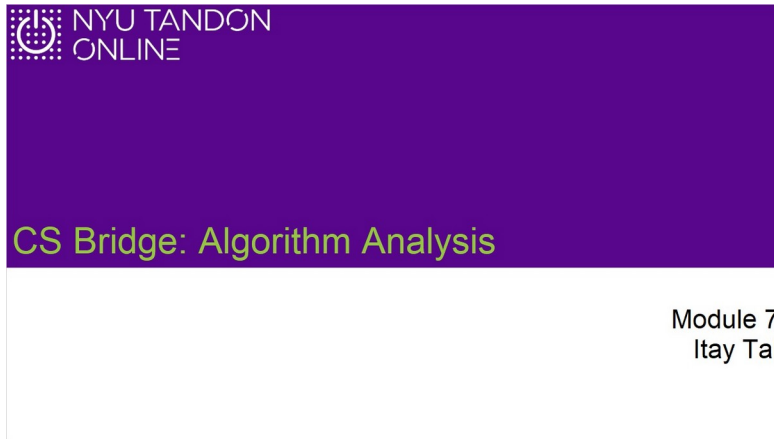


CS Bridge Module 7 Algorithm Analysis

1. Module 7 Algorithm Analysis

1.1 Title Slide



2. The Primality Testing Problem

2.1 Primality Testing

A slide titled "Primality Testing" with a dark background. The title is in light green. The NYU Tandon Online logo is in the top right. The slide contains two definitions and examples of primality testing. A presenter, a man with glasses wearing a brown shirt, is visible on the right side of the slide.

Primality Testing

Definition: Let $num \geq 2$ be an integer. We say that num is *prime*, if its only divisors are 1 and num

Examples:

13 is prime 12 is not prime

Definition: Let $num \geq 2$ be an integer, and let d and k be two divisors of num . We say that k and d are *complementary divisors* of num , if $d * k = num$

Examples:


4 and 25 are complementary divisors of 100
5 and 20 are complementary divisors of 100


2.2 Primality Testing: Solution 1

Primality Testing: Solution 1

```
bool isPrime(int num){
    int countDivs;
    countDivs = 0;
    for(int i=1; i <= num; i++){
        if(num % i == 0)
            countDivs++;
    }

    if(countDivs == 2)
        return true;
    else
        return false;
}
```






2.3 Primality Testing: Solution 2


Primality Testing: Solution 2

```
bool isPrime(int num){
    int countDivs;

    countDivs = 0;
    for(int i=1; i <= num; i++){
        if(num % i == 0)
            countDivs++;
    }

    if(countDivs == 2)
        return true;
    else
        return false;
}
```





Let k be a divisor of num in the second half of the range num .
 That is, $k \in \frac{num}{2}, \dots, num$.
 Let d be k 's complementary divisor, therefore $d = \frac{num}{k}$.
 We have: $d = \frac{num}{k} \leq \frac{num}{\frac{num}{2}} = 2$.
 Therefore, $d = 1$. Since d is a divisor, we get that $d = 1$.
 So: $\frac{num}{k} = 1$, therefore $k = num$.
 This shows that the only divisor in the second half of the range is num itself.


2.4 Primality Testing: Solution 3


Primality Testing: Solution 3

```
bool isPrime(int num){
    int countDivs;

    countDivs = 0;
    for(int i=1; i <= num; i++){
        if(num % i == 0)
            countDivs++;
    }

    if(countDivs == 2)
        return true;
    else
        return false;
}
```





Let k and d be complementary divisors of num ,
 and assume that they are both greater than \sqrt{num} .
 Version III: $num = k \cdot d > \sqrt{num} \cdot \sqrt{num} = num$.
 We therefore have: $num > num$, which is a contradiction.
 This implies that $num > num$, which is a contradiction.
 So: $num = 100$: 1 2 4 5 10 20 25 50 100
 This shows that at least one in each pair of complementary divisors is less than or equal to \sqrt{num} .

