

What is this function's time complexity?

```
function func(n)
```

```
1.  x ← 0;
```

```
2.  for i ← 1 to n do
```

```
3.      for j ← 1 to i do
```

```
4.          x ← x + (i - j);
```

```
      /* end for */
```

```
5.  return(x);
```

Sample “for” loops

```
function func(n)
```

```
1.  x ← 0;
```

```
2.  for i ← 1 to n do
```

```
3.      for j ← 1 to n do
```

```
4.          x ← x + (i - j);
```

```
      /* end for */
```

```
5.  return(x);
```

Sample “for” loops

```
function func(n)
```

```
1.  x ← 0;
```

```
2.  for i ← 1 to n do
```

```
3.      for j ← 1 to i do
```

```
4.          x ← x + (i - j);
```

```
      /* end for */
```

```
5.  return(x);
```

Sample “for” loops

function func(n)

1. $x \leftarrow 0;$
2. **for** $i \leftarrow 1$ **to** n **do**
3. **for** $j \leftarrow i$ **to** n **do**
4. $x \leftarrow x + (i - j);$
- $/*$ end for $*/$
5. **return**(x);

Sample “for” loops

```
function func(n)
```

```
1.  x ← 0;
```

```
2.  for i ← 1 to n do
```

```
3.      for j ← 1 to  $\lfloor \sqrt{n} \rfloor$  do
```

```
4.          x ← x + (i - j);
```

```
      /* end for */
```

```
5.  return(x);
```

Sample “for” loops

function func(n)

1. $x \leftarrow 0;$

2. **for** $i \leftarrow 1$ **to** n **do**

3. **for** $j \leftarrow 1$ **to** $\lfloor \sqrt{i} \rfloor$ **do**

4. $x \leftarrow x + (i - j);$

 /* end for */

5. **return**(x);

Sample “while” loops

function func(n)

1. $x \leftarrow 0;$

2. $i \leftarrow 1;$

3. **while** $i \leq n$ **do**

4. $x \leftarrow x + i;$

5. $i \leftarrow i + 1;$

 /* end while */

6. **return**(x);

Sample “while” loops

```
function func(n)
1.  x ← 0;
2.  i ← 7;
3.  while i ≤ n do
4.      x ← x + i;
5.      i ← i + 1;
    /* end while */
6.  return(x);
```

Sample “while” loops

function func(n)

1. $x \leftarrow 0;$

2. $i \leftarrow 7;$

3. **while** $i \leq n$ **do**

4. $x \leftarrow x + i;$

5. $i \leftarrow i + 3;$

/ end while */*

6. **return**(x);

Sample “while” loops

function func(n)

1. $x \leftarrow 0;$

2. $i \leftarrow 1;$

3. **while** $i \leq n$ **do**

4. $x \leftarrow x + i;$

5. $i \leftarrow i \times 2;$

 /* end while */

6. **return**(x);

Sample “while” loops

```
function func(n)
1.  x ← 0;
2.  i ← 7;
3.  while i ≤ n do
4.      x ← x + i;
5.      i ← i × 2;
    /* end while */
6.  return(x);
```

Sample “while” loops

function func(n)

1. $x \leftarrow 0;$

2. $i \leftarrow 1;$

3. **while** $i \leq n$ **do**

4. $x \leftarrow x + i;$

5. $i \leftarrow i \times 3;$

 /* end while */

6. **return**(x);

Sample “while” loops

function func(n)

1. $x \leftarrow 0;$

2. **for** $i \leftarrow 1$ **to** n **do**

3. $j \leftarrow 1;$

4. **while** $j \leq n$ **do**

5. $x \leftarrow x + (i - j);$

6. $j \leftarrow j \times 2;$

$/*$ end while $*/$

$/*$ end for $*/$

7. **return**(x);

Sample “while” loops

function func(n)

1. $x \leftarrow 0;$

2. **for** $i \leftarrow 1$ **to** n **do**

3. $j \leftarrow 1;$

4. **while** $j \leq i$ **do**

5. $x \leftarrow x + (i - j);$

6. $j \leftarrow j \times 2;$

$/*$ end while $*/$

$/*$ end for $*/$

7. **return**(x);

More sample algorithms

function func(n)

1. $x \leftarrow 0$;

2. **for** $i \leftarrow 1$ **to** n **do**

3. **for** $j \leftarrow 1$ **to** i **do**

4. **for** $k \leftarrow j$ **to** i **do**

5. $x \leftarrow x + (j * k - i)$;

 /* end for */

 /* end for */

 /* end for */

6. **return**(x);

More sample algorithms

function func(n)

1. $x \leftarrow 0$;

2. $i \leftarrow 1$;

3. **while** $i \leq n$ **do**

4. **for** $j \leftarrow 1$ **to** i **do**

5. $x \leftarrow x + (i - j)$;

 /* end for */

6. $i \leftarrow i \times 2$;

 /* end while */

7. **return**(x);

More sample algorithms

```
function func(n)
1.  x ← 0;
2.  for i ← 1 to n do
3.      j ← 0;
4.      while j ≤ n do
5.          x ← x + (i - j);
6.          j ← j + i;
       /* end while */
       /* end for */
7.  return(x);
```

More sample algorithms

function func(n)

1. **if** $n > 100000$ **then return**(0);
2. $x \leftarrow 0$;
3. **for** $i \leftarrow 1$ **to** n **do**
4. **for** $j \leftarrow 1$ **to** n **do**
5. $x \leftarrow x + (i - j)$;
- /* end for */
6. **return**(x);

More sample algorithms

function func(n)

1. **if** $n < 100000$ **then return**(0);
2. $x \leftarrow 0$;
3. **for** $i \leftarrow 1$ **to** n **do**
4. **for** $j \leftarrow 1$ **to** n **do**
5. $x \leftarrow x + (i - j)$;
6. */* end for */*
6. **return**(x);

More sample algorithms

function func(n)

1. **if** n is even **then return**(0);
2. $x \leftarrow 0$;
3. **for** i \leftarrow 1 **to** n **do**
4. **for** j \leftarrow 1 **to** n **do**
5. $x \leftarrow x + (i - j)$;
- /* end for */
6. **return**(x);