Dynamic programming 101

Tobias Nyholm

@tobiasnyholm

Why?

Tobias Nyholm

- Full stack unicorn on Happyr.com
- Certified Symfony developer
- Symfony core member
- PHP-Stockholm
- Open source

Open source

Assert

Neo4j

FriendsOfApi/boilerplate

Guzzle

Buzz

Backup-manager/symfony

nyholm/effective-interest-rate

Puli

Swap

LinkedIn API client

NSA

php-http/discovery

PHP-cache

PHP-Translation

PSR7

CacheBundle

league/geotools

Mailgun

KNP Github API

happyr/normal-distribution-bundle

HTTPlug

SymfonyBundleTest

Stampie

php-http/httplug-bundle

MailgunBundle

php-http/multipart-stream

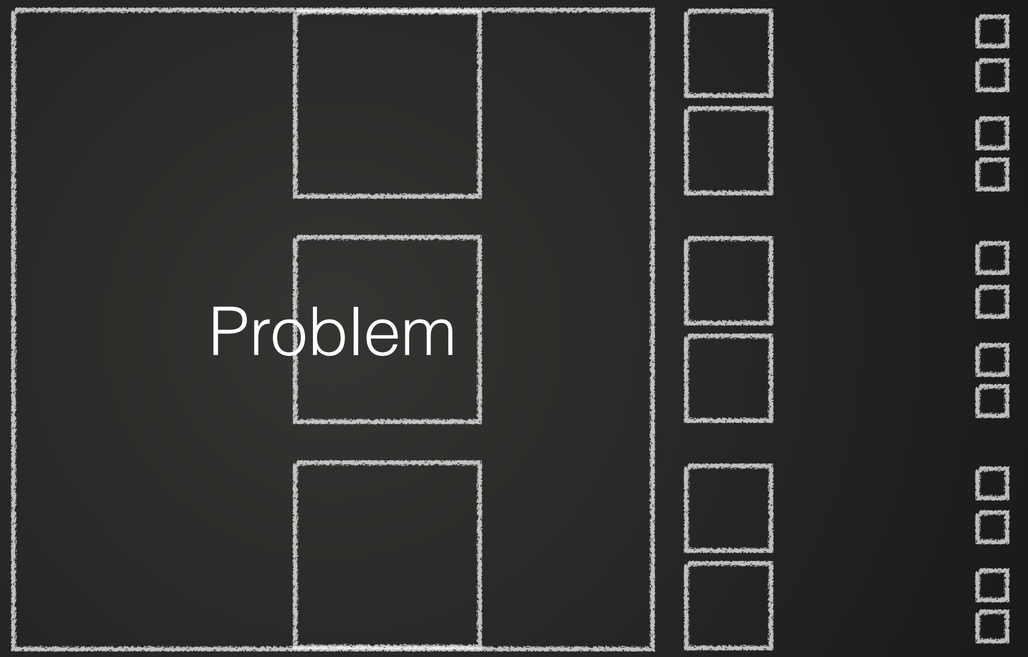
SimpleBus integrations

PHP-Geocoder

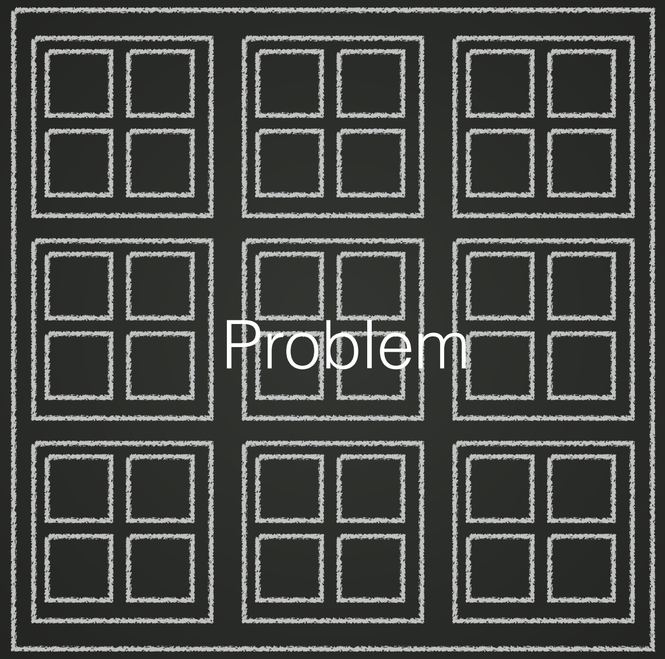
BazingaGeocoderBundle

PSR HTTP clients

You’ve got a problem



Divide and Conquer



Recursion

Iterative function

```php
<?php

function factorial($n)
{
    $result = 1;
    for ($i = 1; $i <= $n; $i++) {
        $result = $result * $i;
    }

    return $result;
}
```

Recursive function

```php
<?php

function factorial($n)
{
    if ($n <= 1) {
        return 1;
    }

    return $n * factorial($n-1);
}
```

Fibonacci

Fibonacci

	0+1=1		2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597
n:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

```php
<?php

function fib($n)
{
    if ($n <= 1) {
        return $n;
    }

    return fib($n-1) + fib($n-2);
}
```

Fibonacci

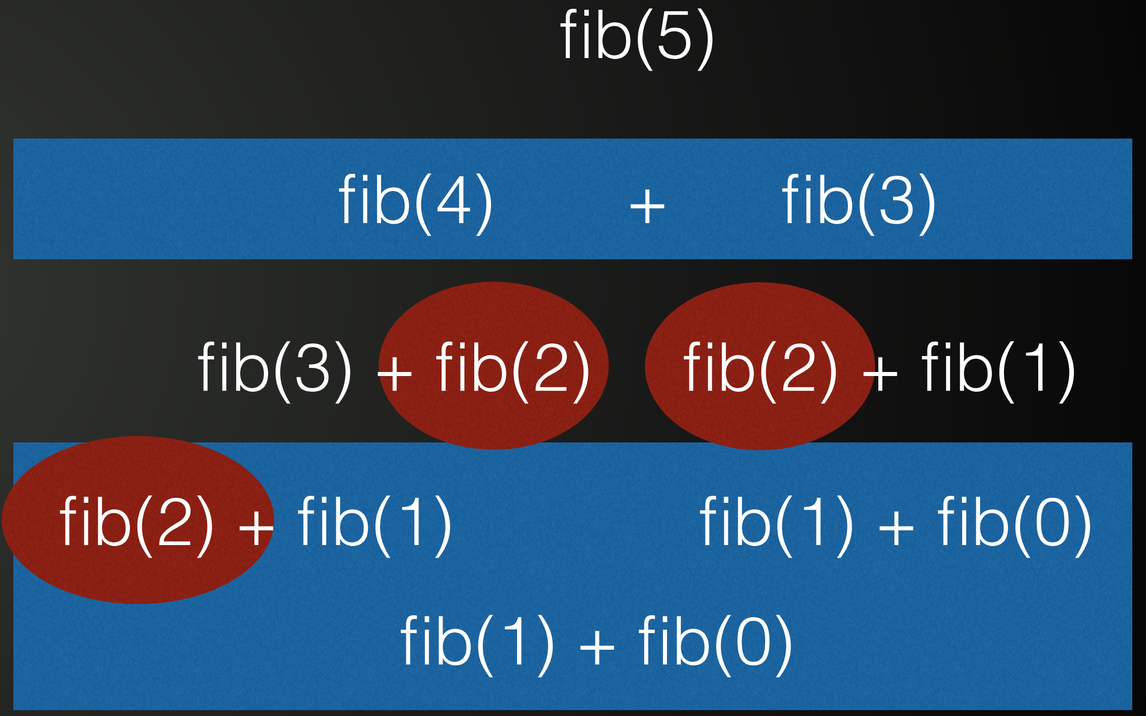
	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597
n:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

$$f(n) = f(n-2) + f(n-1)$$

$$f(1000) = x$$
$$f(1001) = y$$
$$f(1002) = ?$$

Fibonacci

```php
<?php

function fib($n)
{
    if ($n <= 1) {
        return $n;
    }

    return fib($n-1) + fib($n-2);
}
```



