

Question 1

Q 1.1: Given an array of integers. Find the first and second smallest integers from an array. You can answer this in either pseudocode or actual code in any programming languages.

Example input: [13, 1, 15, 2, 99, 87, 101, 555, -1, 11]

Expected output: The first smallest is -1 and the second is 1

A: Use quicksort to solve this problem

```
3 void swap(int *a,int *b)
4 {
5     int t = *a;
6     *a = *b;
7     *b = t;
8 }
9
10 int findpivot(int A[],int low,int hight)
11 {
12     int pivot,i,j;
13     pivot = A[low];
14     i = low;
15     j = hight+1;
16     while(1)
17     {
18         do{
19             ++i;
20         }while(A[i]<=pivot&& i<=hight);
21         do{
22             --j;
23         }while(A[j]>pivot);
24         if(i >= j)
25         {
26             break;
27         }
28         swap(&A[i],&A[j]);
29     }
30
31     swap(&A[low],&A[j]);
32     return j;
33 }
34
35 int quicksort(int A[],int low,int hight)
36 {
37     if(low<hight)
38     {
39         pi = findpivot(A,low,hight);
40         quicksort(A,low,pi-1);
41         quicksort(A,pi+1,hight);
42     }
43 }
44
45 int main()
46 {
47     int a[] = {13,1,15,2,99,87,101,555,-1,11};
48     int size = sizeof(a)/sizeof(a[0]);
49     quicksort(a,0,size-1);
50     for(int i=0;i<size;i++)
51     {
52         cout<<a[i]<<" ";
53     }
54     cout<<"\n";
55     cout<<"The first smallest is "<<a[0]<<" and the second is "<<a[1];
56     return 0;
57 }
```

C:\Users\s-tan\OneDrive - kmutnb.ac.th\Documents\Blockint\Untitled1.exe

-1 1 2 11 13 15 87 99 101 555
The first smallest is -1 and the second is 1

Process exited after 0.07289 seconds with return value 0
Press any key to continue . . .

Q 1.2: What is the Big-O complexity of your solution in Q 1.1?

A: $\theta(n \log(n))$

Question 2

Q: What is wrong with this component, and how would you go fixing or improving it?

```
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      name: this.props.name || 'Anonymous'
    }
  }

  render() {
    return (
      <p>Hello {this.state.name}</p>
    );
  }
}
```

A: It's not wrong component but if the value in props.name has changed more than once the value in state.name will not be changed so to fixing use this component

```
5  class App extends React.Component{
6
7    static defaultProps = {
8      name: 'Anonymous'
9    }
10   render(){
11
12     return(
13       <p>Hello {this.props.name} </p>
14     );
15   }
16 }
17
18
```

This component will be make value in props.name change if have value sent to props **more one time**

Question 3

Q: What should be filled in the code below?

*Note: If you aren't familiar with React Hooks, feel free to answer the way to do this in the class component.

```
import React from 'react';

function Question3() {
  const [counterA, setCounterA] = React.useState(0);
  const [counterB, setCounterB] = React.useState(0);

  /**
   * Fill the code here to log each counter value to console whenever
   their value change
   * Only log counterA value if counterA's value changes
   * Only log counterB value if counterB's value changes
   */

  return (
    <div>
      <button
        onClick={() => setCounterA(counterA + 1)}
      >
        A
      </button>
      <button
        onClick={() => setCounterB(b => b + 1)}
      >
        B
      </button>
    </div>
  );
}
```

A:

```
74 function Question3(){
75
76   const [counterA, setCounterA] = React.useState(0);
77   const [counterB, setCounterB] = React.useState(0);
78
79   useEffect(()=>{
80     console.log("counterA = " + counterA)
81   },[counterA])
82   useEffect(()=>{
83     console.log("counterB = " + counterB)
84   },[counterB])
85
86   return(
87     <div>
88       <button onClick={()=>setCounterA(counterA +1)}>
89         A
90       </button>
91       <button onClick={()=>setCounterB(b => b + 1)}>
92         B
93       </button>
94     </div>
95   )
96 }
97 export default Question3;
98
```