1. Starting with 2, 6, 4, 8, 1, 7, 5, 3 we get the following percolations:
   2, 4, 6, 1, 7, 5, 3, 8 (5 swaps)
   2, 4, 1, 6, 5, 3, 7, 8 (3 swaps)
   2, 1, 4, 5, 3, 6, 7, 8 (3 swaps)
   1, 2, 4, 3, 5, 6, 7, 8 (2 swaps)
   1, 2, 3, 4, 5, 6, 7, 8 (1 swap)

Five percolations are enough to sort the sequence using 14 swaps overall.

2. (a) There are \( N - 1 \) sequences of the form
   \( a_1, \ldots, a_{i-1}, a_{i+1}, a_i, a_{i+2}, \ldots, a_N \)
   for \( i = 1, \ldots, N - 1 \) (with an obvious interpretation when \( i = 1 \) or \( i = N - 1 \)), which require just one swap to be sorted.

*(b) The input sequence is
   \( a_N, a_{N-1}, \ldots, a_2, a_1 \)
   which requires a maximum of
   \[
   (N - 1) + (N - 2) + \ldots + 2 + 1 = \frac{N(N - 1)}{2}
   \]
   swaps. Input sequences requiring exactly one less number of swaps have the form
   \( a_N, a_{N-1}, \ldots, a_{j+1}, a_{j-1}, a_j, a_{j-2}, \ldots, a_2, a_1 \)
   for \( j = N \) down to 2 (with an obvious interpretation when \( j = N \) or \( j = 2 \)).

*(c) The sequences which use exactly three swaps but more than one pass (in fact exactly two passes) of the outer loop have the form:
   \( a_1, a_2, \ldots, a_{i-3}, a_i, a_{i-1}, a_{i-2}, a_{i+1}, \ldots, a_{N-1}, a_N \)
   for \( i = 3 \ldots, N \) (with an obvious interpretation when \( i = 3 \) or \( i = N \)).

3. Use M as an abbreviation for MERGESORT. The procedure splits into
   \[ M(\cos 1, \ldots, \cos 4), \quad M(\cos 5, \ldots, \cos 8) \]
which splits into

\[ M(\cos 1, \cos 2), \ M(\cos 3, \cos 4) \mid M(\cos 5, \cos 6), \ M(\cos 7, \cos 8) \]

which splits into

\[ M(\cos 1), \ M(\cos 2) \mid M(\cos 3), \ M(\cos 4) \]

\[ M(\cos 5), \ M(\cos 6) \mid M(\cos 7), \ M(\cos 8) \]

which produces simply

\[ \cos 1, \ \cos 2 \mid \cos 3, \ \cos 4 \mid \cos 5, \ \cos 6 \mid \cos 7, \ \cos 8 \]

which merge in one step to

\[ \cos 2, \ \cos 1 \mid \cos 3, \ \cos 4 \mid \cos 5, \ \cos 6 \mid \cos 8, \ \cos 7 \]

which in turn merge in one step to

\[ \cos 3, \ \cos 4, \ \cos 2, \ \cos 1 \mid \cos 8, \ \cos 5, \ \cos 7, \ \cos 6 \]

which finally merges into the sorted list

\[ \cos 3, \ \cos 4, \ \cos 2, \ \cos 1, \ \cos 8, \ \cos 5, \ \cos 7, \ \cos 6. \]

If angles are in degrees then the original sequence is strictly decreasing so the sorted list will be

\[ \cos 8, \ \cos 7, \ \cos 6, \ \cos 5, \ \cos 4, \ \cos 3, \ \cos 2, \ \cos 1. \]

4. (a) The sequence from the first question is manipulated as follows:

\[
\begin{array}{cccc|cccc}
2, & 6, & 4, & 8, & 1, & 7, & 5, & 3 \\
2, & 4, & 6, & 8, & 1, & 3, & 5, & 7 \\
\end{array}
\]