Fibonacci

```php
<?php

class Calculator
{
    private $cache = [0 => 0, 1 => 1];

    public function fib($n)
    {
        if (!isset($this->cache[$n])) {
            $this->cache[$n] = fib($n - 1) + fib($n - 2);
        }

        return $this->cache[$n];
    }
}
```

Dynamic programming

Divide the problem into overlapping subproblems
to reduce the number of calculations

Fibonacci

fib(5)

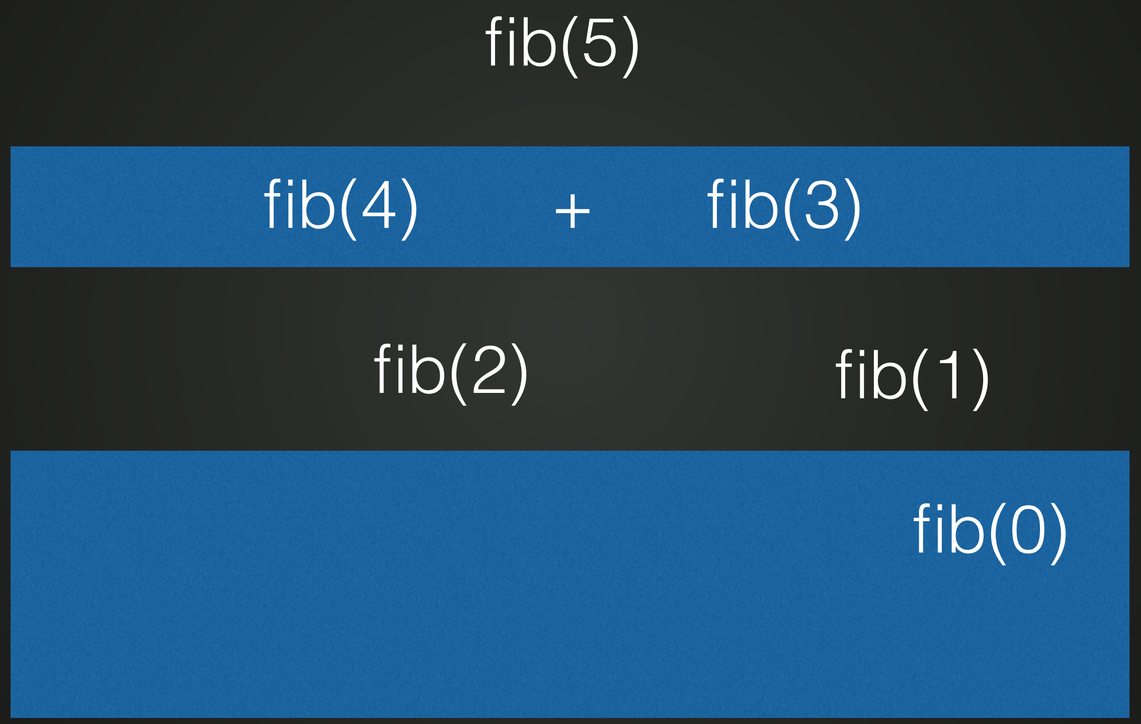
fib(4) + fib(3)

fib(3) + fib(2) fib(2) + fib(1)

fib(2) + fib(1) fib(1) + fib(0)

fib(1) + fib(0)