2.5 Runtime Analysis Part 1


Runtime Analysis

The running time depends on the size of the input. We compare the number of primitive operations executed by a process, as a function of its input size.

The running time depends on the operators we use, and on the types of the data they are applied on.

Ignore machine-dependent constants. We count each primitive operation as 1.

Let n be the size of the input ($n = \text{num}$)




Version 1.2.3

$T_1(n) = 5n + 4$

$T_2(n) = 3n + 4$

$T_3(n) = 6\sqrt{n} + 4$



2.6 Runtime Analysis Part 2

Runtime Analysis


```
bool isPrime(int num){
    int countDivs;
    countDivs = 0;
    for(int i=1; i <= num; i++){
        if(num % i == 0)
            countDivs++;
    }
    if(countDivs == 2)
        return true;
    else
        return false;
}
```

Let n be the size of the input ($n = \text{num}$)

$T_1(n) = 5n + 4$

$T_2(n) = 3n + 4$

$T_3(n) = 6\sqrt{n} + 4$




Version 1.2.3

$T_1(n) = 5n + 4$

$T_2(n) = 3n + 4$

$T_3(n) = 6\sqrt{n} + 4$



2.7 Runtime Analysis Part 3

Runtime Analysis

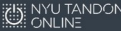
The running time depends on the size of the input. We compare the number of primitive operations executed by a process, as a function of its input size.

The running time depends on the operators we use, and on the types of the data they are applied on.

Ignore machine-dependent constants. We count each primitive operation as 1.

Rule of thumb to get the order of growth: Drop low-order terms and ignore leading constants. Look at the order of growth of $T(n)$.

$T(n) = 3n^2 + 6n + 15 = O(n^2)$

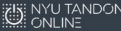


Version 1.2.3

$T_1(n) = 5n + 4$

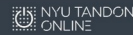
$T_2(n) = 3n + 4$

$T_3(n) = 6\sqrt{n} + 4$



2.8 Runtime Analysis Part 4

Runtime Analysis



Let n be the size of the input ($n = \text{num}$)

$$T_1(n) = 5n + 4 = \theta(n) \rightarrow T_1(n) = \theta(n)$$

$$T_2(n) = 3n + 4 = \theta(n) \rightarrow T_2(n) = \theta(n)$$

$$T_3(n) = 6\sqrt{n} + 4 = \theta(\sqrt{n}) \rightarrow T_3(n) = \theta(\sqrt{n})$$

Conclusions:

- $T_1(n)$ and $T_2(n)$ are asymptotically equivalent
- $T_3(n)$ is asymptotically better than $T_1(n)$ and $T_2(n)$

3. Asymptotic Analysis

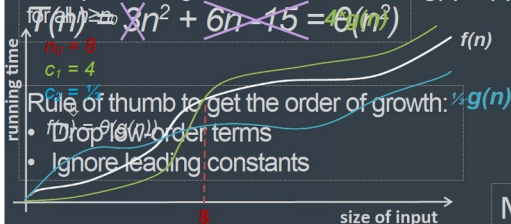
3.1 Order of Growth: Formal Definition

Order of Growth: Formal Definition



Definition
We compare the asymptotic order of two functions $f(n)$ and $g(n)$ by mapping positive integers to positive real numbers. We say that $f(n) = \theta(g(n))$ if there exist positive real constants c_1, c_2 and a positive integer constant n_0 such that $c_2 \cdot g(n) \leq f(n) \leq c_1 \cdot g(n)$.

Example: $f(n) = 3n^2 + 6n + 15 = \theta(n^2)$



More Formally ...