*(b) To merge two lists of size \(2^{i-1}\), up to \(2^i - 1\) comparisons are made (because the last number is forced) and this is done \(N/2^i = 2^{k-i}\) times in the \(i\)th pass for \(1 \leq i \leq k\). Adding to this \(N\) instances of writing down each number at each pass, the running time may be estimated to be

\[
T(N) = N + \sum_{i=1}^{k} (N + 2^{k-i}(2^i - 1)) = N + \sum_{i=1}^{k} (N + 2^k - 2^{k-i})
\]

\[
= N + \sum_{i=1}^{k} O(2N) = N + (\log_2 N)O(N) = O(N \log N).
\]
*(c) Suppose the input sequence has length any positive integer $N$ which is not a power of 2. Then $2^{k-1} < N < 2^k$ for some $k$. Note that $2N > 2^k$. We can modify the algorithm by adding a prior instruction to add $2^k - N$ fillers of the form $\infty, \ldots, \infty$ to the sequence, where $\infty$ behaves as being bigger than any number. Applying the previous algorithm now yields a sorted sequence which terminates in the padding $\infty, \ldots, \infty$ which can then be deleted. The running time is, from (b), but with an adjustment for the addition and deletion of padding,

$$T(N) = O(2^k \log(2^k)) + O(2^k - N) + O(2^k - N)$$
$$= O(2N \log(2N)) + O(2N) + O(2N)$$
$$= O(N \log N + N \log 2) + O(N) = O(N \log N) .$$

5. (a) By definition $\lceil X \rceil$ is the smallest integer $\geq X$. Certainly then $\lceil X \rceil + M$ is an integer $\geq X + M$. But $\lceil X \rceil - 1$ is an integer less than $X$ so $\lceil X \rceil + M - 1$ is an integer less than $X + M$. This proves that $\lceil X \rceil + M$ is the smallest integer $\geq X + M$, whence $\lceil X + M \rceil = \lceil X \rceil + M$.

*(b) If $X$ is an integer then $X = \lceil X \rceil$ and the result is obviously true. We therefore suppose $X$ is not an integer and write

$$X = a_0 + a_1 M + a_2 M^2 + \ldots + a_N M^N$$

for some nonnegative integers $N, a_1, \ldots, a_N$ and a real number $a_0$ such that $0 < a_0 < M$. Observe that

$$0 < a_0 \leq \lceil a_0 \rceil \leq M ,$$

so that

$$0 < a_0 / M \leq \lceil a_0 \rceil / M \leq 1 ,$$

whence

$$\left[ \frac{a_0}{M} \right] = \left[ \left[ \frac{a_0}{M} \right] \right] = 1 .$$

If $X = a_0$ then we are done, so we suppose $X \neq a_0$ giving $N > 0$. Then, using (a),

$$\left[ \frac{X}{M} \right] = \left[ \frac{a_0}{M} + a_1 + \ldots + a_N M^{N-1} \right]$$
$$= \left[ \frac{a_0}{M} \right] + a_1 + \ldots + a_N M^{N-1}$$
$$= \left[ \left[ \frac{a_0}{M} \right] \right] + a_1 + \ldots + a_N M^{N-1}$$
$$= \left[ \left[ \frac{a_0}{M} \right] + a_1 + \ldots + a_N M^{N-1} \right]$$
$$= \left[ \left[ \frac{a_0}{M} + a_1 M + \ldots + a_N M^N \right] / M \right] = \left[ \left[ \frac{X}{M} \right] \right] .$$
This equation fails, for example if \( X = 1.1 = M \) for then 

\[
\left\lfloor \frac{X}{M} \right\rfloor = 1 < 2 = \left\lfloor \frac{2}{1.1} \right\rfloor = \left\lfloor \frac{X}{M} \right\rfloor.
\]

*(c) Observe that \( a_0 = \left\lfloor X \right\rfloor = \left\lfloor X/M^0 \right\rfloor \), which starts an induction. Suppose \( k \geq 0 \) and \( a_k = \left\lfloor X/M^k \right\rfloor \). Then, using (b),

\[
a_{k+1} = \left\lfloor a_k/M \right\rfloor = \left\lfloor \left\lfloor x/M^k \right\rfloor /M \right\rfloor = \left\lfloor \frac{x}{M^{k+1}} \right\rfloor,
\]

which established the inductive step. The result now follows by induction. Clearly \( \lim_{n \to \infty} a_n = 1 \).