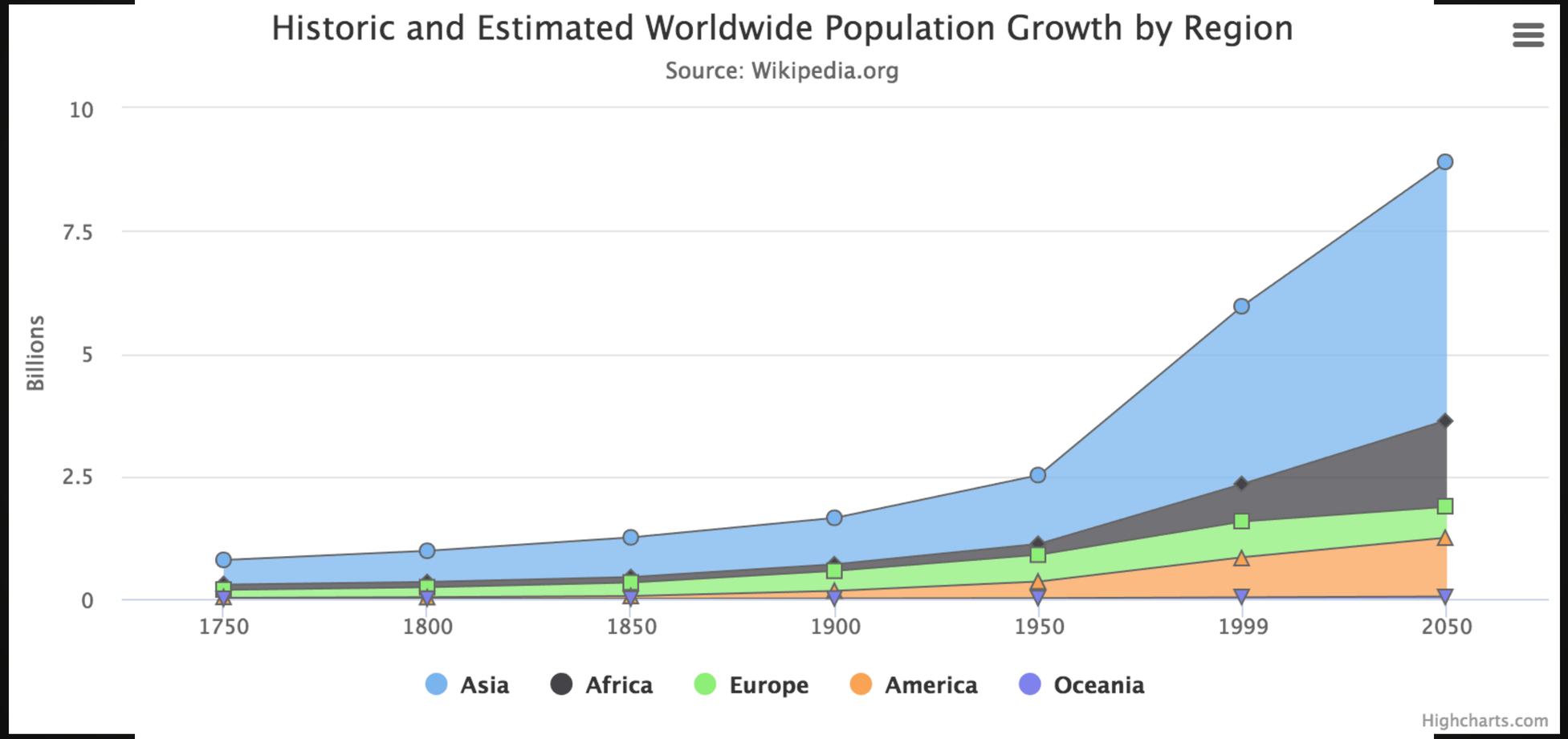
Fibonacci



The more fun stuff

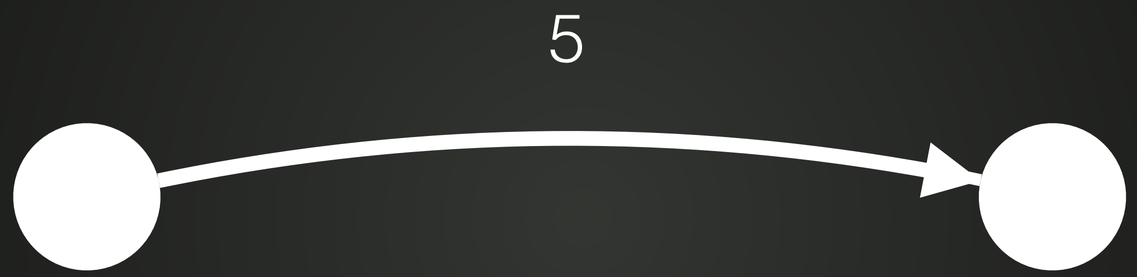
Graph theory

Graph theory



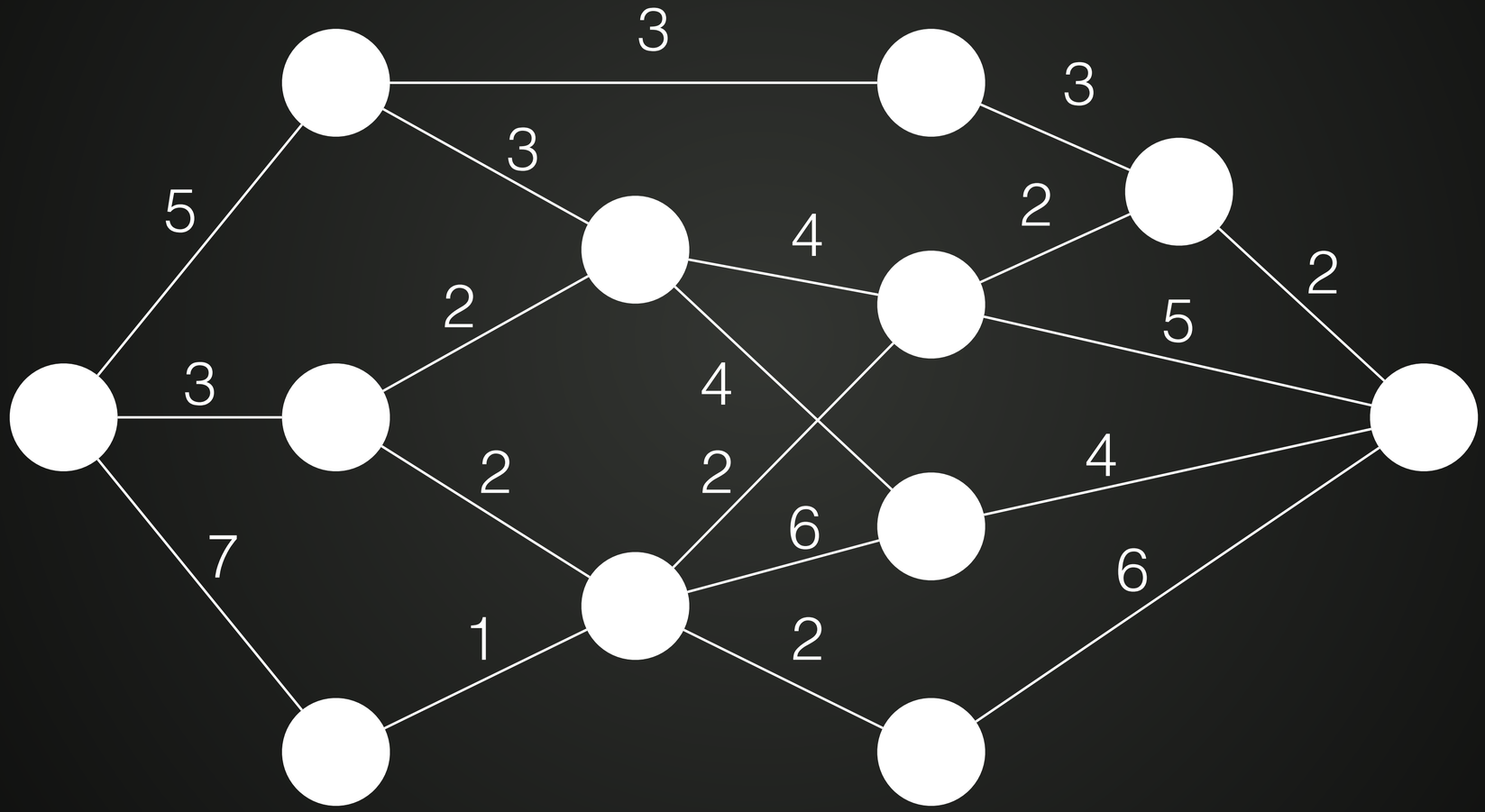
Highcharts.com

Graph theory

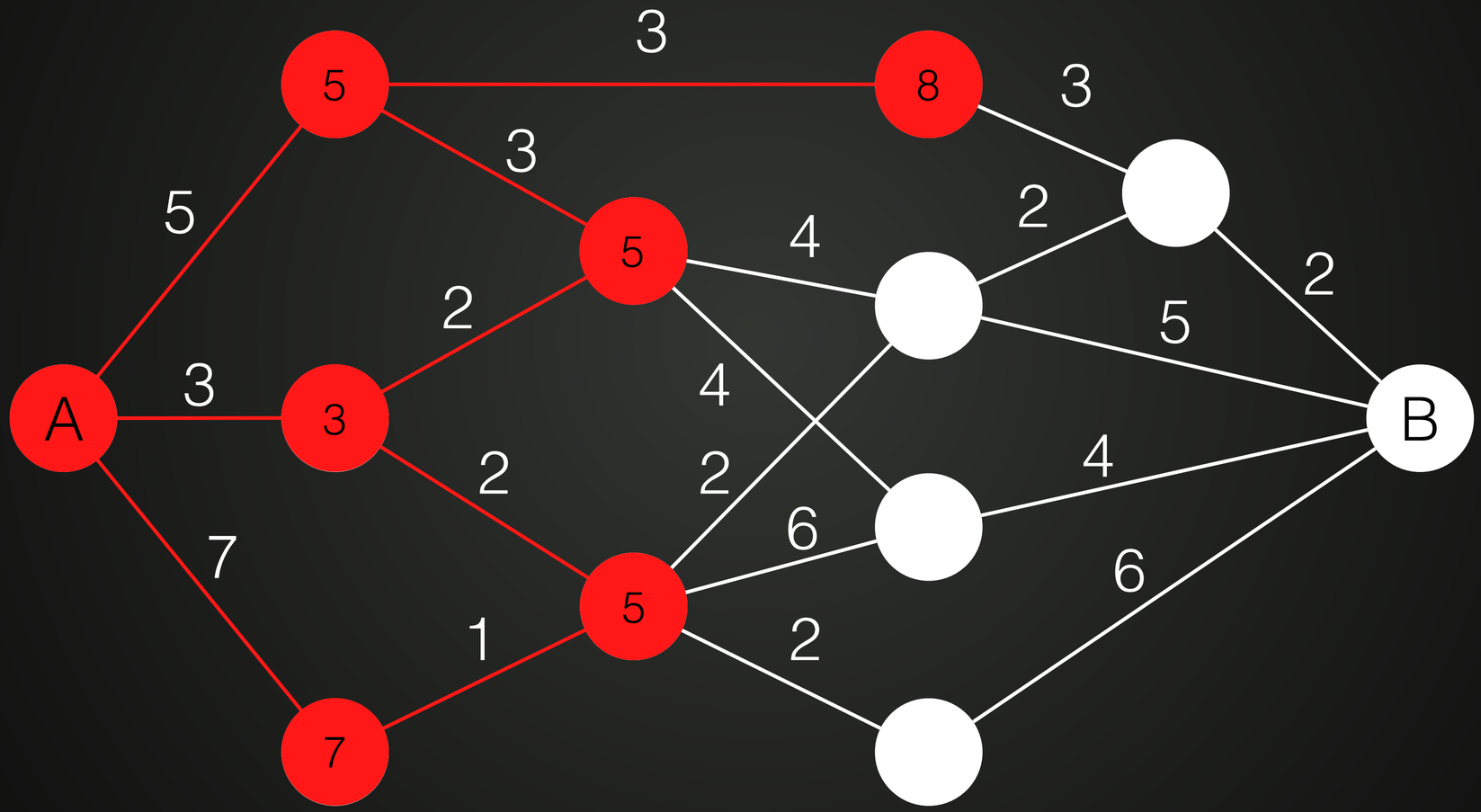


@tobiasnyholm

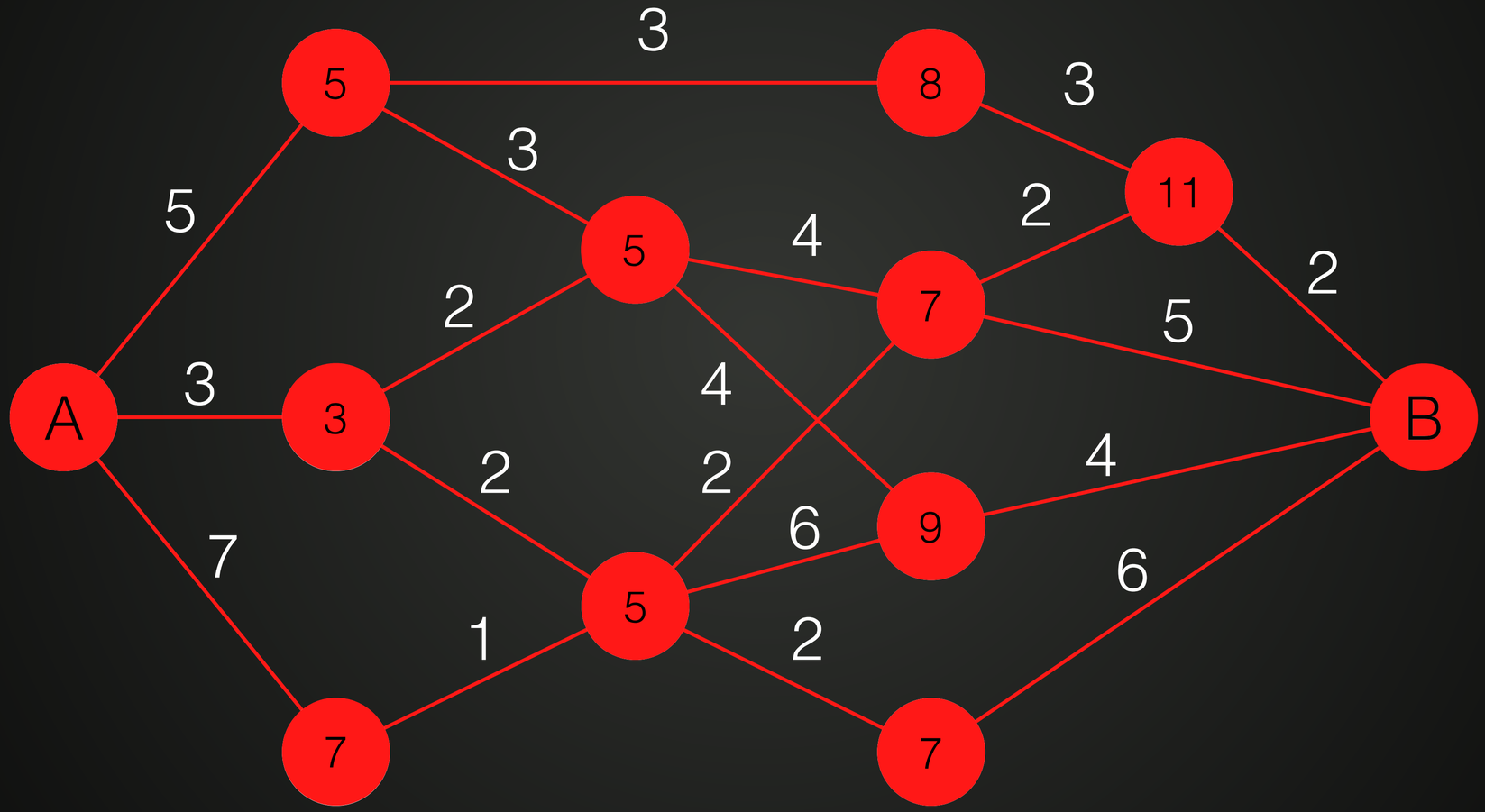
Graph theory



Graph theory

