COMP108
Data Structures and Algorithms

Algorithm Efficiency (Part II - Exercises)

Professor Prudence Wong

pwong@liverpool.ac.uk

2020-21

Exercises

Determine the order of growth of the following functions.

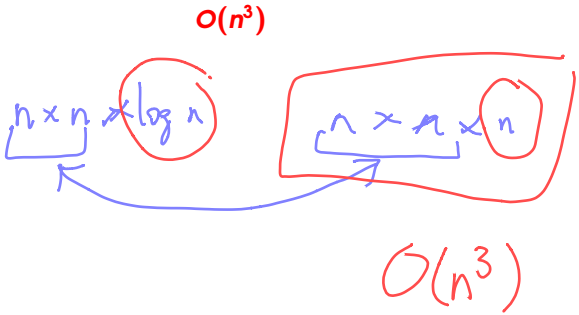
1. $n^3 + 3n^2 + 3$ $O(n^3)$ n^3, n^2, constant
2. $4n^2 \log n + n^3 + 5n^2 + n$
3. $2n^2 + n^2 \log n$
4. $6n^2 + 2^n$

Look for the term highest in the hierarchy

Exercises

Determine the order of growth of the following functions.

1. $n^3 + 3n^2 + 3$
2. $4n^2 \log n + n^3 + 5n^2 + n$
3. $2n^2 + n^2 \log n$
4. $6n^2 + 2^n$



Look for the term highest in the hierarchy

Exercises

Determine the order of growth of the following functions.

1. $n^3 + 3n^2 + 3$ $O(n^3)$
2. $4n^2 \log n + n^3 + 5n^2 + n$ $O(n^3)$
3. $2n^2 + n^2 \log n$
4. $6n^2 + 2^n$

Look for the term highest in the hierarchy

Exercises

Determine the order of growth of the following functions.

1. $n^3 + 3n^2 + 3$ — $O(n^3)$
2. $4n^2 \log n + n^3 + 5n^2 + n$ — $O(n^3)$
3. $2n^2 + n^2 \log n$ — $O(n^2 \log n)$
4. $6n^2 + 2^n$ — exponential, $O(2^n)$

polynomial

Look for the term highest in the hierarchy

Exercises

Determine the order of growth of the following functions.

1. $n^3 + 3n^2 + 3$ — $O(n^3)$
2. $4n^2 \log n + n^3 + 5n^2 + n$ — $O(n^3)$
3. $2n^2 + n^2 \log n$ — $O(n^2 \log n)$
4. $6n^2 + 2^n$ — $O(2^n)$

Look for the term highest in the hierarchy

More exercises

Are the following correct?

1. $n^2 \log n + n^3 + 3n^2 + 3$ — $O(n^2 \log n)$? ✗ $o(n^3)$
2. $n + 1000$ — $O(n)$?
3. $6n^{20} + 2^n$ — $O(n^{20})$?
4. $n^3 + 5n^2 \log n + n$ — $O(n^2 \log n)$?

More exercises

Are the following correct?

1. $n^2 \log n + n^3 + 3n^2 + 3$ $O(n^2 \log n)$? **$O(n^3)$**
2. $n + 1000$ $O(n)$?
3. $6n^{20} + 2^n$ $O(n^{20})$?
4. $n^3 + 5n^2 \log n + n$ $O(n^2 \log n)$?

More exercises

Are the following correct?

1. $n^2 \log n + n^3 + 3n^2 + 3$	$O(n^2 \log n)$?	$O(n^3)$
2. $n + 1000$	$O(n)$?	YES
3. $6n^{20} + 2^n$	$O(n^{20})$?	
4. $n^3 + 5n^2 \log n + n$	$O(n^2 \log n)$?	

More exercises

Are the following correct?

1. $n^2 \log n + n^3 + 3n^2 + 3$	$O(n^2 \log n)$?	**$O(n^3)$**
2. $n + 1000$	$O(n)$?	**YES**
3. $6n^{20} + 2^n$	$O(n^{20})$?	**$O(2^n)$**
4. $n^3 + 5n^2 \log n + n$	$O(n^2 \log n)$?	

More exercises

Are the following correct?