3.2 Asymptotic Analysis Example

Asymptotic Analysis Example NYU TANDON ONLINE

Show that: $3n^2 + 6n - 15 = \theta(n^2)$

Proof: $f(n) \quad g(n)$

if we take $c_1 = 9$
 $c_2 = 3$
 $n_0 = 3$

Then for all $n \geq n_0$ we have:

$6n - 15 \geq 0$
 \Downarrow
 $6n \geq 15$
 \Downarrow
 $n \geq 2.5$

$3n^2 \leq 3n^2 + 6n - 15 \leq 3n^2 + 6n \leq 3n^2 + 6n^2 = 9n^2$
 \Downarrow
 $3n^2 \leq 3n^2 + 6n - 15 \leq 9n^2$

Therefore: $3n^2 + 6n - 15 = \theta(n^2)$

Notes:

4. Runtime Analysis

4.1 Runtime Analysis: Example 1

Runtime Analysis: Example 1 NYU TANDON ONLINE

```
int main() {
    int n, i, j;
    cout<<"Please enter n:"<<endl;
    cin>>n;
    for(i = 1; i <= n; i++){
        for(j = 1; j <= n; j++){
            cout<<' * ';
            cout<<endl;
        }
    }
    return 0;
}
```

$n \cdot n$

$T(n)$

* * * *


* * * *

* * * *

* * * *

4.2 Runtime Analysis: Example 2

Runtime Analysis: Example 2




```
int main() {
    int n, i, j;
    cout<<"Please enter n:"<<endl;
    cin>>n;
    for(i = 1; i <= n; i++){
        for(j = 1; j <= i; j++)
            cout<<"*";
        cout<<endl;
    }
    return 0;
}
```

$T(n) = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2} = \frac{1}{2} \cdot n^2 + \frac{1}{2} \cdot n = \theta(n^2)$

\Downarrow


$T(n) = \theta(n^2)$



4.3 Knowledge Check

(Fill-in-the-Blank, 10 points, unlimited attempts permitted)

Knowledge Check



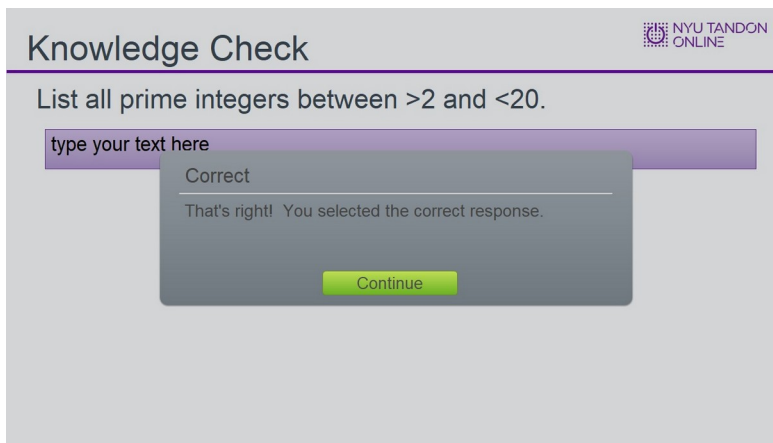
List all prime integers between >2 and <20.

Choice
3, 5, 7, 11, 13, 17, 19
3,5,7,11,13,17,19
3 5 7 9 11 13 17 19

Feedback when correct:

That's right! You selected the correct response.

Correct (Slide Layer)



The interface shows a 'Knowledge Check' slide with the question 'List all prime integers between >2 and <20.' Below the question is a text input field with the placeholder 'type your text here'. A grey feedback box is overlaid on the input field, displaying the word 'Correct' in bold, followed by the message 'That's right! You selected the correct response.' and a green 'Continue' button.

Knowledge Check

NYU TANDON ONLINE

List all prime integers between >2 and <20.