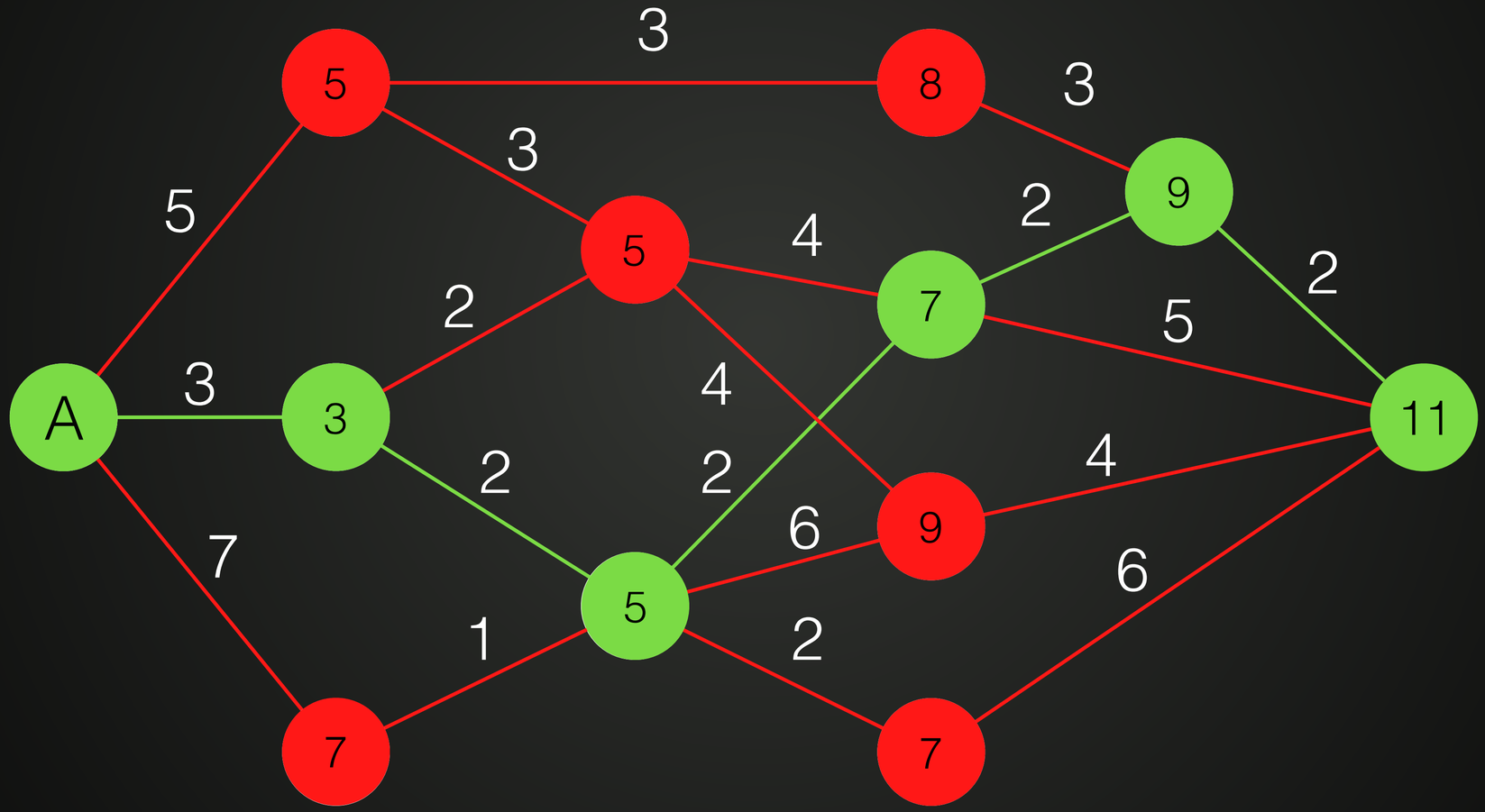


Graph theory



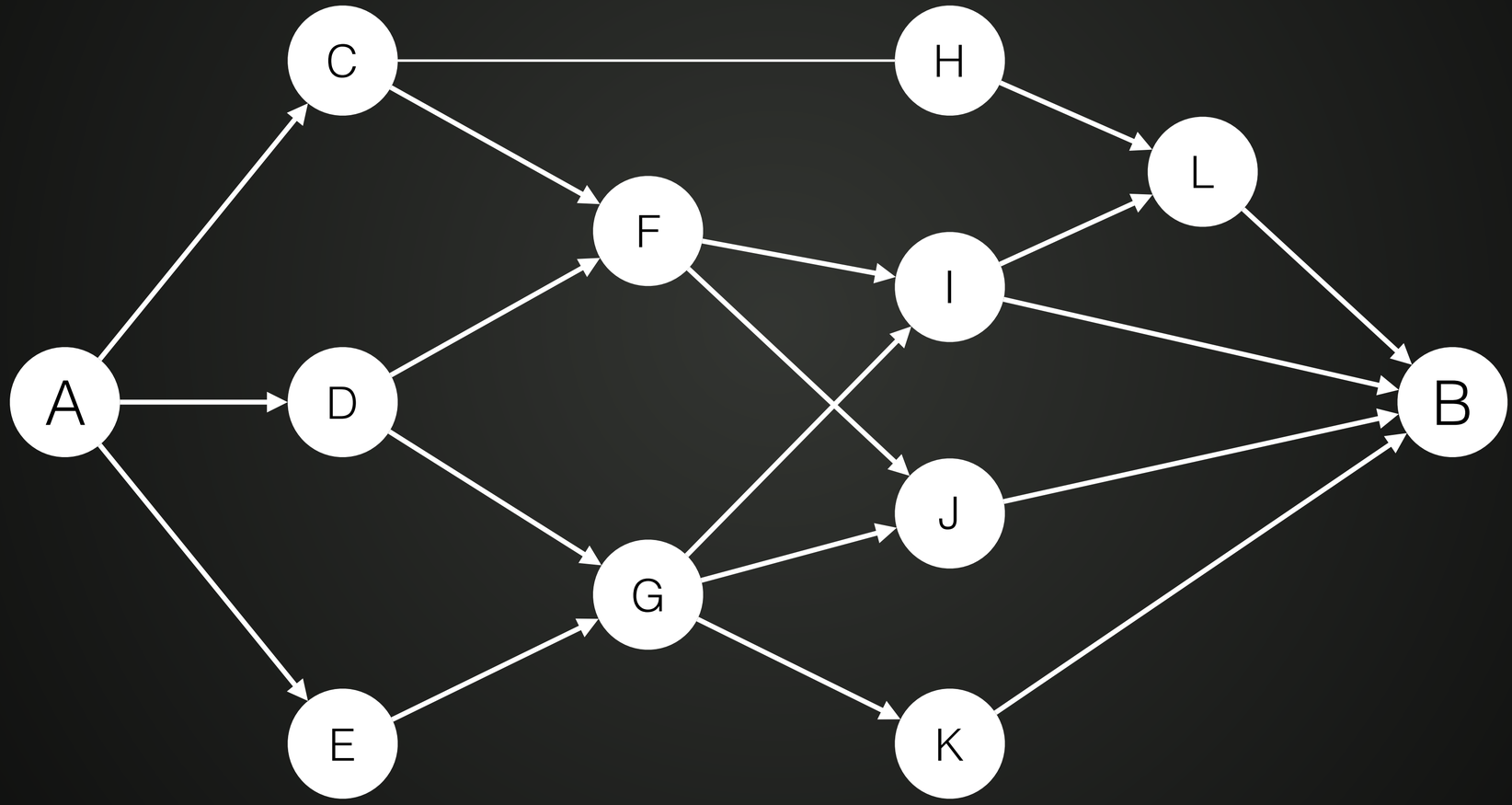
@tobiasnyholm

Graph theory



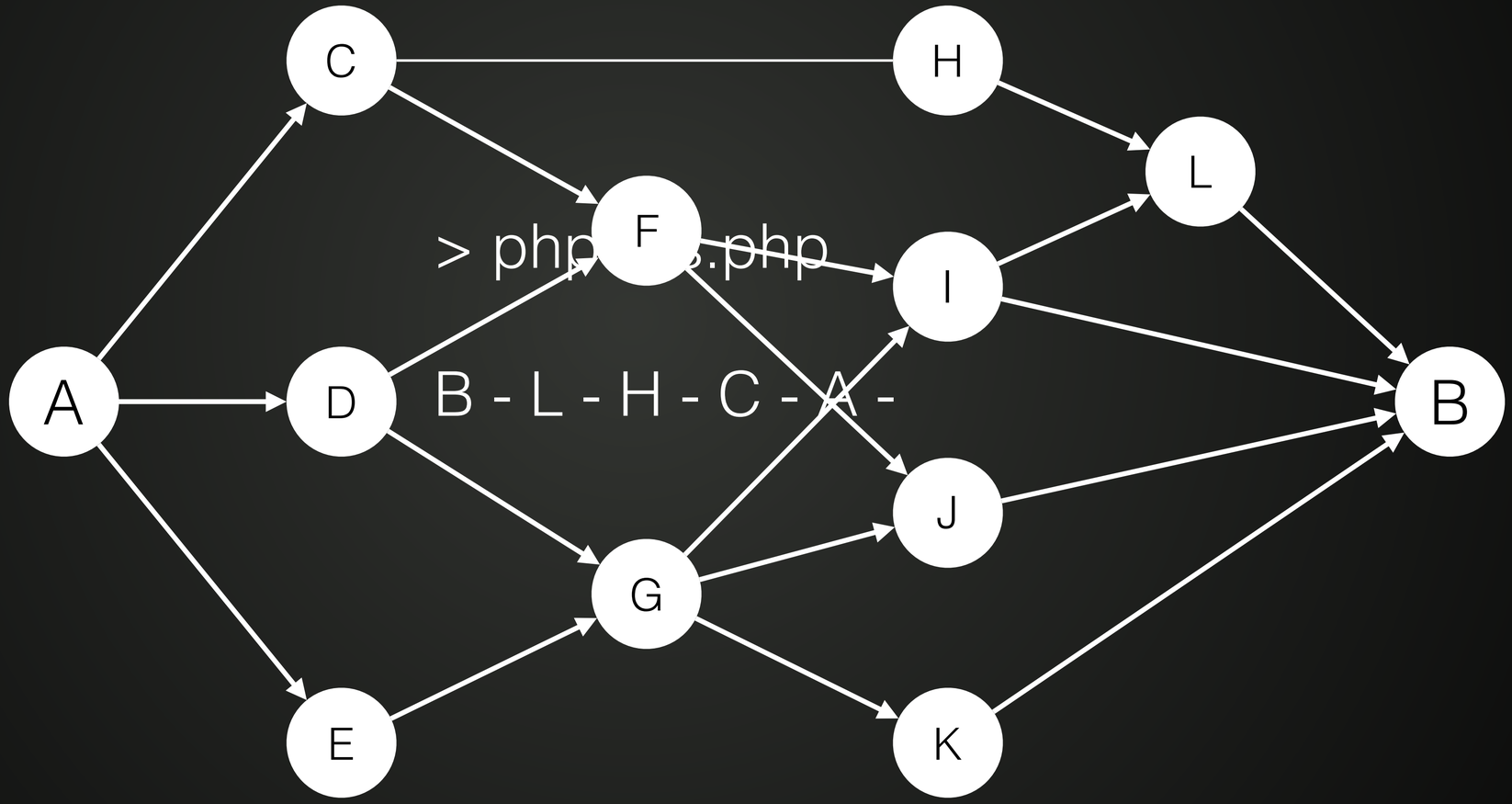
@tobiasnyholm

Breadth first

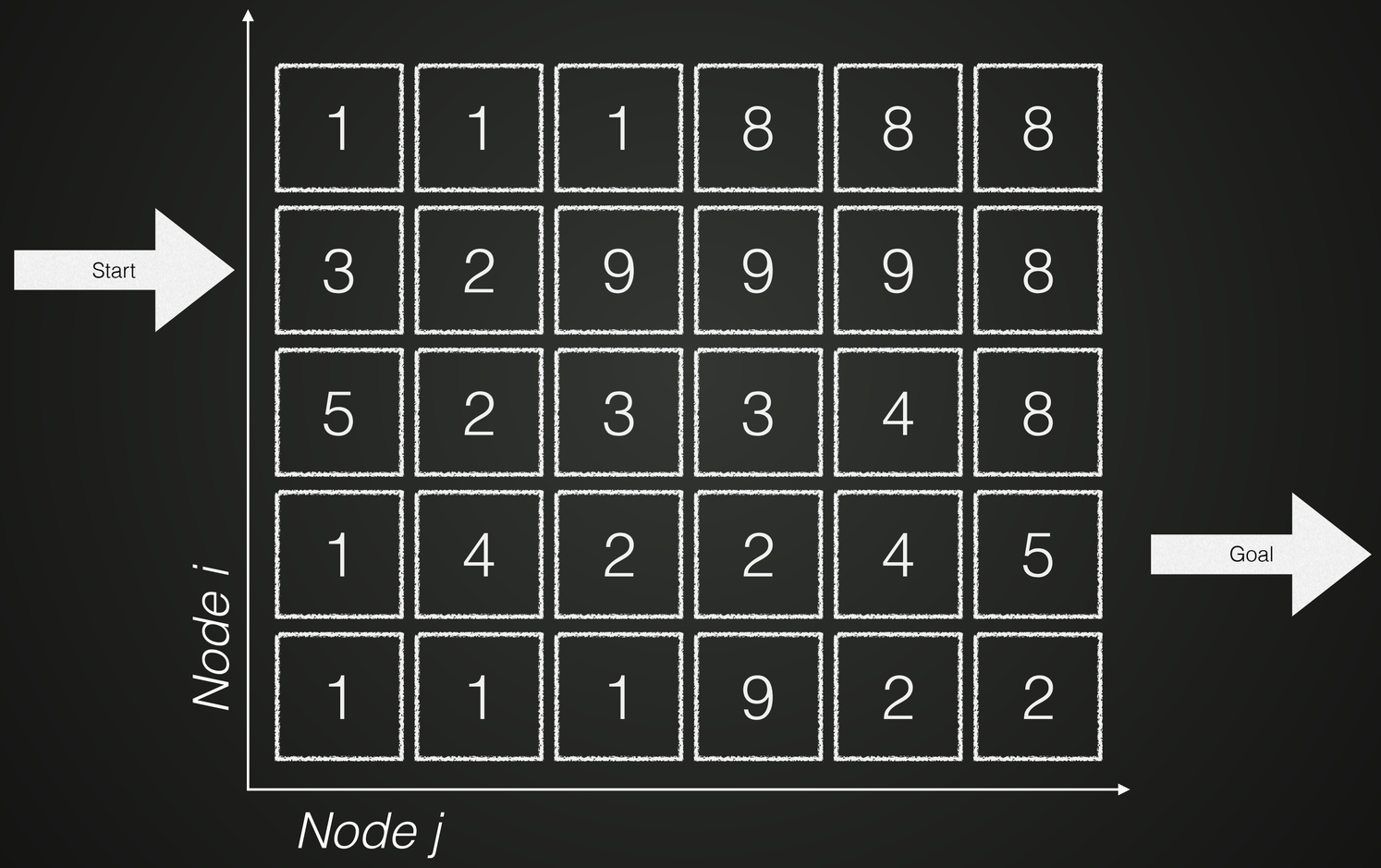


```php
<?php

class Graph {
    private $edges = [];

    public function addEdge(string $from, string $to)
    {
        if (!isset($this->edges[$from])) {
            $this->edges[$from] = [];
        }
        $this->edges[$from][] = $to;
    }

    public function bfs(string $startNode, string $goalNode)
    {
        $queue = [$startNode];
        $visited = [$startNode => true];

        while (!empty($queue)) {
            $current = array_shift($queue);
            if (!isset($this->edges[$current])) {
                continue; // If no edges
            }

            foreach ($this->edges[$current] as $to) {
                $queue[] = $to;
                if (!isset($visited[$to])) {
                    $visited[$to] = $current; // First time
                }
```

