

Name: Jingyi Ouyang

PID: A53108909

```
1)  num=0;
    for (i = n; i ≥ 0; i--)
        num++;
```

Answer: Running time is $O(n)$

Explanation: There is a single loop running n times. Each time the loop runs it executes 1 instruction in the loop header and 1 instruction in the body of the loop.

The total number of instruction is $2*n + 1$ (for the last loop check) = $O(n)$.

```
2)  num=0;
    for (i = 1; i ≤ n*n; i=i+2)
        num++;
```

Answer: Running time is $O(n^2)$

Explanation: There is a single loop running $(n^2)/2$ times. Each time the loop runs it executes 1 instruction in the loop header and 1 instruction in the body of the loop.

The total number of instruction is $(n^2)/2 + (n^2)/2 + 1$ (for the last loop check) = $O(n^2)$.

```
3)  num=0;
    for (i = 1; i ≤ n; i=i*2)
        num = num + n;
```

Answer: Running time is $O(\log n)$

Explanation: There is a single loop running $\log n$ times. Each time the loop runs it executes 1 instruction in the loop header and 1 instruction in the body of the loop.

The total number of instruction is $2 \cdot \log n + 1$ (for the last loop check) = $O(\log n)$.

```
4)  num=0;
    for (i = 1; i ≤ 100; i++)
        for (j = 1; j ≤ n; j++)
            num = num + i;
```

Answer: Running time is $O(n)$

Explanation: This is a nested loop. Each time the outer loop runs, it executes 1 instruction in the loop header and 1 inner loop. Each time the inner loop runs, it executes 1 instruction in the loop header and 1 instruction in the body of the loop.

The total number of instruction is $100(n+1) + 1$ (for the last loop check) = $O(n)$.

```
5)  num=0;
    for (i = 1; i ≤ n; i++)
        for (j = 1; j ≤ i; j++)
            num = num + i;
```

Answer: Running time is $O(n^2)$

Explanation: This is a nested loop. Each time the outer loop runs, it executes 1 instruction in the loop header and 1 inner loop. Each time the inner loop runs, it executes 1 instruction in the loop header and 1 instruction in the body of the loop.

The total number of instruction is $(1+n+1) \cdot (n+1)/2 = O(n^2)$.

```
6)  num=0;
    for (i = 1; i ≤ n; i++)
        for (j = 1; j ≤ n; j=j*2)
            num = num + i;
```

Answer: Running time is $O(n \log n)$

Explanation: This is a nested loop. Each time the outer loop runs, it executes 1 instruction in the loop header and 1 inner loop. Each time the inner loop runs, it executes 1 instruction in the loop header and 1 instruction in the body of the loop.

The total number of instruction is $\log n + 1 + \log n + 1 + \dots + \log n + 1 + 1$ (for the last loop check) = $O(n \log n)$

```
7) //not nested
   for (i = 1; i ≤ 2*n; i++)
       num++;
   for (j = 0; j ≤ 2*n; j++)
       num++;
```

Answer: Running time is $O(n)$

Explanation: There are two loops running $4n$ times. Each time the first loop runs, it executes 1 instruction in the loop header and 1 instruction in the body of the loop. Each time the second loop runs, it executes 1 instruction in the loop header and 1 instruction in the body of the loop. The total number of instruction is $2n + 1 + 2n + 1 + 1$ (for the last loop check twice) = $O(n)$

```
8) num = 0;
   for (i=1; i; 2500; i++)
       num++;
```

Answer: Running time is $O(1)$

Explanation: There is a single loop running 2500 times. Each time the loop runs, it executes 1 instruction in the loop header and 1 instruction in the body of the loop. The total number of instruction is $2500 + 2500 + 1$ (for the last loop check) = $O(1)$

```
9) for (i = n-1; i > 0; i--) { MPos = i;
   for (j = 0; j < i; j++) { if (a[j] > a[MPos])
```

```

        MPos = j;
    }
    swap(i, MPos);
}

```

Answer: Running time is $O(n^2)$

Explanation: There is a nested loop running n^2 times. Each time the outer loop runs, it executes 1 instruction in the loop header, 1 inner loop and 1 instruction. Each time the inner loop runs, it executes 1 instruction in the loop header and 1 instruction in the body of inner loop.

The total number of instruction is

$$(n-1)+(n-2)+(n-3)+\dots+2 + (n-1)+1=((1+n-1)*(n-1)/2+1) + (n-1)(\text{total swap}) + (1)(\text{last check}) = O(n^2)$$

```

10) for (i = n; i > 0; i = i/2) {
    for (j = i; j > 0; j = j/2) {
        //constant time operations
    }
}

```

Answer: Running time is $O((\log n)^2)$

Explanation: There is a nested loop running $(\log n)^2$ times. Each time the outer loop runs, it executes 1 instruction in the loop header, 1 inner loop. Each time the inner loop runs, it executes 1 instruction in the loop header and constant time operation in the body of inner loop.

The total number of instruction is

$$\begin{aligned}
 &\log n + \log(n/2) + \log(n/4) + \dots (\log n \text{ times}) \dots + 1(\text{last check}) = \log n + \log n - \log 2 + \log n - \log 4 \dots + 1 \\
 &= \log n * \log n - (\log 1 + \log 2 + \log 4 + \dots + \log 2^{(\log n)}) + 1 = (\log n)^2 - \log 2^{((1+\log n)*\log n/2)} + 1 \\
 &= (\log n)^2 - (1+\log n)*\log n/2 + 1 = O((\log n)^2)
 \end{aligned}$$

