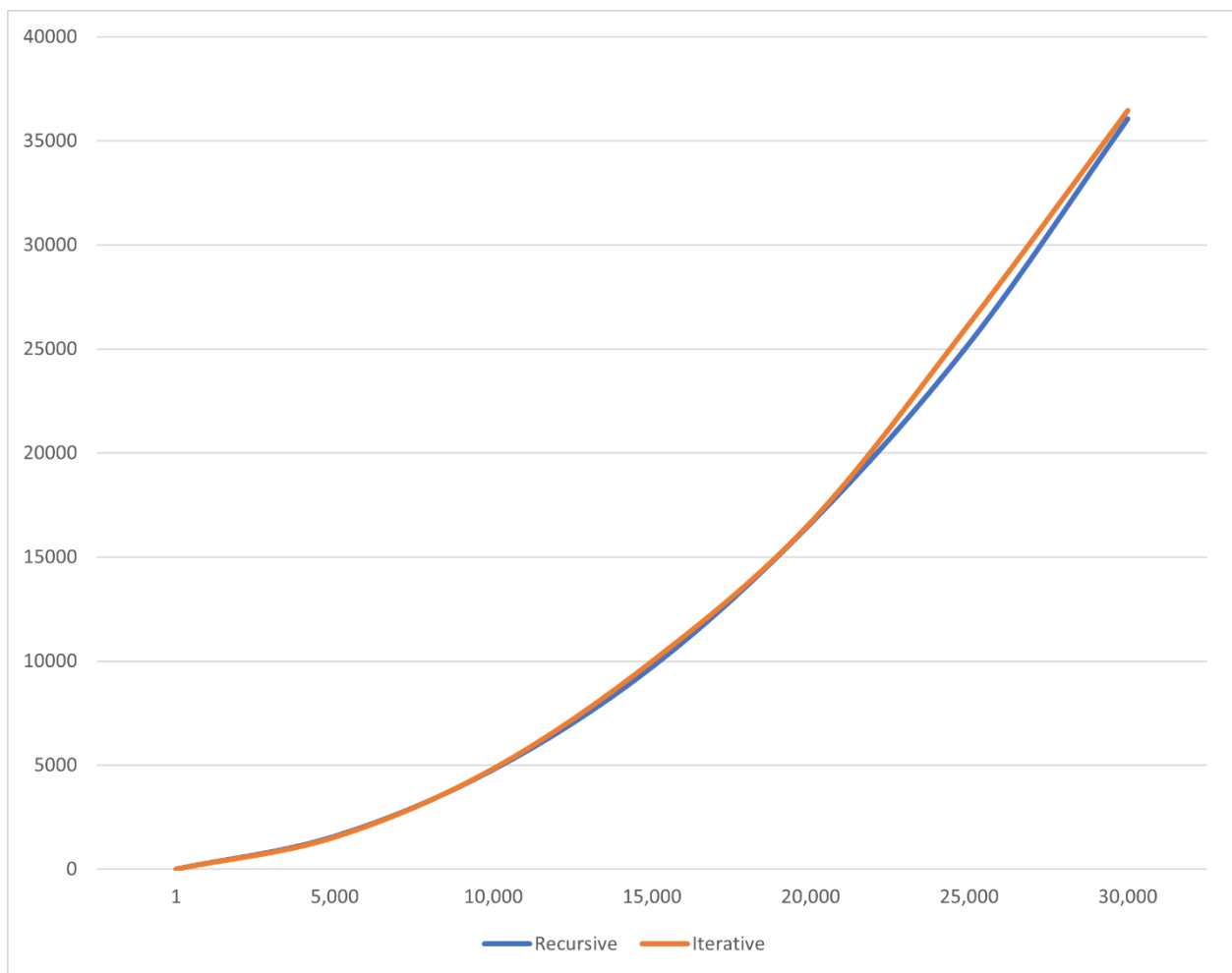


Index Number – 210098R

Name – De Silva APC

Chart –



Terminal outputs –

Length	Time taken in micro seconds	
	Recursive	Iterative
1	0	0
5,000	1572	1533
10,000	4780	4826
15,000	9704	9979
20,000	16598	16663
25,000	25259	26212
30,000	36055	36471

