Assignment 2 - Data structures and algorithms

Hassan Mualla gusmuaha@student.gu.se
Fredrik Ullman gusullmfr@student.gu.se
Group # 33

Question 1.

The complexity is O(N)

Question 2.

T(N) = AT($\frac{N}{B}$) + O (N^k)
where A > 1 & B > 1

1.
T(N) = 4T($\frac{N}{2}$) + O (N^2)
4 = 2² , therefore

T(N) = O (N^k log N)
→ O(N² * log N)
= O(N²logN)

2.
T(N) = 5T($\frac{N}{2}$) + O (N^2)
5>2² , therefore

T(N) = O($N^{log_B A}$)

→ T(N) = O($N^{log_2 5}$)
O($N^{2,3219}$)

Question 6 .

Benchmark Summary		Execution time(CPU seconds)	
		Sorted (Ascending)	Sorted (Descending)
bubble sort	Array size : 100	2.96331* 10^{-4}	3.05522* 10^{-4}
	Array size : 100000	15.718445955	15.591719718 (Actual 15 sec!!)
merge sort	Array size : 100	7.9516* 10^{-5}	9.035* 10^{-5}
	Array size : 100000	6.1817372* 10^{-2}	6.552729* 10^{-2}
insertion sort	Array size : 100	6.9461* 10^{-5}	7.0073* 10^{-5}
	Array size : 100000	9.03166493* 10^{-1}	9.21904788* 10^{-1}
quick sort pivot : median	Array size : 100	5.0849* 10^{-5}	3.9291* 10^{-5}
	Array size : 100000	6.5199845 * 10^{-2}	5.2413082* 10^{-2}
quick sort pivot [0]	Array size : 100	3.7616* 10^{-5}	5.2179* 10^{-5}
	Array size : 100000	5.0933619 * 10^{-2}	4.332952* 10^{-2}

Further questions :

1. Bubble sort was the worst for all of the cases.

2. For small number of elements (100), the quicksort with pivot at index 0 performed slightly better since the quicksort with median had to take an extra step to find median for each recursion.

   While when we tested with 100000 elements the performance of the quicksort with median improved compared to index at 0. In conclusion, the quicksort with median should perform better in theory when the number of elements is large enough.

3. In theory, the answer should be the same as question 2. However, since we are not sampling and the array used for ascending is different than in descending, there is potential bias in the answer as the array order before sorting might be convenient for some algorithms.

4. It performs equally as fast as the other sortings algorithms for small elements but it is not optimal for large arrays even though its not as slow as bubble sort.

5. It performs almost the same for ascending and descending cases. And the performance is almost matched with quicksort for both 100 and 100000 element.