National University of Computer and Emerging Sciences

FAST School of Computing   Fall-2022   Islamabad Campus

Question 1 [20 Marks]

1. Consider the following code snippet of pancake sorting algorithm.

temp 1 8 7
start 0 1
a

```
void flip(int arr[], int i)
{
    int temp, start = 0;
    while (start < i) {
        temp = arr[start];
        arr[start] = arr[i];
        arr[i] = temp;
        start++;
        i--;
    }
}
int findMax(int arr[], int n)
{
    int mi, i;
    for (mi = 0, i = 0; i < n; ++i)
        if (arr[i] > arr[mi])
            mi = i;
    return mi;
}
void pancakeSort(int* arr, int n)
{
    for (int curr_size = n; curr_size > 1; --curr_size)
    {
        int mi = findMax(arr, curr_size);

        if (mi != curr_size - 1) {
            flip(arr, mi);  // Flip Up
            flip(arr, curr_size - 1); // Flip Down
        }
    }
}
int main()
{
    int arr[] = { 1, 4, 5, 2, 3, 8, 6, 7, 9, 0 };
    int n = sizeof(arr) / sizeof(arr[0]);
    pancakeSort(arr, n);
    return 0;
}
```

0, 0

0 1 0

0

mi i
0 0
0 1
0 2
0 3
2 4
2 5
2
0 6
2 6
6
0 7 6

mi i
0 0
0 1 1
0 1 2

mi i
0 1
1 2
2 3
3 4
3 5
3 6
6 7
4 5
0 1

9

arr 8

a) Perform dry run and write the output of each iteration after flip up and flip down in the table below. [8 marks]

9 ~~7~~ 4 ~~2~~5 2 3 8 6 ~~0~~$\overset{7}{1}$ ~~9~~, 0

rr_size	mi	arr after `flip(arr,mi)`	arr after `flip(arr, curr_size-1)`
10	8	9, 7, 6, 8, 3, 2, 5, 4, 1, 0	0, 1, 4, 5, 2, 3, 8/ 6, 7, 9
~~9~~	~~8~~	~~7, 6, 8, 3, 2, 5, 4, 1, 0, 9~~	~~0, 1, 4, 5~~
9	6 ✓	8, 3, 2, 5, 4, 1, 0, 6, 7/9	7, 6, 0, 1, 4, 5, $\overset{2}{0}$, 3, 8/, 9
8	0 ✓	7, 6, 0, 1, 4, 5, $\overset{12}{0}$, 3/, 8, 9	3, $\overset{2}{0}$, 5, 4/ 1, 0, 6/ 7/ 8, 9
7	6 ✓	6, 0, 1, 4, 5, $\overset{2}{0}$, 3, 7, 8, 9	3, $\overset{2}{0}$, 5/, 4, 1, 0, /6, 7, 8, 9
6	2 ✓	5, 0, 3, 4, 1, $\overset{2}{0}$/, 6, 7, 8, 9	0, 1, 4, 3, /0/, 5, 6, 7, 8, 9
5	2 ✓	4, 1, 0, 3, $\overset{2}{0}$, 5, 6, 7, 8, 9	0, 3, $\overset{2}{0}$, 1, 4, 5, 6, 7, 8, 9
4	1 ✓	3, 0, $\overset{2}{0}$, 1, 4, 5, 6, 7, 8, 9	1, 0, $\overset{2}{0}$, 3, 4, 5, 6, 7, 8, 9
~~0~~3	~~0~~	0, 1, 0, 0, 3, 4, 5, 6, 7, 8, 9	0, 0, 1, 2, 3, 4, 5, 6, 8, 9
			05
	09		

0-1/2

b) What is the worst-case time complexity of the above algorithm? Justify your answer by providing the time complexity of flip and findMax operations. [2 marks]

Max runs till n, n-1, ... 0
Flip up till mi
Flip down till n ... 0
Pancake itself has ~~an~~ n loop
so n^4

. Consider the following code snippet.

4, 5, 4, 2, 8, 0 2

```
void CocktailSort(int a[], int n)
{
    bool swapped = true;
    int start = 0;
    int end = n - 1;

    while (swapped) {
        swapped = false;
        for (int i = start; i < end; ++i) { //Forward pass
            if (a[i] > a[i + 1]) {
                swap(a[i], a[i + 1]);
                swapped = true;
            }
        }
        if (!swapped)
            break;
        swapped = false;
        --end;
        for (int i = end - 1; i >= start; --i) { //backward pass
            if (a[i] > a[i + 1]) {
                swap(a[i], a[i + 1]);
                swapped = true;
            }
        }
        ++start;
    }
}
int main()
{
    int a[] = { 5, 1, 4, 2, 8, 0, 2 };
    int n = sizeof(a) / sizeof(a[0]);
    CocktailSort(a, n);
    return 0;
}
```