1. $n^2 \log n + n^3 + 3n^2 + 3$	$O(n^2 \log n)$?	**$O(n^3)$**
2. $n + 1000$	$O(n)$?	**YES**
3. $6n^{20} + 2^n$	$O(n^{20})$?	**$O(2^n)$**
4. $n^3 + 5n^2 \log n + n$	$O(n^2 \log n)$?	**$O(n^3)$**

Time complexity of algorithm

```
sum ← 0, i ← 1
while i ≤ n do
begin
   sum ← sum + i
   i ← i + 1
end
output sum
```

$O(?)$

Time complexity of algorithm

```
sum ← 0, i ← 1
while i ≤ n do
begin
   sum ← sum + i
   i ← i + 1
end
output sum
```

$O(?)$

$\boldsymbol{O(n)}$

Time complexity of algorithm

```
sum ← 0, i ← 1
while i ≤ n do
begin
   sum ← sum + i
   i ← i + 1
end
output sum
```

$O(?)$

$\boldsymbol{O(n)}$

```
sum ← n * (n + 1)/2
output sum
```

$O(?)$

Time complexity of algorithm

```
sum ← 0, i ← 1
while i ≤ n do
begin
   sum ← sum + i
   i ← i + 1
end
output sum
```

$O(?)$

$O(n)$

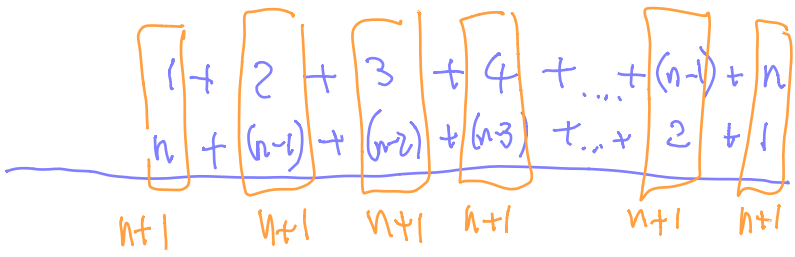
```
sum ← n * (n + 1)/2
output sum
```

$O(?)$

$O(1)$

$$\frac{2 \times Sum}{2} = \frac{n \times (n+1)}{2}$$

$$Sum = \frac{n \times (n+1)}{2}$$

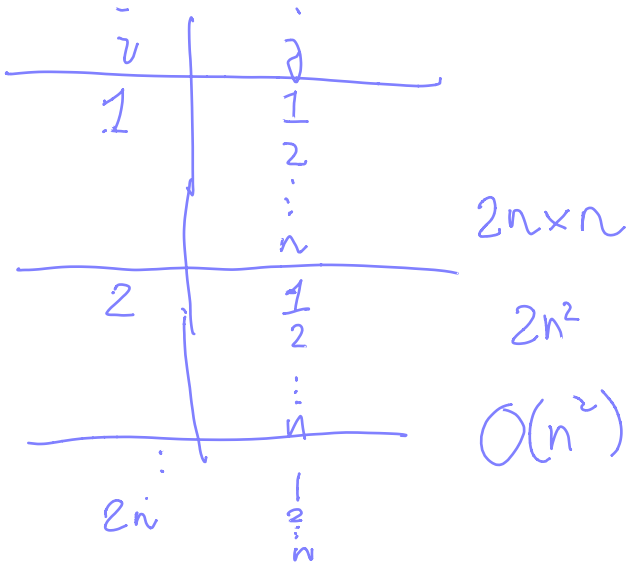


Time complexity of algorithm (2)

nested loop

```
i ← 1
while i ≤ 2n do
begin
   j ← 1
   while j ≤ n do
      j ← j + 1
   i ← i + 1
end
```

O(?)



Time complexity of algorithm (2)

```
i ← 1
while i ≤ 2n do
begin
   j ← 1
   while j ≤ n do
      j ← j + 1
   i ← i + 1
end
```

$O(?)$

$\boldsymbol{O(n^2)}$

Time complexity of algorithm (2)

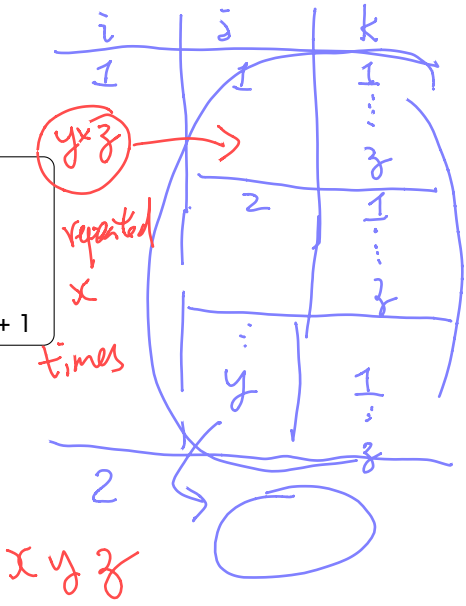
```
i ← 1
while i ≤ 2n do
begin
   j ← 1
   while j ≤ n do
      j ← j + 1
   i ← i + 1
end
```

