

CompSci 161

Spring 2021 Lecture 2:

Algorithm Time Complexity

2 Time to run if input doubled?

Time	Change
n	$2\times$ (twice as long)
$20n$	$2\times$
$10n + 37$	$2\times$
n^2	$4\times$
n^4	$16\times$
$n^5 + 10n^3 + 21$	$32\times$

$$\frac{20(2n)}{20n} = 2$$

$$\frac{2n}{n} = 2$$

$$\frac{10(2n) + 37}{10n + 37} \rightarrow 2$$

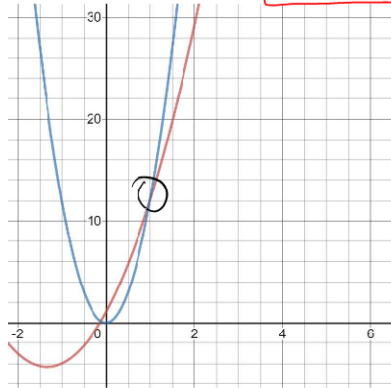
$$\frac{(2n)^2}{n^2} = \frac{4n^2}{n^2}$$

$$\frac{(2n)^4}{n^4} = \frac{16n^4}{n^4} = 16$$

3

\mathcal{O} Notation

- ▶ $f(n)$ is $\mathcal{O}(g(n))$ if $\exists c > 0$ and integer $n_0 \geq 1$ such that for all $n > n_0$, $f(n) \leq cg(n)$.
- ▶ Show that $3n^2 + 12n + 1$ is $\mathcal{O}(n^2)$



4

Running time of linear search

```
int linSearch(const std::vector<int> & numbers,
              int target)
{
    int i;
    int n = numbers.size();
    for(i=0; i < n; i++)
        if( numbers[i] == target )
            return i;
    throw ElementNotFoundException("msg");
}
```

$\mathcal{O}(n)$

constant
time to
eval

4

Running time of linear search

```
int linSearch(const std::vector<int> & numbers,
             int target)
{
    int i;
    int n = numbers.size();
    for(i=0; i < n; i++)
        if( numbers[i] == target )
            return i;
    throw ElementNotFoundException("msg");
}
```

upper bound
still linear

- Does this time change if the vector is sorted?

5

Running time of binary search

```
int binarySearch(const std::vector<int> & numbers,
                int target){
    return binarySearch(numbers, target, 0,
                        numbers.size() - 1);
}

int binarySearch(const std::vector<int> & numbers,
                int target, int low, int high)
{
    if( low > high )
        throw ElementNotFoundException("msg");
    int mid = (low + high) / 2;
    if( numbers[mid] == target )
        return mid;
    else if (target < numbers[mid])
        return binarySearch(numbers, target, low, mid-1);
    else
        return binarySearch(numbers, target, mid+1, high);
}
```

6 What is binary search doing?

↓

2	4	5	7	8	9	12	14	17	19	22	25	27	28	33
---	---	---	---	---	---	----	----	----	----	----	----	----	----	----

Select int $\in [1, 1000]$

500? [501, 1000] 648? [649, 655]

$\frac{501+749}{2} \rightarrow$ 750? [501, 749] 652? [649, 651]

625? [626, 749] 650?

688? [626, 687] 651 ☺

656? [626, 655]

640? [641, 655]

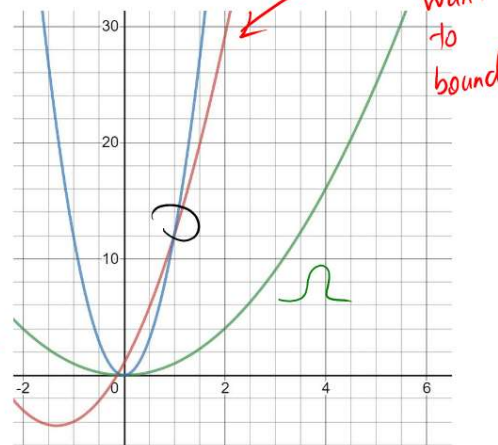
7 Ω and Θ

► \mathcal{O} : upper bound on growth rate of function.

► What are Ω and Θ used to describe?

Ω : lower bound
on growth rate
of function

Shows red is $\Omega(n^2)$



Red: $y = 3n^2 + 8n + 1$

Logarithms

- ▶ $f(n) = \log_{10} n$ and $g(n) = \log_2 n$.

How do they relate?

$$\log_2 n = C \cdot \log_{10} n$$

for some c

- ▶ $f(n) = \log n$. What base do I mean?

$$\log_2 n$$