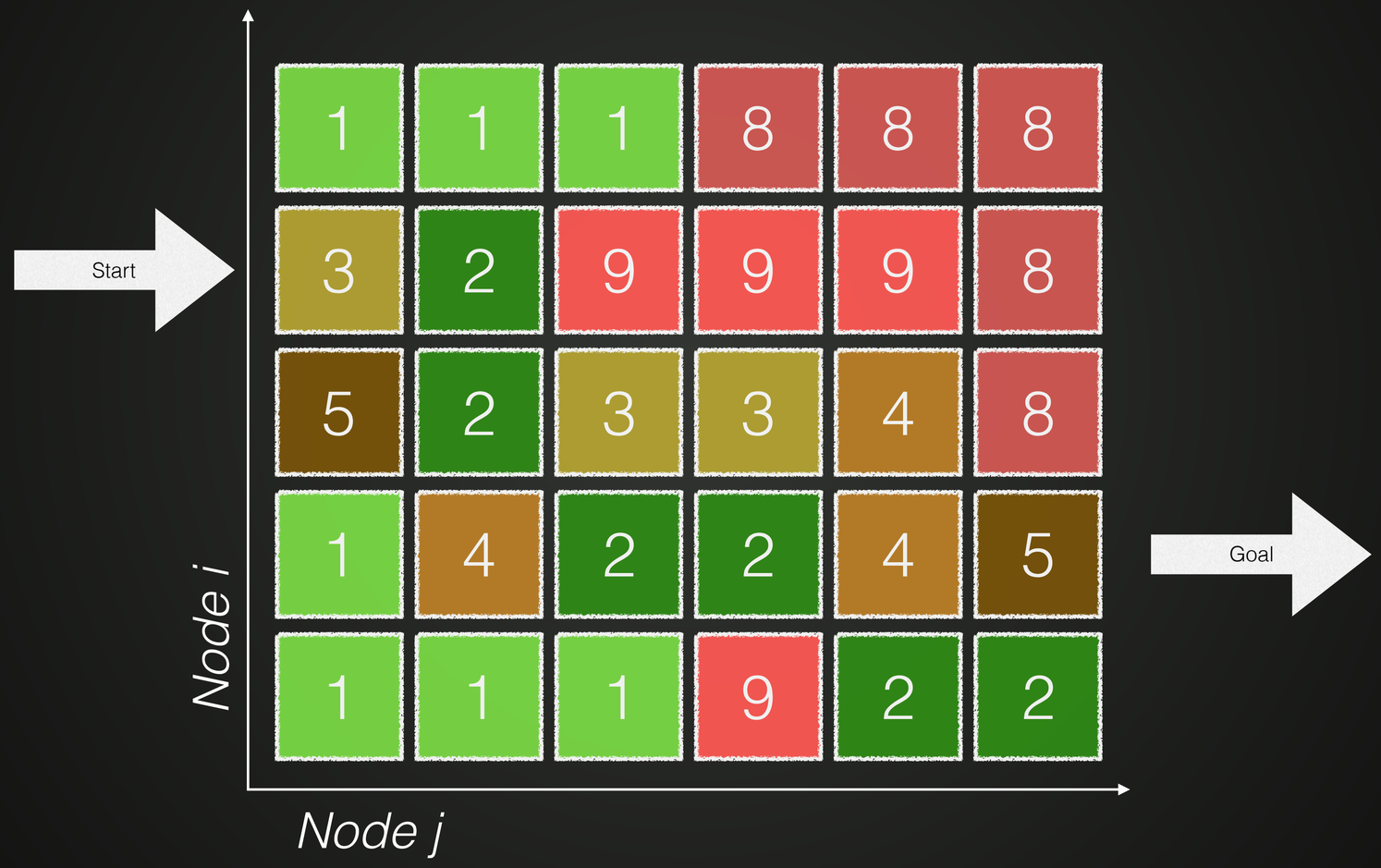
Breadth first



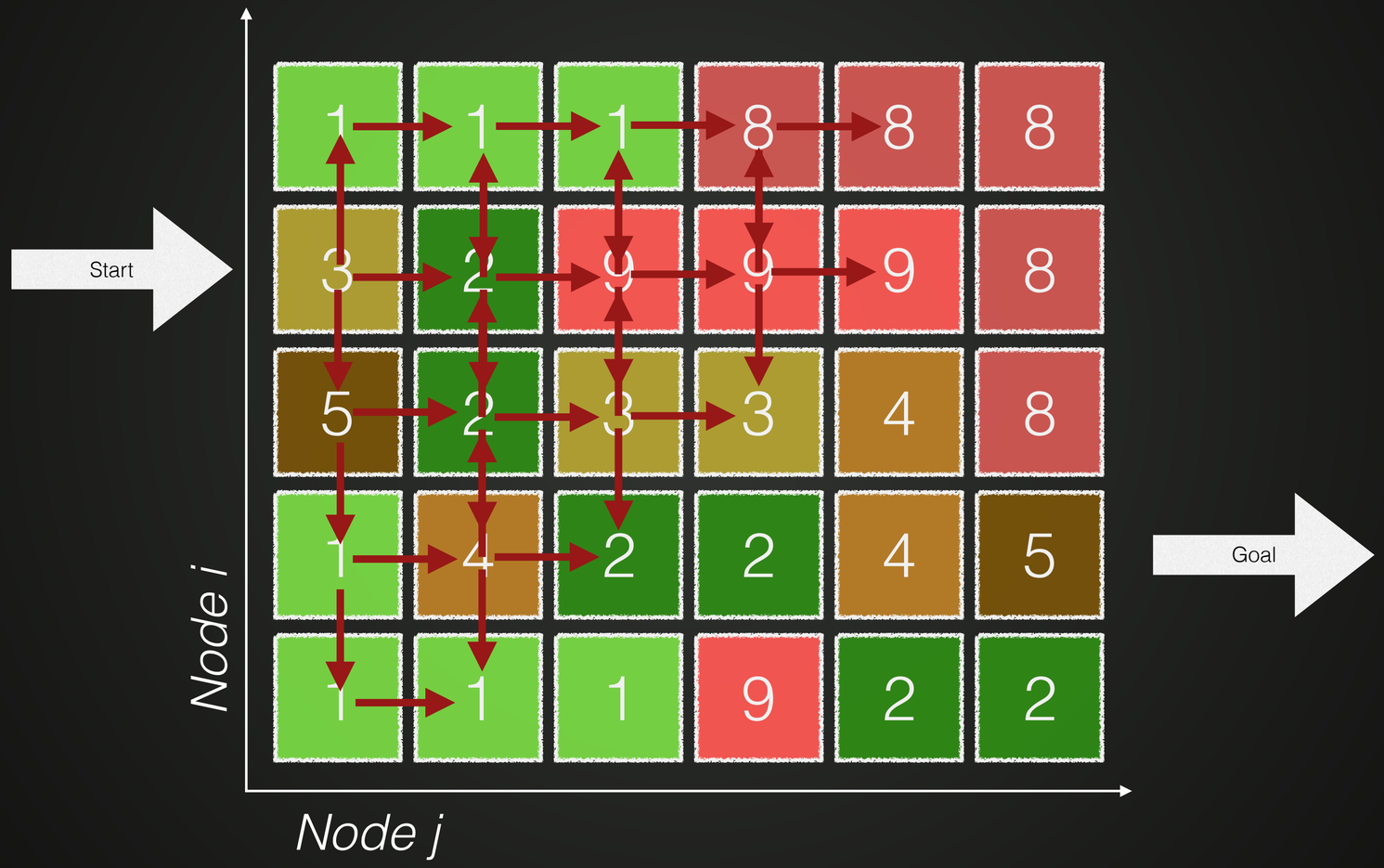
Matrix maze



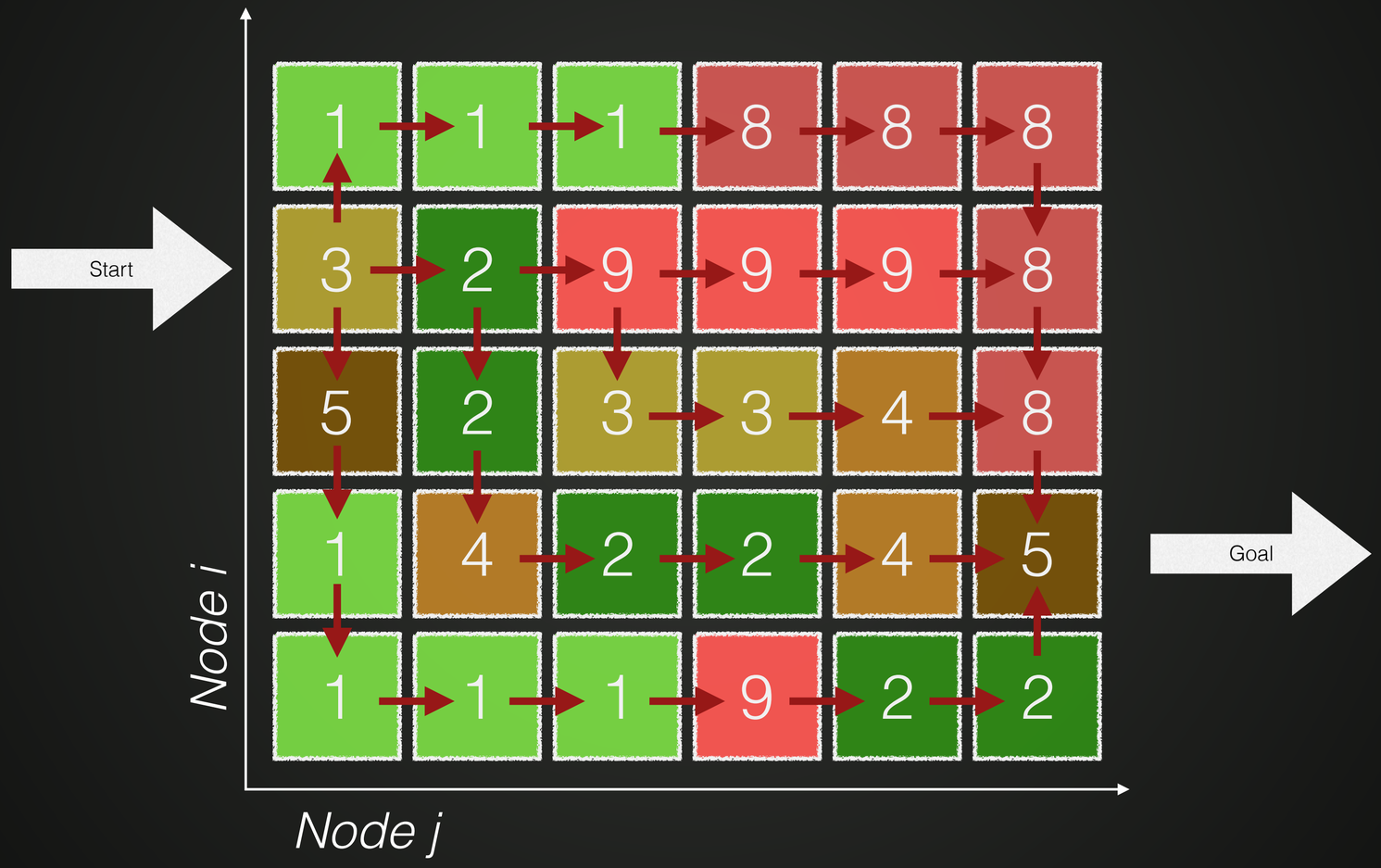
