771	0.0008017	5.059249096
50,000	0.004107	0.00402	0.004044	0.004049	0.004023	0.004049	0.004028	0.0041	0.004059	0.004081	0.004056	2.375986193
100,000	0.00978	0.009753	0.009596	0.009558	0.009544	0.009654	0.00975	0.009578	0.009595	0.009562	0.009637	5.330486666

500,000	0.051593	0.051582	0.051191	0.052343	0.050275	0.05227	0.051415	0.051984	0.050774	0.050272	0.0513699	2.161374657
1,000,000	0.10895	0.111359	0.108475	0.109894	0.108647	0.110009	0.109349	0.12387	0.107604	0.112139	0.1110296	5.888619791
5,000,000	0.640592	0.640821	0.643266	0.643198	0.653758	0.65357	0.649138	0.717994	0.64463	0.651144	0.6538111	

C qsort()

(times measured in seconds)

Data Size	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**	比例
100	0.000014	0.000004	0.000007	0.000006	0.000007	0.000005	0.000004	0.000005	0.000006	0.000004	**0.0000062**	6.1612903
500	0.000046	0.000036	0.000057	0.000035	0.000034	0.000029	0.000034	0.000038	0.000033	0.00004	**0.0000382**	1.9162304
1,000	0.000088	0.000066	0.000077	0.000072	0.000075	0.000064	0.000071	0.000068	0.000073	0.000078	**0.0000732**	5.3497268
5,000	0.000392	0.00044	0.000379	0.000376	0.000396	0.000393	0.000374	0.000373	0.000365	0.000428	**0.0003916**	2.1149132
10,000	0.000795	0.000862	0.000815	0.000987	0.000777	0.000786	0.00083	0.000789	0.00085	0.000791	**0.0008282**	5.9226032
50,000	0.00466	0.00466	0.004666	0.005143	0.004706	0.004646	0.004673	0.004995	0.006216	0.004686	**0.0049051**	2.0260137
100,000	0.009923	0.010043	0.010121	0.009847	0.01003	0.009763	0.010002	0.009891	0.009802	0.009956	**0.0099378**	5.7384632
500,000	0.055544	0.055156	0.055772	0.055722	0.064277	0.056613	0.056842	0.055726	0.055364	0.059261	**0.0570277**	2.046893
1,000,000	0.116375	0.117849	0.117239	0.115771	0.116674	0.118314	0.116098	0.117128	0.116159	0.115689	**0.1167296**	5.7251922
5,000,000	0.66471	0.66101	0.656378	0.668747	0.661812	0.64349	0.672253	0.697968	0.684761	0.671865	**0.6682994**	

C++ Sort()

(times measured in seconds)

Data Size	1	2	3	4	5	6	7	8	9	10	Avg.	比例
100	0.000007	0.000006	0.000006	0.000013	0.000005	0.000007	0.000006	0.000007	0.000006	0.000007	**0.000007**	6.2857143
500	0.000041	0.000033	0.000079	0.000041	0.000051	0.000038	0.00004	0.000041	0.000038	0.000038	**0.000044**	1.7477273
1,000	0.00007	0.000086	0.000073	0.000082	0.000074	0.00007	0.000083	0.000082	0.000074	0.000075	**0.0000769**	5.7022107
5,000	0.000438	0.000442	0.000434	0.000433	0.000431	0.000429	0.000434	0.000435	0.000462	0.000447	**0.0004385**	2.3384265
10,000	0.000942	0.000946	0.000955	0.000939	0.001127	0.000948	0.000954	0.000977	0.000944	0.001522	**0.0010254**	5.8439633
50,000	0.006497	0.005421	0.005307	0.005758	0.005771	0.007636	0.005315	0.007177	0.005284	0.005758	**0.0059924**	1.9245878
100,000	0.012769	0.011065	0.011103	0.011461	0.011604	0.011117	0.011592	0.011243	0.011599	0.011776	**0.0115329**	5.8578502
500,000	0.064128	0.064076	0.065007	0.065763	0.07713	0.06347	0.063922	0.064978	0.070945	0.076161	**0.067558**	1.9769709
1,000,000	0.133298	0.134097	0.133821	0.133589	0.132818	0.133799	0.132521	0.133632	0.134884	0.133143	**0.1335602**	5.7078306
5,000,000	0.756803	0.802682	0.752354	0.754644	0.761051	0.782518	0.740307	0.782941	0.753709	0.736381	**0.762339**	

From the above data, it is evident that Selection Sort is the least efficient sorting algorithm. As the data size doubles, the time taken to complete the sorting process increases fourfold. Moreover, when the data size increases by a factor of five, the time taken rises to over 20 times the original duration. Among the algorithms analysed, the radix sort function emerges as the best performer, taking an average of 0.65 seconds to sort five million numbers, followed by qsort() by only left behind with 0.01 seconds. Interestingly, all other sorting algorithms, except for Selection Sort, demonstrate efficiency close to linearity. This implies that as the data size doubles, the time taken also approximately doubles.

An intriguing observation I made is the significant impact of CPU clock speed on sorting efficiency. During an experiment where I ran the sorting algorithms on my laptop with an underclocked CPU, the time taken for completion was consistently twice as long as under normal conditions.