6. (a) Use B as an abbreviation for BINARYSEARCH.

If we input \( \lambda = 32.4 \) then at step (2), B(1, \ldots, 64) halts and outputs the interval [32, 33).

If we input \( \lambda = 12.97 \) then B(1, \ldots, 64) calls on B(1, \ldots, 32) which in turn calls on B(1, \ldots, 16), which in turn calls on B(9, \ldots, 16), which halts at its step (2) and outputs the interval [12, 13).

If we input \( \lambda = 35.1 \) then B(1, \ldots, 64) calls on B(33, \ldots, 64) which in turn calls on B(33, \ldots, 48), which in turn calls on B(33, \ldots, 40), which in turn calls on B(33, \ldots, 36), which in turn calls on B(35, 36), which halts at its step (2) and outputs the interval [35, 36).

If we input \( \lambda = -1 \) then B(1, \ldots, 64) calls on B(1, \ldots, 32) which in turn calls on B(1, \ldots, 16), which in turn calls on B(1, \ldots, 8), which in turn calls on B(1, \ldots, 4), which in turn calls on B(1, 2), which in turn calls on B(1), which halts at its step (1) and outputs the interval \((-\infty, 1]\).

(b) Steps (1) and (2) collectively take \( O(1) \) time. Only one of steps (3) or (4) is executed which use \( T(N/2) \) time as the sequences \( S_L \) and \( S_R \) have length \( N/2 \). Adding these gives

\[
T(N) = T(N/2) + O(1).
\]

*(c) Observe that

\[
T(N) = T(N/2) + O(1) = T(N/4) + 2O(1) = T(N/8) + 3O(1) = \ldots = T(N/2^k) + kO(1) = T(1) + (\log_2 N)O(1) = O(1) + O(\log N) = O(\log N).
\]
*(d) The instructions remain the same for any $N$. However we adjust the definitions of $S_L$, $S_R$, $M_L$ and $M_R$ as follows:

$$S_L = \{a_1, \ldots, a_{\lceil N/2 \rceil}\}, \quad S_R = \{a_{\lfloor N/2 \rfloor + 1}, \ldots, a_N\},$$

$$M_L = a_{\lceil N/2 \rceil}, \quad M_R = a_{\lfloor N/2 \rfloor + 1}.$$ 

Note that step (2) can only be executed at the moment $B(S)$ is called if $S$ has an even number of elements, since if the size of $S$ is odd then $M_L = M_R$.

**(e)** The derivation of the recurrence is the same as in part (b) except that at steps (3) and (4) the new sequences $S_L$ and $S_R$ both have size $\lceil N/2 \rceil$. Consider now any positive integer $N$, so

$$2^{k-1} < N \leq 2^k.$$ 

for some nonnegative $k$. Note that $2^k < 2N$ so $k < \log_2 N + 1$, giving $k = O(\log N)$. Iterating the recurrence, and using part (b) of the previous question with $M = 2$, we get

$$T(N) = T(\lceil N/2 \rceil) + O(1)$$

$$= T\left(\lceil \frac{\lceil N/2 \rceil}{2} \rceil\right) + 2O(1)$$

$$= T\left(\left\lceil \frac{N}{4} \right\rceil\right) + 2O(1)$$

$$= T\left(\left\lceil \frac{\lceil N/4 \rceil}{2} \right\rceil\right) + 3O(1)$$

$$= T\left(\left\lceil \frac{N}{8} \right\rceil\right) + 3O(1)$$

$$= \ldots$$

$$= T\left(\left\lceil \frac{N}{2^k} \right\rceil\right) + kO(1)$$

$$= T(1) + kO(1)$$

$$= O(1) + O(k)$$

$$= O(\log N).$$