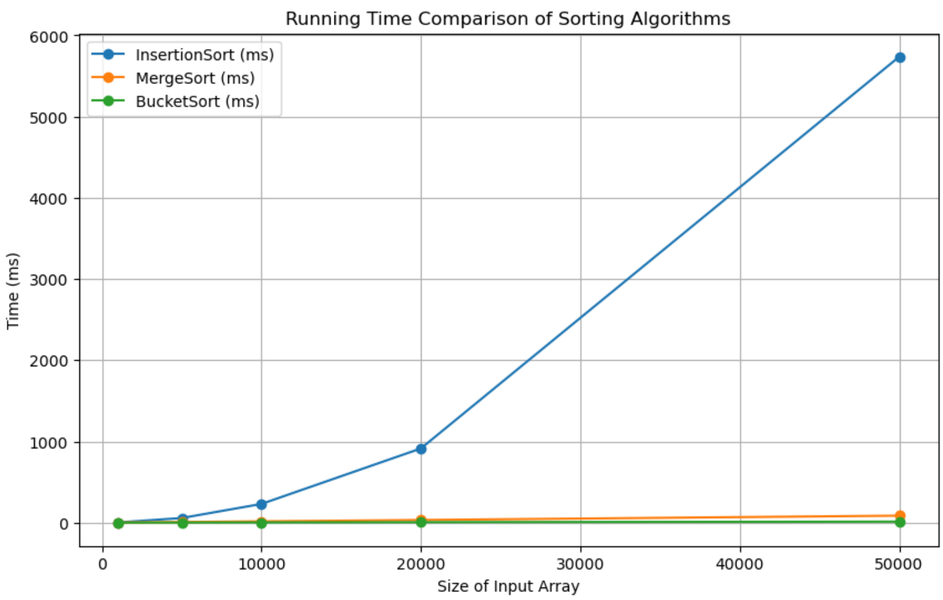
Problem 1

$f(n)$	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\log n$	$2^{1.00\times10^6}$	$2^{6.00\times10^7}$	$2^{3.60\times10^9}$	$2^{8.64\times10^{10}}$	$2^{2.59\times10^{12}}$	$2^{3.15\times10^{13}}$	$2^{3.15\times10^{15}}$
$\sqrt{n}$	1.00×10^{12}	3.60×10^{15}	1.29×10^{19}	7.46×10^{21}	6.71×10^{24}	9.94×10^{26}	9.94×10^{30}
n	1.00×10^{6}	6.00×10^{7}	3.60×10^{9}	8.64×10^{10}	2.59×10^{12}	3.15×10^{13}	3.15×10^{15}
$n\log n$	6.27×10^{4}	2.80×10^{6}	1.33×10^{8}	2.75×10^{9}	7.18×10^{10}	7.97×10^{11}	6.86×10^{13}
n^2	1.00×10^{3}	7.74×10^{3}	6.00×10^{4}	2.93×10^{5}	1.60×10^{6}	5.61×10^{6}	5.61×10^{7}
n^3	1.00×10^{2}	3.91×10^{2}	1.53×10^{3}	4.42×10^{3}	1.37×10^{4}	3.15×10^{4}	1.46×10^{5}
2^n	19	25	31	36	41	44	51
$n!$	9	11	12	13	15	16	17

Problem 2

Size	InsertionSort (ms)	MergeSort (ms)	BucketSort (ms)
1000	2.21681	1.41739	0.386627
5000	56.805	7.5942	1.53927
10000	230.205	15.8429	2.74471
20000	913.366	32.2628	5.57126
50000	5736.25	85.8542	12.306



Problem 3

$A = 3,\ 41,\ 52,\ 26,\ 38,\ 57,\ 9,\ 49$
$split - A = 3,\ 41,\ 52,\ 26 \quad B = 38,\ 57,\ 9,\ 49$
$split - A = 3,\ 41 \quad B = 52,\ 26 \quad C = 38,\ 57 \quad D = 9,\ 49$
$split - A = 3 \quad B = 41 \quad C = 52 \quad D = 26 \quad E = 38 \quad F = 57 \quad G = 9 \quad H = 49$
$merge - A = 3,\ 41 \quad B = 26,\ 52 \quad C = 38,\ 57 \quad D = 9,\ 49$
$merge - A = 3,\ 26,\ 41,\ 52 \quad B = 9,\ 38,\ 49,\ 57$
$merge - A = 3,\ 9,\ 26,\ 38,\ 41,\ 49,\ 52,\ 57$

