Aikeboer Aizezi
1310044086

CSE 321_Introduction to Algorithm Design

Homework_02

1. Sort the array A={3, 44, 38, 5, 47, 15} in increasing order by Selection sort, bubble sort, insertion sort, quick sort, show steps as well.

Selection Sort:

① | 3 | 44 | 38 | 5 | 47 | 15 |

② 3, 5, 38, 44, 47, 15

③ 3, 5, 15, 44, 47, 38

④ 3, 5, 15, 38, 47, 44

⑤ 3, 5, 15, 38, 44, 47

⑥ 3, 5, 15, 38, 44, 47

bubble Sort:

3, 44, 38, 5, 47, 15

① 3, 44, 38, 5, 47, 15

② 3, 38, 44, 5, 47, 15

③ 3, 38, 5, 44, 47, 15

④ 3, 38, 5, 44, 47, 15

⑤ 3, 38, 5, 44, 15, 47

⑥ 3, 5, 38, 44, 15, 47

⑦ 3, 5, 38, 44, 15, 47

⑧ 3, 5, 38, 15, 44, 47

⑨ 3, 5, 38, 15, 44, 47

⑩ 3, 5, 15, 38, 44, 47

⑪ 3, 5, 15, 38, 44, 47

: not changed

sorted: 3, 5, 15, 38, 44, 47

Insertion sort:

3, 44, 38, 5, 47, 15

① 3, 44, 38, 5, 47, 15

② 3, 44, 38, 5, 47, 15

③ 3, 38, 44, 5, 47, 15

④ 3, 38, 44, 5, 47, 15

⑤ 3, 5, 38, 44, 47, 15

⑥ 3, 5, 15, 38, 44, 47

quick sort:

3, 44, 38, 5, 47, 15

① 3, 44, 38, 5, 47, 15

② 3, 44, 38, 5, 15, 47

3, 15, 38, 5, 44, 47

③ 3, 15, 5, 38, 44, 47

3, 5, 15, 38, 44, 47

④ 3, 5, 15, 38, 44, 47

⑤ 3, 5, 15, 38, 44, 47

⑥ 3, 5, 15, 38, 44, 47

2. a) Is selection sort stable?

Selection sort isn't stable. Because the swap that occurs at the end of each "round" can change the relative order of items having the same value.

b) Is bubble sort stable?

Bubble sort is stable, because it only swaps elements that are of different values. It follows from the fact that it swaps adjacent elements only, provided $A[j+1] \geq A[j]$.

c) Yes, to finding the smallest element and swapping it both can be done as efficiently with the linked list.

d) Yes, it is possible, but, while inserting A[i] we will have to scan the sorted part left to right.

3. Algorithm:

Starting with the first and ending with the last light disk, swap it with each of the i ($1 \leq i \leq n$) dark disks to the left of it. The ith iteration of the algorithm can be illustrated by the following diagram, in which 1's and 0's correspond the dark and light disks, respectively.

00...011...11010...10

$i-1$ $i-1$

$\Rightarrow$ 00...0011...1110...10

i i

The total number of swaps: $\sum_1^n i = \frac{n(n+1)}{2}$

4. The worst-case input for this problem is n zeros and the pattern 0...01 (which is, let's say, j-1) ●, the total character comparision on such input is:

$$\sum_{j=1}^{n} j = \frac{j(n-j+1)}{2}.$$

5. a) Initialize the number of substrings to 0. Scan the give string left to right, apply this algorithm to every characters except the last one:

If an A is found, count all the B's next to it, and add it to the count of substring. When came to the end of the target string, return the count.

worst-case: $\sum_{i=1}^{n} i = \Theta(n^2)$.

b) Initialize the count of substrings and A's to 0. Scan the given string lef to right.

If an A is seen, add 1 to the A's count. If B is seen, add the count of A to the count of substring.

Return the count of substring.

The algorithm is linear, $T(n) = \Theta(1)$.

6. Firs of all, the two boys take the boat to the other side, after which one of them returns with the boat. Then a soldier takes the boat to the other side and stays there while the other boy returns the boat. These four trips reduce the problems instance of size n to n-1. So, if this four-trip procedure repeated n times, the problem will be solved after the total 4n trips.

7. Begin with all n people as candidates to be a celebrity. Ask a random person, x if he knows random person y. If so eliminate x as a candidate to be a celebrity. If not, eliminate y as a candidate to be a celebrity. Continue in this manner until, after n-1 people questions, there is only one candidate to be a celebrity. then check with each of the other n-1 people to confirm that they all know that person. And with the person to confirm that she does not know any of the other people. Thus, it takes at most 3n-3 questions to identify a celeb. if such a person exists.

$$T(n) = O(n).$$

8. The code is provided.

9. Algrithm:

```
maxProduct(int n)        // calculating x for which
                         // f(x) = x(n-x) takes maximum value.
    if(n/2 = 0)
        x = n/2
    else
        x = ⌊n/2⌋
    return x
```