

Time complexity notations and master theorem

Due No due date

Points 5

Questions 5

Time limit None

Allowed attempts Unlimited

[Take the quiz again](#)

Attempt history

	Attempt	Time	Score
KEPT	Attempt 2	3 minutes	5 out of 5
LATEST	Attempt 2	3 minutes	5 out of 5
	Attempt 1	27 minutes	2 out of 5

Submitted 19 Apr at 22:56

Question 1

1 / 1 pts

What is in $o(n^2)$? (multiple answers)

Correct!

☒ 1

$$\frac{1}{n^2} \rightarrow 0 \text{ as } n \rightarrow \infty$$

Correct!

☒ $n \log(n)$

$$\frac{\log(n) n}{n^2} = \frac{\log(n)}{n} \rightarrow 0 \text{ as } n \rightarrow \infty$$

Correct!

☒ $\text{sqrt}(n)$

$$\frac{\sqrt{n}}{n^2} = \frac{1}{\sqrt{n} n} \rightarrow 0 \text{ as } n \rightarrow \infty$$

☐ $\log(n) n^2$

☐ n^3

☐ n^2

Correct!

☒ n

$$\frac{n}{n^2} = \frac{1}{n} \rightarrow 0 \text{ as } n \rightarrow \infty$$

Question 2

1 / 1 pts

Which of the following is true? (multiple answers)

Correct!

☒ if $f(n)=o(n)$ and $g(n)=o(n)$ then $f(n)+g(n)=o(n)$

$$\frac{f(n)}{n} \rightarrow 0 \text{ as } n \rightarrow \infty \text{ and } \frac{g(n)}{n} \rightarrow 0 \text{ as } n \rightarrow \infty, \text{ hence } \frac{f(n)+g(n)}{n} \rightarrow 0 \text{ as } n \rightarrow \infty$$

☐ $O(n)$ is the complexity in the worst case

☐ if $f(n)=\Omega(g(n))$ and $f(n)=O(h(n))$, then $g(n)\leq h(n)$ for all $n>0$

☐ if $f(n)=O(n)$ then $f(n)=o(n)$

Correct!

☒ if $f(n)=\Theta(g(n))$ then $g(n)=\Theta(f(n))$

Yes, by the definition of Θ .

Question 3

1 / 1 pts

Prove that functions $T_1(n) = T_1(n/2) + n$, and $T_2(n) = 2T_2(n/2) + 2$ are of the same order.

Select this order.

Correct!

☒ n

☐ $\log(n)$

☐ $n \log(n)$

☐ n^2

☐ 1

Use master theorem.

Question 4

1 / 1 pts

We considered Binary search which halves our sorted arrays and then recursively searches in one of the halves.

Now imagine we want instead split our sorted array into 3 parts and then continue searching in one of 3 parts. Let's call this recursive algorithm Ternary search.

What is the computational complexity of Ternary search?

☐ $\Theta(n^3)$

☐ $\Theta(n)$

☐ $\Theta(n \log n)$

☐ $\Theta(1)$

Correct!

☒ $\Theta(\log n)$

Question 5

1 / 1 pts

What is the time complexity of this function?

// Input: array and its size.

```
int f (vector<int> inputArray, int n) {
```

```
    if (n < 2) return 1;
```

// resize n/2 leaves only n/2 first elements.

```
    return f(inputArray.resize(n/2), n/2) + 7*n + n*n + 11;
}
```

Correct!

☒ $\Theta(\log(n))$

☐ $\Theta(n^2)$

☐ $\Theta(n)$

☐ $\Theta(n \log(n))$

Apply master theorem. $a=1$, $b=2$, $d=0$ (because of the constant number of operations in each function call).