Problem 4

We can first sort all the elements in the array, which takes $O(n \log n)$ time. Then, we can use two pointers that point to the first and last elements of the sorted array, respectively, to find whether there are two elements in S that sum up to x. If the sum of the two elements that the pointers point to is less than x, we move the left pointer to the next element. Otherwise, we move the right pointer to the previous element. The time complexity of this algorithm is $O(n \log n) + O(n)$. Since $O(n \log n)$ dominates $O(n)$, the overall time complexity is $\Theta(n \log n)$.

Problem 5

1. By definition, $f(n) = O(g(n))$ means that there exist constants $c_1 > 0$ and n_1 such that:

$$|f(n)| \leq c_1 |g(n)| \quad \text{for all } n \geq n_1.$$

Similarly, $g(n) = O(h(n))$ means there exist constants $c_2 > 0$ and n_2 such that:

$$|g(n)| \leq c_2 |h(n)| \quad \text{for all } n \geq n_2.$$

Let $n_0 = \max(n_1, n_2)$. For $n \geq n_0$, combining the two inequalities, we get:

$$|f(n)| \leq c_1 |g(n)| \leq c_1 c_2 |h(n)|.$$

This shows that $f(n) = O(h(n))$ with constants $c = c_1 c_2$ and n_0. Therefore, the statement is proven.

2. ($\Rightarrow$) Assume $f(n) = O(g(n))$. By definition, there exist constants $c > 0$ and n_0 such that:

$$|f(n)| \leq c|g(n)| \quad \text{for all } n \geq n_0.$$

Dividing both sides by $|g(n)|$ (assuming $g(n) \neq 0$):

$$\left| \frac{f(n)}{g(n)} \right| \leq c \quad \text{for all } n \geq n_0.$$

This implies that $\frac{f(n)}{g(n)}$ is bounded by some constant c, which means:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = O(1).$$

($\Leftarrow$) Conversely, if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = O(1)$, it means there exists a constant $c > 0$ such that:

$$\left|\frac{f(n)}{g(n)}\right| \le c \quad \text{for all sufficiently large } n.$$

Multiplying both sides by $|g(n)|$, we get:

$$|f(n)| \le c|g(n)| \quad \text{for all sufficiently large } n.$$

This shows that $f(n) = O(g(n))$. Therefore, the statement is proven.

3. By definition:

   - $f(n) = o(g(n))$ means that for every $\epsilon > 0$, there exists an n_0 such that:

   $$|f(n)| < \epsilon|g(n)| \quad \text{for all } n \ge n_0.$$

   - $g(n) = \omega(f(n))$ means that for every $c > 0$, there exists an n_0 such that:

   $$|g(n)| > c|f(n)| \quad \text{for all } n \ge n_0.$$

   These two definitions are equivalent because $|f(n)| < \epsilon|g(n)|$ implies $|g(n)| > \frac{1}{\epsilon}|f(n)|$. Therefore, the statement is proven.

4. ($\Rightarrow$) Assume $f(n) = o(g(n))$. By definition, for every $\epsilon > 0$, there exists n_0 such that:

   $$|f(n)| < \epsilon|g(n)| \quad \text{for all } n \ge n_0.$$

   Dividing both sides by $|g(n)|$:

   $$\left|\frac{f(n)}{g(n)}\right| < \epsilon \quad \text{for all } n \ge n_0.$$

   Since this holds for any $\epsilon > 0$, we have:

   $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

   ($\Leftarrow$) Conversely, if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, then for every $\epsilon > 0$, there exists n_0 such that:

   $$\left|\frac{f(n)}{g(n)}\right| < \epsilon \quad \text{for all } n \ge n_0,$$

   which implies:

   $$|f(n)| < \epsilon|g(n)| \quad \text{for all } n \ge n_0.$$

   This shows that $f(n) = o(g(n))$. Therefore, the statement is proven.