

## A3-B1-01

### DAA Lab Pract 1

Aim: Time and complexity analysis of loops for a sensor data monitoring system by generating random sensor readings such as temperature, and pressure. The goal is to analyze and compare the performance of different algorithms.

Codes:

1. Pair-wise:

```
2. #include<stdio.h>
3. int findMax(int* arr,int n){
4.     int temp[(n+1)/2];
5.     int count = n;
6.     int* curr;
7.     curr = arr;
8.     while(count != 1){
9.         for(int i = 0; i < count; i = i+2){
10.            if(i+1 < count){
11.                if(curr[i] < curr[i+1]){
12.                    temp[i/2] = curr[i+1];
13.                }else{
14.                    temp[i/2] = curr[i];
15.                }
16.            }else{
17.                temp[i / 2] = curr[i];
18.            }
19.        }
20.        count = (count+1)/2 ;
21.        curr = temp;
22.    }
23.    return temp[0];
24.}
25.int findMin(int* arr,int n){
26.    int temp[(n+1)/2];
27.    int count = n;
28.    int* curr;
29.    curr = arr;
30.    while(count != 1){
31.        for(int i = 0; i < count; i = i+2){
32.            if(i+1 < count){
33.                if(curr[i] > curr[i+1]){
34.                    temp[i/2] = curr[i+1];
35.                }else{
36.                    temp[i/2] = curr[i];
```

```

37.         }
38.     }else{
39.         temp[i / 2] = curr[i];
40.     }
41. }
42.     count = (count+1)/2 ;
43.     curr = temp;
44. }
45.     return temp[0];
46.}
47.int main(){
48.    int sensor1[5] = {4,-8,20,35,75};
49.    int sensor2[4] = {10,20,-20,4.5};
50.    int sensor3[4] = {25,7,38,6};
51.    int sensor4[4] = {40,-3,2,80};
52.    int sensor5[4] = {50,20,-10,20};
53.
54.    printf("sensor1 max reading: %d\tmin reading %d\n",findMax(&sensor1[0],sizeof(sensor1)/sizeof(sensor1[0])),findMin(&sensor1[0],sizeof(sensor1)/sizeof(sensor1[0])));
55.    printf("sensor2 max reading: %d\tmin reading %d\n",findMax(&sensor2[0],sizeof(sensor2)/sizeof(sensor2[0])),findMin(&sensor2[0],sizeof(sensor2)/sizeof(sensor2[0])));
56.    printf("sensor3 max reading: %d\tmin reading %d\n",findMax(&sensor3[0],sizeof(sensor3)/sizeof(sensor3[0])),findMin(&sensor3[0],sizeof(sensor3)/sizeof(sensor3[0])));
57.    printf("sensor4 max reading: %d\tmin reading %d\n",findMax(&sensor4[0],sizeof(sensor4)/sizeof(sensor4[0])),findMin(&sensor4[0],sizeof(sensor4)/sizeof(sensor4[0])));
58.    printf("sensor5 max reading: %d\tmin reading %d\n",findMax(&sensor5[0],sizeof(sensor5)/sizeof(sensor5[0])),findMin(&sensor5[0],sizeof(sensor5)/sizeof(sensor5[0])));
59.}

```

Time complexity:  $O(n)$

- Space complexity:  $O(n)$

## 2. Linear Search

```
#include<stdio.h>
```

```
int findMax(int* arr,int n){
```

```

int max;
for(int i=0;i<n;i++){
    if(arr[i]> max){
        max = arr[i];
    }
}
return max;
}

```

```

int findMin(int* arr,int n){
    int min;
    for(int i=0;i<n;i++){
        if(arr[i]< min){
            min = arr[i];
        }
    }
    return min;
}

```

```

int main(){
    int sensor1[5] = {4,-8,20,35,75};
    int sensor2[4] = {10,20,-20,4.5};
    int sensor3[4] = {25,7,38,6};
    int sensor4[4] = {40,-3,2,80};
    int sensor5[4] = {50,20,-10,20};

    printf("sensor1 max reading: %d\tmin reading %d\n",findMax(&sensor1[0],sizeof(sensor1)/sizeof(sensor1[0])),findMin(&sensor1[0],sizeof(sensor1)/sizeof(sensor1[0])));
}

```

```
    printf("sensor2 max reading: %d\tmin reading %d\n",findMax(&sensor2[0],sizeof(sensor2)/sizeof(sensor2[0])),findMin(&sensor2[0],sizeof(sensor2)/sizeof(sensor2[0])));

    printf("sensor3 max reading: %d\tmin reading %d\n",findMax(&sensor3[0],sizeof(sensor3)/sizeof(sensor3[0])),findMin(&sensor3[0],sizeof(sensor3)/sizeof(sensor3[0])));

    printf("sensor4 max reading: %d\tmin reading %d\n",findMax(&sensor4[0],sizeof(sensor4)/sizeof(sensor4[0])),findMin(&sensor4[0],sizeof(sensor4)/sizeof(sensor4[0])));

    printf("sensor5 max reading: %d\tmin reading %d\n",findMax(&sensor5[0],sizeof(sensor5)/sizeof(sensor5[0])),findMin(&sensor5[0],sizeof(sensor5)/sizeof(sensor5[0])));

}
```

the time complexity for each function is  $O(n)$ , where  $n$  is the size of the array.

the space complexity is  $O(1)$ , constant space