5, 1, 4, 2, 8, 0 2
1, 4, 2, 5, 0, 2, 8
1, 2, 4, 0, 2, 5, 8
1, 2, 0, 2, 4, 5, 8
1, 0, 2, 2, 4 5, 8
0, 1, 2, 2, 4, 5, 6

5, 1, 4, 2, 8, 0, 2
~~1~~ 4 2 5 0 0 2 6
~~1~~ ~~2~~ 4 0, 2, 5 8
1 ~~2~~ ~~0~~ 4 2, 4 5 8
1 0, 2 2, 4, 5, 8

a) Perform dry and write the output after each forward and backward pass in the table below. [5 marks]

5, 4, 4, 2, 8, 0, 2   7   6   6 5 5

002

National University of Computer and Emerging Sciences

FAST School of Computing   Fall-2022   Islamabad Campus

Pass (Forward/backward)	Starting value of i	a (Elements in array after the pass)
Foward	0	1, 4, 2, 5, 0,
F	1	1, 2, 4, 0, 2, 5
F	2	1, 2, 0, 4, 2, 5, 8
F	3	1, 2, 0, 2, 4, 5, 8
F	4	1, 2, 0, 2, 4, 5, 8
B B	5 8	0, 1, 2, 2, 4, 5, 8

0 1

b) What is the best-case time complexity of the above algorithm? Justify your answer! [2 marks]

O(n) because if array is sorted the two loops are run once to verify so n+n = 2n

✓ 02

c) What is the worst-case time complexity of the above algorithm? Justify your answer! [1 mark]

$O(n^2)$

✓ 01

d) What problem in the Bubble-Sort does the Cocktail-Sort tend to solve to improve the performance? [2 marks]

reduce number of swaps

✗

Question 2 [20 Marks]

1. You are given a multidimensional array containing information regarding final exam scores of top students of fast from different campuses. Your array is given as

int Campus=5, school=3, students=10, final_scores=6;
double score [campus][school][students][final_scores] 180

a) Given the base Address as 1000, find the address of score [2][1][7][4]. [7 Marks]

~~4~~
~~42~~
~~1 60~~
~~360~~
~~466 × 8 = 3728~~

6
42
60
360
448 × 8 = 3 5 35 7 6

~~Formula is (~~ ~~A+~~ ~~(~~

~~+ (1 * (final_scores * students) + ((7 * final_s) + 4))~~

Formula is –

$$(A + (\overset{2}{\cancel{5}} * (\overset{3}{\text{school}} * \overset{10}{\text{students}}) * \overset{6}{\text{final\_scores}}) + (1 * (\overset{10}{\text{students}} * \overset{6}{\text{final\_scores}}))$$

$$+ (7 * \overset{6}{\text{final\_scores}}) + (\overset{?}{\boxed{6}})) * \text{size\_of (double)}$$

How?

&score[2][1][7][4] = 4728 3.5

Consider the following code snippet:

```
void foo(int A[], int n)
{
        if (n < 1) return;
        int write_index = n - 1;
        int read_index = n - 1;

        while(read_index >= 0)
        {
                if(A[read_index] != 0)
                {
                        A[write_index] = A[read_index];
                        write_index--;
                }
                read_index--;
        }

        while(write_index >= 0)
        {
                A[write_index] = 0;
                write_index--;
        }
}
```

wr 8 8 7 6 7
rea 8 7 6 ~~6~~ 5

5 5 4 3
3 2 1 0

8
8

a) Perform the complete dry run on the given algorithm and show array content on each iteration. Assume contents of array are {1, 10, 20, 0, 59, 63, 0, 88, 0}. [6 marks]

Iteration	Array
0	1, 10, 20, 0, 59, 63, 0, 88, 0
1	1, 10, 20, 0, 59, 63, 0, 0, 88
2	1, 10, 20, 0, 59, 63, 0, 0, 88
3	1, 10, 20, 0, 59, 0, 0, 63, 88
4	1, 10, 20 0, 0, 0 59, 63, 88
5	1, 10, 0, 0, 0, 20, 59, 63, 88
6	1, 0, 0, 0 10 20, 59, 63, 88
7	0, 0, 0, 1, 10, 20, 59, 63, 88
8	0, 0, 0, 1, 10, 20, 59, 63, 88
9	0 0 0 1 10 20 59 63 88
10	0 0 0 1 10 20 59 63 88
11	0

4.5?

