Name: KimJaeHwan
Student ID: 20230499

CSED261: Discrete Mathematics for Computer Science
Homework 4: Algorithms

Question 1. Use the bubble sort to sort 6, 2, 3, 1, 5, 4, showing the lists obtained at each step.

Solutions

Bubble sort (Ascending) ((for q4, not use caption))

```
N be size of the list
for i = 1 to N-1 do
    for j = 1 to N-i do
        if l thenist[i] >list[i+1] then
            Swap list[i] and list[i+1]
        end if
    end for
end for
```

Criterion of step is i in pseudocode.

Input	6, 2, 3, 1, 5, 4
Step 1	2, 6, 3, 1, 5, 4
Step 1	2, 3, 6, 1, 5, 4
Step 1	2, 3, 1, 6, 5, 4
Step 1	2, 3, 1, 5, 6, 4
Step 1	2, 3, 1, 5, 4, 6
Step 2	2, 1, 3, 5, 4, 6
Step 2	2, 1, 3, 4, 5, 6
Step 3	1, 2, 3, 4, 5, 6

So, we get ascending ordered list.

Question 2. Compare the number of comparisons used by the insertion sort and the binary insertion sort to sort the list 7, 4, 3, 8, 1, 5, 4, 2.

Solutions

Insertion sort (Ascending order)

Input	7, 4, 3, 8, 1, 5, 4, 2	Comparison
Step 1	((4, 7)), 3, 8, 1, 5, 4, 2	(4, 7) change
Step 2	(4, (3, 7)), 8, 1, 5, 4, 2	(3, 7) change
Step 2	((3, 4), 7), 8, 1, 5, 4, 2	(3, 4) change
Step 3	(3, 4, (7, 8)), 1, 5, 4, 2	(7, 8) no-change
Step 4	(3, 4, 7, (1, 8)), 5, 4, 2	(1, 8) change
Step 4	(3, 4, (1, 7), 8), 5, 4, 2	(1, 7) change
Step 4	(3, (1, 4), 7, 8), 5, 4, 2	(1, 4) change
Step 4	((1, 3), 4, 7, 8), 5, 4, 2	(1, 3) change
Step 5	(1, 3, 4, 7, (5, 8)), 4, 2	(5, 8) change
Step 5	(1, 3, 4, (5, 7), 8), 4, 2	(5, 7) change
Step 5	(1, 3, (4, 5), 7, 8), 4, 2	(4, 5) no-change
Step 6	(1, 3, 4, 5, 7, (4, 8)), 2	(4, 8) change
Step 6	(1, 3, 4, 5, (4, 7), 8), 2	(4, 7) change
Step 6	(1, 3, 4, (4, 5), 7, 8), 2	(4, 5) change
Step 6	(1, 3, (4, 4), 5, 7, 8), 2	(4, 4) no-change
Step 7	(1, 3, 4, 4, 5, 7, (2, 8))	(2, 8) change
Step 7	(1, 3, 4, 4, 5, (2, 7), 8)	(2, 7) change
Step 7	(1, 3, 4, 4, (2, 5), 7, 8)	(2, 5) change
Step 7	(1, 3, 4, (2, 4), 5, 7, 8)	(2, 4) change
Step 7	(1, 3, (2, 4), 4, 5, 7, 8)	(2, 4) change
Step 7	(1, (2, 3), 4, 4, 5, 7, 8)	(2, 3) change
Step 7	((1, 2), 3, 4, 4, 5, 7, 8)	(1, 2) no-change

Binary insertion sort (Ascending order)

Input	7, 4, 3, 8, 1, 5, 4, 2	Comparison
Step 1	7	Initial input
Step 2	4, 7	(4, 7) compare (mid : 4)
Step 3	3, 4, 7	(4, 3) compare (mid : 4)
Step 4	3, 4, 7, 8	(4, 8) compare (mid : 4)
Step 4	3, 4, 7, 8	(7, 8) compare (mid : 7)
Step 5	1, 3, 4, 7, 8	(4, 1) compare (mid : 4)
Step 5	1, 3, 4, 7, 8	(3, 1) compare (mid : 3)
Step 6	1, 3, 4, 5, 7, 8	(4, 5) compare (mid : 4)
Step 7	1, 3, 4, 4, 5, 7, 8	(4, 4) compare (mid : 4)
Step 8	1, 2, 3, 4, 4, 5, 7, 8	(4, 2) compare (mid : 4)
Step 8	1, 2, 3, 4, 4, 5, 7, 8	(3, 2) compare (mid : 3)

Conclusion : The number of comparisons used by the insertion sort is 22 and the number of comparisons used by the binary insertion sort is 11. The binary insertion sort is faster than the insertion sort.

Question 3. Show that each of these pairs of functions are of the same order.

1. $3x + 7, x$
2. $\log(x^2 + 1), \log_2 x$

Solutions

1. $3x + 7, x$ Generally, we can use limit of the ratio of two functions to show that pairs of functions are of the same order

$$\lim_{x \to \infty} \frac{3x + 7}{x} = 3$$

Since the limit is a (nonw-zero) constant, $3x + 7$ and x are of the same order.

2. $\log(x^2 + 1), \log_2 x$ Let's use same method.

$$\lim_{x \to \infty} \frac{\log(x^2 + 1)}{\log_2 x} = \lim_{x \to \infty} \frac{\log(x^2 + 1)}{\frac{\log x}{\log 2}} = \lim_{x \to \infty} \frac{\log(x^2 + 1)}{\log x} \cdot \log 2$$

To compute the limit of fraction having log in numerator and denominator, we can use L'Hopital's rule:

$$\lim_{x \to \infty} \frac{f(x)}{g(x)} = \lim_{x \to \infty} \frac{f'(x)}{g'(x)}$$

So we get:

$$\lim_{x \to \infty} \frac{\log(x^2 + 1)}{\log x} = \lim_{x \to \infty} \frac{\frac{2x}{x^2+1}}{\frac{1}{x}} = \lim_{x \to \infty} \frac{2x^2}{x^2 + 1} = 2$$

So, the limitation of ratio of two functions is a non-zero constant, $2 \log 2$, so $\log(x^2 + 1)$ and $\log_2 x$ are of the same order.

Question 4. An algorithm is called optimal for the solution of a problem with respect to a specified operation if there is no algorithm for solving this problem using fewer operations.

Algorithm 1 Finding the Maximum Element in a Finite Sequence

```
procedure max(a_1, a_2, ..., a_n: integers)
    max := a_1
    for i := 2 to n do
        if max < a_i then
            max := a_i
        end if
    end for
    return max                                  ▷ max is the largest element
end procedure
```

1. Show that Algorithm 1 is an optimal algorithm with respect to the number of comparisons of integers.
2. Is the linear search algorithm optimal with respect to the number of comparisons of integers?

Solutions

1. Let's assume that there is another algorithm 2 that can find max more efficiently than the given algorithm. However, Algorithm 2 also be required to compare $n-1$ times to find the max in worst case, so it is $O(n)$. Therefore, the given algorithm is optimal with respect to the number of comparisons of integers.
2. Let n be the size of the list. Then, the linear search algorithm is required to compare n times to find the max in worst case, so it is $O(n)$. However, if input is sorted, we can find the max in $O(1)$ time and any value in $O(\log n)$ time using binary search. Therefore, the linear search algorithm is not optimal with respect to the number of comparisons of integers.