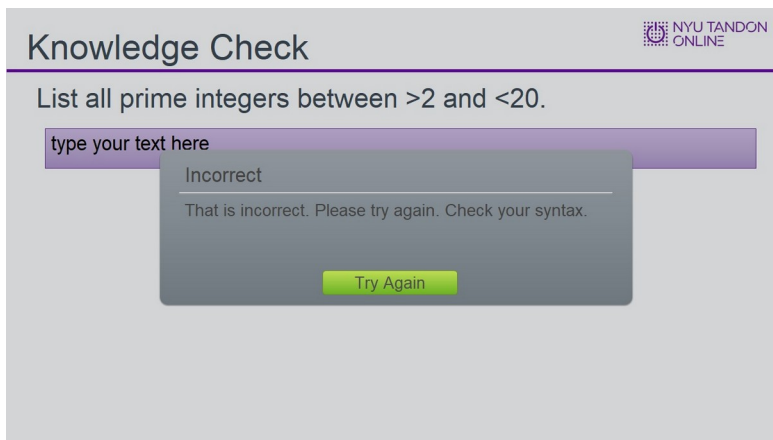
type your text here

Correct

That's right! You selected the correct response.

Continue

Try Again (Slide Layer)



The interface shows the same 'Knowledge Check' slide with the question 'List all prime integers between >2 and <20.' Below the question is a text input field with the placeholder 'type your text here'. A grey feedback box is overlaid on the input field, displaying the word 'Incorrect' in bold, followed by the message 'That is incorrect. Please try again. Check your syntax.' and a green 'Try Again' button.

Knowledge Check

NYU TANDON ONLINE

List all prime integers between >2 and <20.

type your text here

Incorrect

That is incorrect. Please try again. Check your syntax.

Try Again

4.4 Knowledge Check

(Multiple Choice, 10 points, 4 attempts permitted)

Knowledge Check

The output of an algorithm $T(n)$ can be described by the following equation, where the input is n . What is this algorithm's running time?

$$T(n) = 5n^2 + 2n - 3$$

- ☐ $O(n)$
- ☐ $O(5n^2)$
- ☒ $O(n^2)$
- ☐ $O(5n^2 + 2n)$

Correct	Choice
	$O(n)$
	$O(5n^2)$
X	$O(n^2)$
	$O(5n^2 + 2n)$

Feedback when correct:

That's right! You selected the correct response.

Correct (Slide Layer)

Knowledge Check

The output of an algorithm $T(n)$ can be described by the following equation, where the input is n . What is this algorithm's running time?

$$T(n) = 5n^2 + 2n - 3$$

- ☐ $O(n)$
- ☐ $O(5n^2)$
- ☒ $O(n^2)$
- ☐ $O(5n^2 + 2n)$

Correct

That's right! You selected the correct response.

Continue

Try Again (Slide Layer)

Knowledge Check

NYU TANDON
ONLINE

The output of an algorithm $T(n)$ can be described by the following equation, where the input is n . What is this algorithm's running time?
 $T(n) = 5n^2 + 2n - 3$

Incorrect

That is incorrect. Please try again.

Try Again

☐ $O(n)$

☐ $O(5n^2)$

☒ $O(n^2)$

☐ $O(5n^2 + 2n)$

4.5 Results Slide

(Results Slide, 0 points, 1 attempt permitted)

Results

NYU TANDON
ONLINE

Your Score: %Results.ScorePercent%% (%Results.ScorePoints% points)

Passing Score: %Results.PassPercent%% (%Results.PassPoints% points)

Result:

Retry Quiz

Review Quiz

Results for
4.3 Knowledge Check
4.4 Knowledge Check

Result slide properties


Passing 80%

Score

Notes:

Success (Slide Layer)

Results




Your Score:

%Results.ScorePercent%% (%Results.ScorePoints% points)

Passing Score:

%Results.PassPercent%% (%Results.PassPoints% points)

Result:

 Congratulations, you passed.

Retry Quiz

Review Quiz

Failure (Slide Layer)

Results

NYU TANDON
ONLINE

Your Score:

%Results.ScorePercent%% (%Results.ScorePoints% points)

Passing Score:

%Results.PassPercent%% (%Results.PassPoints% points)

Result:

✖ You did not pass.

Retry Quiz

Review Quiz

4.6 End of Module

NYU TANDON
ONLINE

End of Module

Exit