Missing iterations

b) What is the purpose of the given algorithm? [2 Marks]

to bring all 0 elements to left 2

c) What will be the final output of foo()? [1 Mark]

nothing as void

d) What is the complexity of given code in terms of Big-Oh? [2 Marks]

n^2

e) What will be the best-case scenario for the given algorithm? [2 Marks]

n

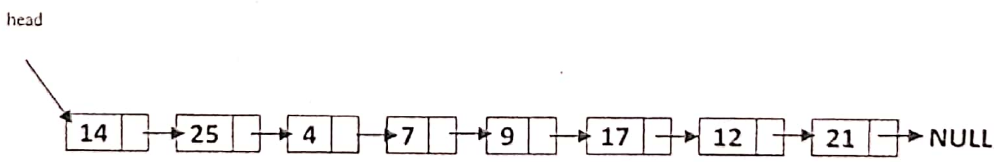
Question 3 [20 Marks]

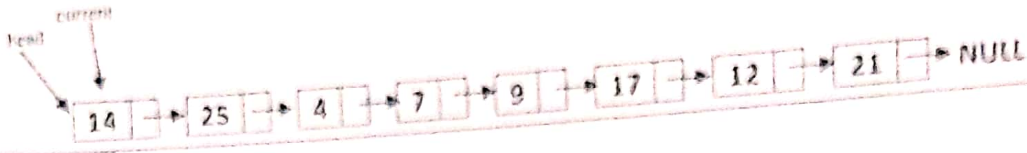
1. Consider the following code snippet:

```
void list::Game1()
{
        node* headref = head;
        genrated = NULL;
        node* current = headref;
        while (current != NULL) {
                node* next = current->next;
                Game2(current);
                current = next;
        }
        head = genrated;
}

void list::Game2(node* newnode)
{
                Scope                                    =
        if (genrated == NULL || genrated->data >= newnode->data) {
                newnode->next = genrated;                ^
                genrated = newnode;
        }
        else {
                node* current = genrated;
                while (current->next != NULL
                        && current->next->data < newnode->data) {
                        current = current->next;
                }
                newnode->next = current->next;
                current->next = newnode;
        }
}
```

a) Given a linked list (given below), perform a complete dry run of the algorithm and at each iteration, display the structure of linked list. [7 Marks]



Iteration	Linked List (Structure)
	generated=NULL  head current 14 25 4 7 9 17 12 21 NULL

b) What is the wors complexity of the above code in terms of Big-Oh? Justify your answer! [1 Marks]

(n²) if the while loop is run for each node and else case of ~~Oam~~ 2 is triggered

1

c) What is the best-case time complexity of the above algorithm? Justify your answer! [2 Marks]

o(1) if list is empty

d) What is the purpose of the given algorithm? and write down the name of algorithm. [2 Marks]

2. Given the following code of selection sort.

```
void Func()
{
        node* ptr_1, * ptr_2, * min;
        ptr_1 = head;
        while (ptr_1->next != NULL)
        {
            ptr_2 = ptr_1              <-- Statement is missing here
            min = nullptr              <-- Statement is missing here
            while (ptr_2 != NULL)
            {
                    if (min->data > ptr_2->data)
                            min = ptr_2;
                    ptr_2 = ptr_2->next;
            }
            int tmp = ptr_1->data;
            ptr_1->data = min->data;
            min->data = tmp;
            ptr_1 = ptr_1->next                <-- Statement is missing here
        }
}
```

2

a) Add the missing lines of code to the above-mentioned function. [3 Marks]

b) What is the complexity of the above code in terms of Big-Oh? [1 Marks]

n^2

0120

National University of Computer and Emerging Sciences

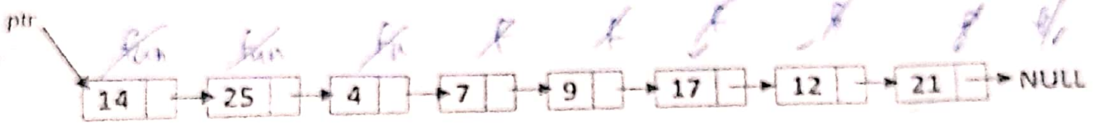
FAST School of Computing Fall-2022 Islamabad Campus

3. Consider the following function

```
void fun(node* ptr)
{
        if (ptr == NULL)
                return;
        fun(ptr->next);
        if(ptr->data%2==0)
                cout << ptr->data << " ";
}
```

0

a) Given a linked list (given below), perform the complete dry run of the algorithm and display the output. [3 Marks]



0 fun (14) -> fun(25) ->(fun 4) ->(fun 7) ->(fun 9) ->(fun 17)

fun (12) -> fun (21) -> Null

21 % 2 != 0 , 12 % 2 = = 0, 17 % 2 != 0

9 % 2 != 0 , 7 != 0 , 4 % 2 = = 0, 25 % 2 != 0 , 14 % 2 == 0

12 4 14

3

1

b) What is the complexity of the above code in terms of Big-Oh? [1 Marks]

O(n)