Exectuion timings

Name: Daniyal hassan

Reg No: BCS231008

**BubbleSort.cpp:**

#include <iostream>

#include <vector>

#include <algorithm>

#include <chrono>

using namespace std;

using namespace chrono;

void bubbleSort(vector<int>& arr) {

int n = arr.size();

for (int i = 0; i < n - 1; ++i) {

for (int j = 0; j < n - i - 1; ++j) {

if (arr[j] > arr[j + 1]) {

swap(arr[j], arr[j + 1]);

}

}

}

}

int main() {

const int vectorSize = 100000;

vector<int> data(vectorSize);

for (int i = 0; i < vectorSize; ++i) {

data[i] = vectorSize - i;

}

auto bubbleStart = high\_resolution\_clock::now();

bubbleSort(data);

auto bubbleEnd = high\_resolution\_clock::now();

auto bubbleDuration = duration\_cast<milliseconds>(bubbleEnd - bubbleStart);

cout << "Bubble Sort Execution Time: " << bubbleDuration.count() << " milliseconds" << endl;

// Displaying only the first 10 and last 10 elements for brevity

cout << "First 10 elements after Bubble Sort: ";

for (int i = 0; i < min(10, vectorSize); ++i) {

cout << data[i] << " ";

}

cout << endl;

cout << "Last 10 elements after Bubble Sort: ";

for (int i = max(0, vectorSize - 10); i < vectorSize; ++i) {

cout << data[i] << " ";

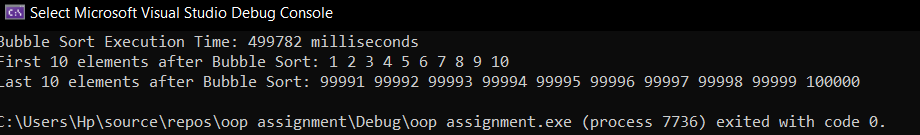
}

cout << endl;

return 0;

}

Output:



**STLSort.cpp:**

// STLSort.cpp

#include <iostream>

#include <vector>

#include <algorithm>

#include <chrono>

using namespace std;

using namespace chrono;

int main() {

const int vectorSize = 100000;

vector<int> data(vectorSize);

for (int i = 0; i < vectorSize; ++i) {

data[i] = vectorSize - i;

}

auto stlStart = high\_resolution\_clock::now();

sort(data.begin(), data.end());

auto stlEnd = high\_resolution\_clock::now();

auto stlDuration = duration\_cast<milliseconds>(stlEnd - stlStart);

cout << "STL Sort Execution Time: " << stlDuration.count() << " milliseconds" << endl;

// Displaying only the first 10 and last 10 elements for brevity

cout << "First 10 elements after STL Sort: ";

for (int i = 0; i < min(10, vectorSize); ++i) {

cout << data[i] << " ";

}

cout << endl;

cout << "Last 10 elements after STL Sort: ";

for (int i = max(0, vectorSize - 10); i < vectorSize; ++i) {

cout << data[i] << " ";

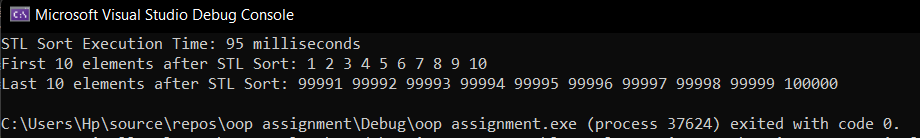
}

cout << endl;

return 0;

}

Output:



System Requirements:

C:\Users\Hp>systeminfo

Host Name: DANISLAPTOP

OS Name: Microsoft Windows 11 Pro

OS Version: 10.0.22000 N/A Build 22000

OS Manufacturer: Microsoft Corporation

OS Configuration: Standalone Workstation

OS Build Type: Multiprocessor Free

Registered Owner: Hp

Registered Organization:

Product ID: 00331-10000-00001-AA339

Original Install Date: 2/14/2023, 10:09:04 PM

System Boot Time: 12/15/2023, 1:39:37 PM

System Manufacturer: HP

System Model: HP 255 15.6 inch G9 Notebook PC

System Type: x64-based PC

Processor(s): 1 Processor(s) Installed.

[01]: AMD64 Family 25 Model 80 Stepping 0 AuthenticAMD ~2301 Mhz

BIOS Version: Insyde F.10, 10/20/2022

Windows Directory: C:\Windows

System Directory: C:\Windows\system32

Boot Device: \Device\HarddiskVolume1

System Locale: en-us;English (United States)

Input Locale: en-us;English (United States)

Time Zone: (UTC+05:00) Islamabad, Karachi

Total Physical Memory: 7,488 MB

Available Physical Memory: 1,224 MB

Virtual Memory: Max Size: 23,360 MB

Virtual Memory: Available: 4,463 MB

Virtual Memory: In Use: 18,897 MB

Page File Location(s): D:\pagefile.sys

Domain: WORKGROUP

Logon Server: \\DANISLAPTOP

Hotfix(s): 6 Hotfix(s) Installed.

[01]: KB5030650

[02]: KB5028954

[03]: KB5011048

[04]: KB5012170

[05]: KB5031358

[06]: KB5031591

Network Card(s): 3 NIC(s) Installed.

[01]: Realtek PCIe GbE Family Controller

Connection Name: Ethernet

Status: Media disconnected

[02]: Realtek RTL8822CE 802.11ac PCIe Adapter

Connection Name: Wi-Fi

DHCP Enabled: Yes

DHCP Server: 192.168.0.1

IP address(es):

(I have removed my IP address since my repository is public and sir you might hack my pc hehe I don’t know)

[03]: Bluetooth Device (Personal Area Network)

Connection Name: Bluetooth Network Connection

Status: Media disconnected

Hyper-V Requirements: VM Monitor Mode Extensions: Yes

Virtualization Enabled In Firmware: Yes

Second Level Address Translation: Yes

Data Execution Prevention Available: Yes