$O(?)$

$\boldsymbol{O(n^2)}$

```
count ← 0
for i ← 1 to x do
   for j ← 1 to y do
      for k ← 1 to z do
         count ← count + 1
```

$O(?)$



Time complexity of algorithm (2)

```
i ← 1
while i ≤ 2n do
begin
   j ← 1
   while j ≤ n do
      j ← j + 1
   i ← i + 1
end
```

$O(?)$

$\boldsymbol{O(n^2)}$

```
count ← 0
for i ← 1 to x do
   for j ← 1 to y do
      for k ← 1 to z do
         count ← count + 1
```

$O(?)$

$\boldsymbol{O(xyz)}$

Time complexity of algorithm (2)

```
i ← 1
while i ≤ 2n do
begin
   j ← 1
   while j ≤ n do
      j ← j + 1
   i ← i + 1
end
```

O(?)

O(n^2)

$n+n+n+\ldots+n$

```
count ← 0
for i ← 1 to x do
   for j ← 1 to y do
      for k ← 1 to z do
         count ← count + 1
```

O(?)

O(xyz)

$\frac{n^2}{2}+\frac{n}{2} \leftarrow \frac{n\times(n+1)}{2}$

$O(n^2)$

```
i ← 1
while i ≤ n do
begin
   j ← i
   while j ≤ n do
      j ← j + 1
   i ← i + 1
end
```

i	j
1	1 … n
2	2 … n

O(?)

$n+(n-1)+(n-2)+\ldots+1$

Time complexity of algorithm (2)

```
i ← 1
while i ≤ 2n do
begin
   j ← 1
   while j ≤ n do
      j ← j + 1
   i ← i + 1
end
```

$O(?)$

$\boldsymbol{O(n^2)}$

```
count ← 0
for i ← 1 to x do
   for j ← 1 to y do
      for k ← 1 to z do
         count ← count + 1
```

$O(?)$

$\boldsymbol{O(xyz)}$

```
i ← 1
while i ≤ n do
begin
   j ← i
   while j ≤ n do
      j ← j + 1
   i ← i + 1
end
```

$O(?)$

$\boldsymbol{O(n^2)}$

Time complexity of algorithm (3)

```
i ← 1
count ← 0
while i < n do
begin
   i ← i * 2
   count ← count + 1
end
output count
```

$O(?)$

Time complexity of algorithm (3)

```
i ← 1
count ← 0
while i < n do
begin
   i ← i * 2
   count ← count + 1
end
output count
```

$O(?)$

- suppose n = 8

iteration	i before	i after	count
before loop		1	0
1	1	2	1
2	2	4	2
3	4	8	**3**

Time complexity of algorithm (3)

```
i ← 1
count ← 0
while i < n do
begin
   i ← i * 2
   count ← count + 1
end
output count
```

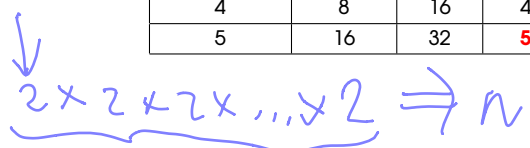
$O(?)$

- suppose n = 8

iteration	i before	i after	count
before loop		1	0
1	1	2	1
2	2	4	2
3	4	8	**3**

- suppose n = 32

iteration	i before	i after	count
before loop		1	0
1	1	2	1
2	2	4	2
3	4	8	3
4	8	16	4
5	16	32	**5**



Time complexity of algorithm (3)

```
i ← 1
count ← 0
while i < n do
begin
   i ← i * 2
   count ← count + 1
end
output count
```

$O(?)$ **$O(\log n)$**

- suppose n = 8

iteration	i before	i after	count
before loop		1	0
1	1	2	1
2	2	4	2
3	4	8	**3**

- suppose n = 32

iteration	i before	i after	count
before loop		1	0
1	1	2	1
2	2	4	2
3	4	8	3
4	8	16	4
5	16	32	**5**

Summary: Measuring algorithm efficiency

Next: Data structures - arrays

For note taking