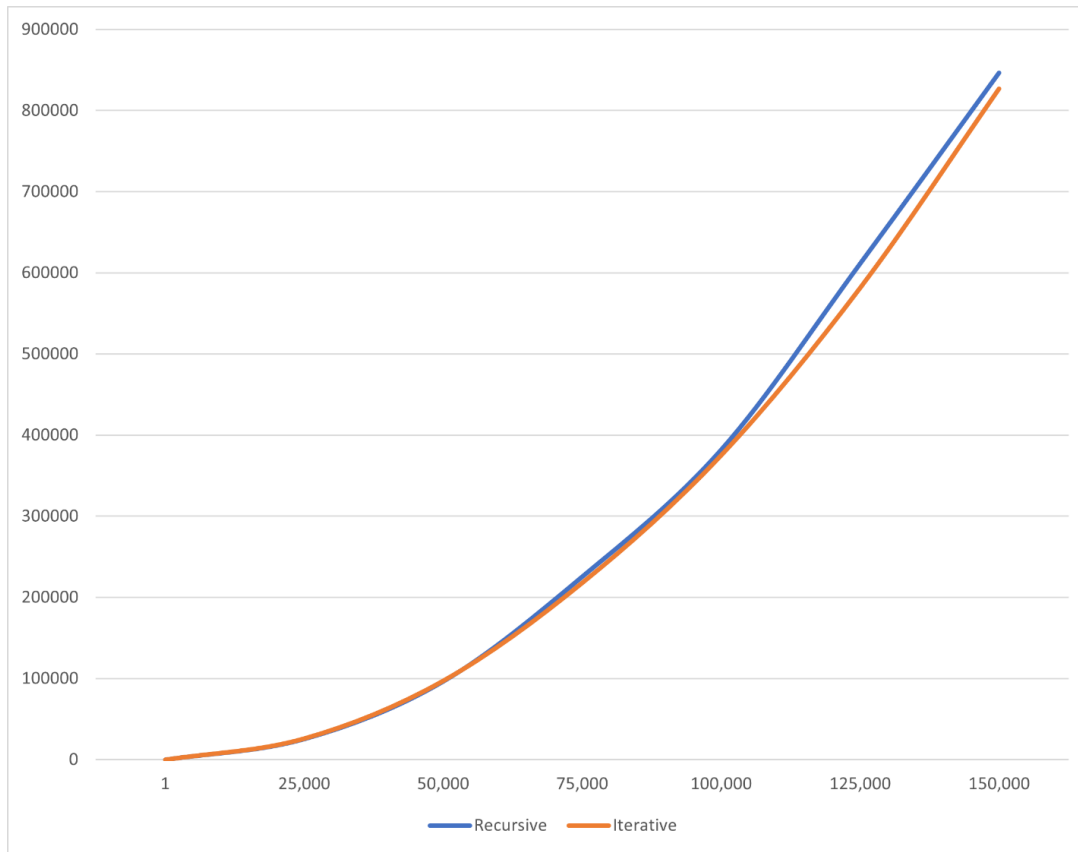
Discussion –

By repeated execution the time varies for each length but the above plot is done by the random values it produced for the first execution for each array length without retrying and taking a biased result. So it turns out that almost all the time the recursive algorithm is faster than the iterative implementation when it comes to quick sort algorithm. But its noticeable that the margin is not a huge one anyway.

Probably this is happening because of the fact that recursive algorithm has the advantage of the call stack to reduce the overhead of maintaining an explicit stack. But with much larger arrays, iterative option should be faster. So I increased the scale and redid the process and it confirmed that is what exactly happens here.

Chart and Outputs after rescaling –

Length	Time taken in micro seconds	
	Recursive	Iterative
1	0	0
25,000	25562	26101
50,000	96526	97261
75,000	225482	217861
100,000	381802	375022
125,000	611157	581145
150,000	846274	826608



Q2 – Console output when the input is 10.

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help HelloWorld - main.cpp
HelloWorld main.cpp
main.cpp 19 cout << "The random array that has been generated is:- " << endl;
20 cout << " | ";
Run HelloWorld
C:\Users\accha\CLionProjects\HelloWorld\cmake-build-debug\HelloWorld.exe
Enter the size of the array:10
The random array that has been generated is:-
| 62 | 52 | 77 | 51 | 26 | 39 | 17 | 22 | 31 | 21 |
[62] 62.0
[52, 62] 57.0
[52, 62, 77] 62.0
[51, 52, 62, 77] 57.0
[26, 51, 52, 62, 77] 52.0
[26, 39, 51, 52, 62, 77] 51.5
[17, 26, 39, 51, 52, 62, 77] 51.0
[17, 22, 26, 39, 51, 52, 62, 77] 45.0
[17, 22, 26, 31, 39, 51, 52, 62, 77] 39.0
[17, 21, 22, 26, 31, 39, 51, 52, 62, 77] 35.0
Process finished with exit code 0
```