Matrix maze



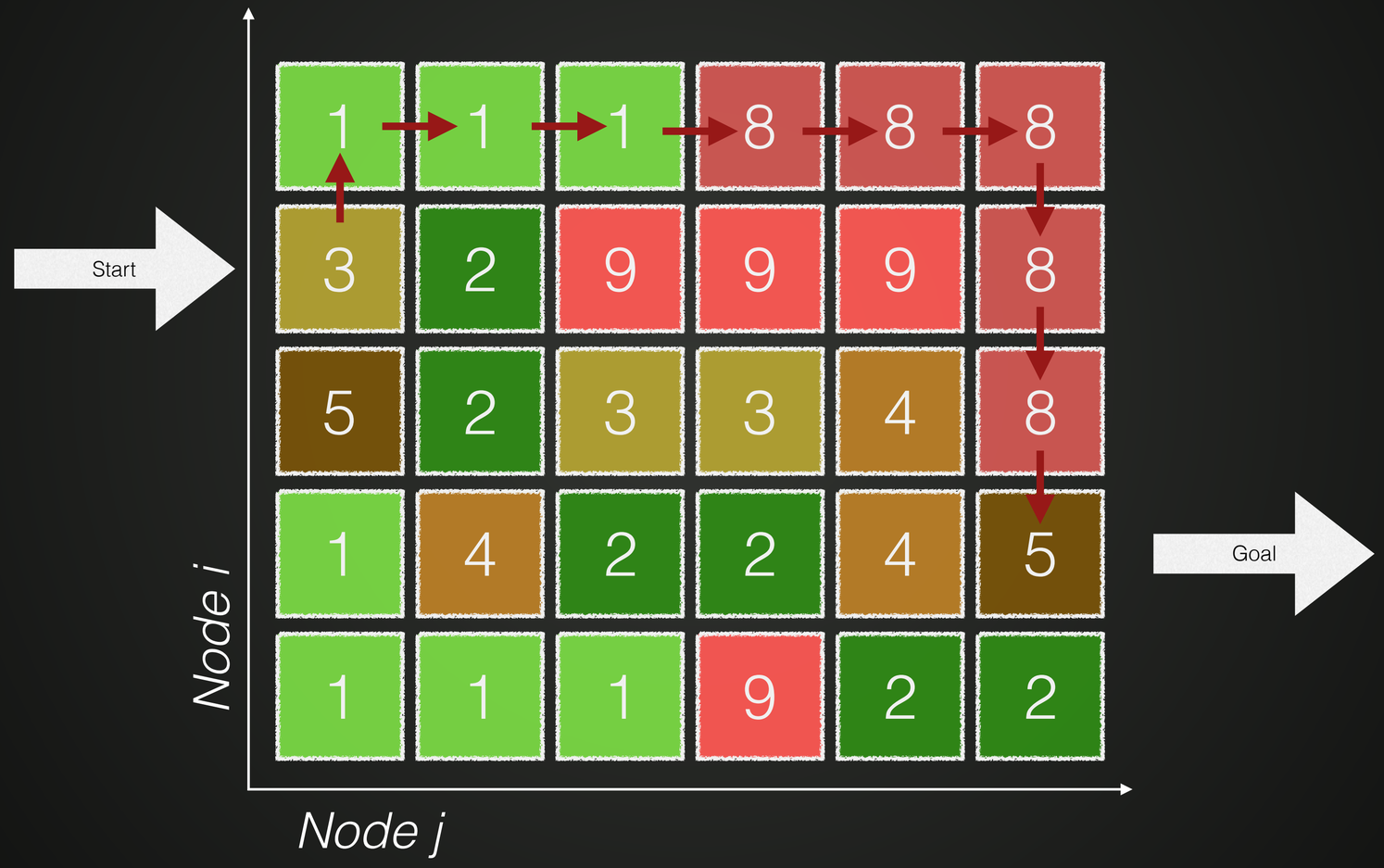
Breadth first



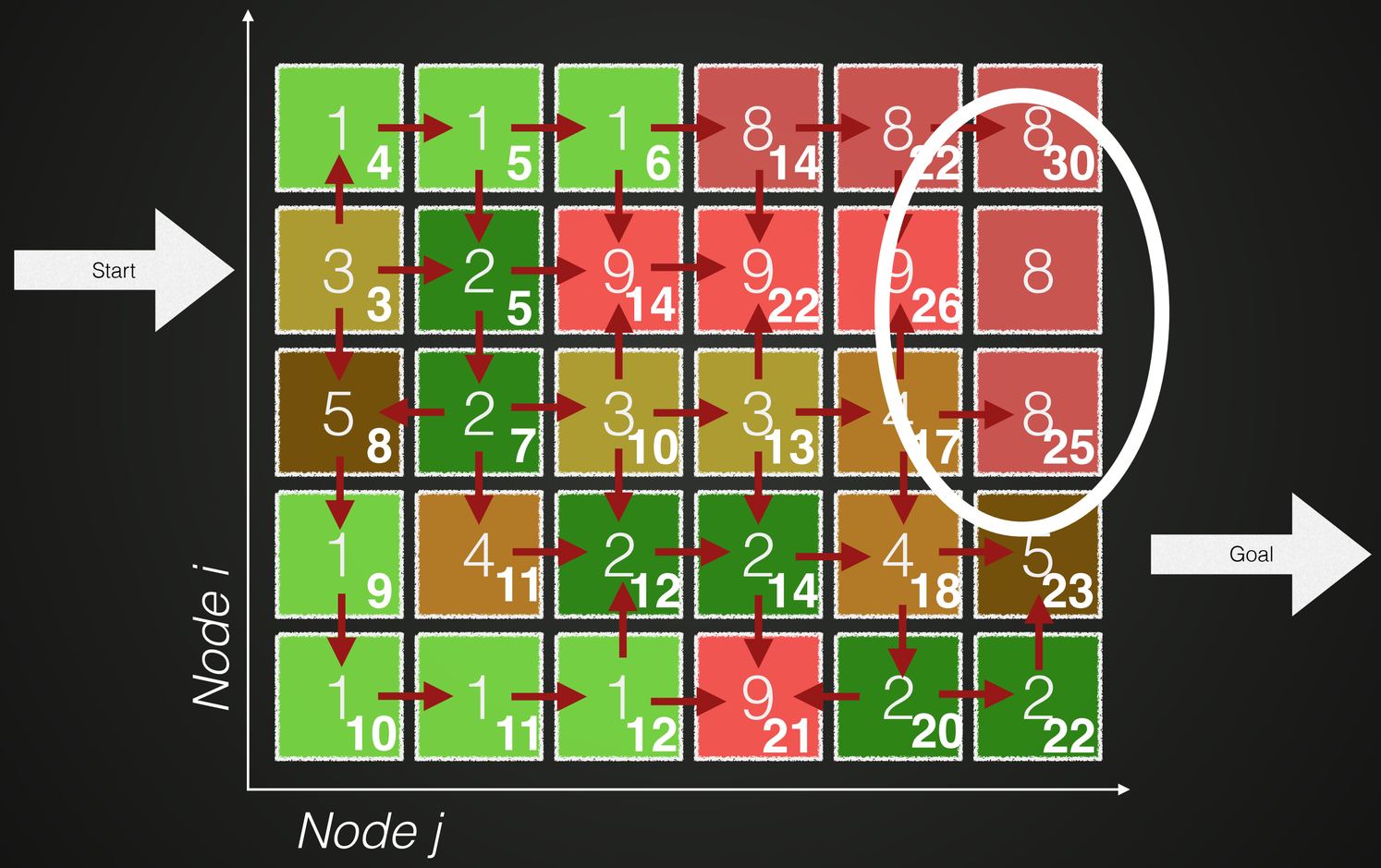
Depth first



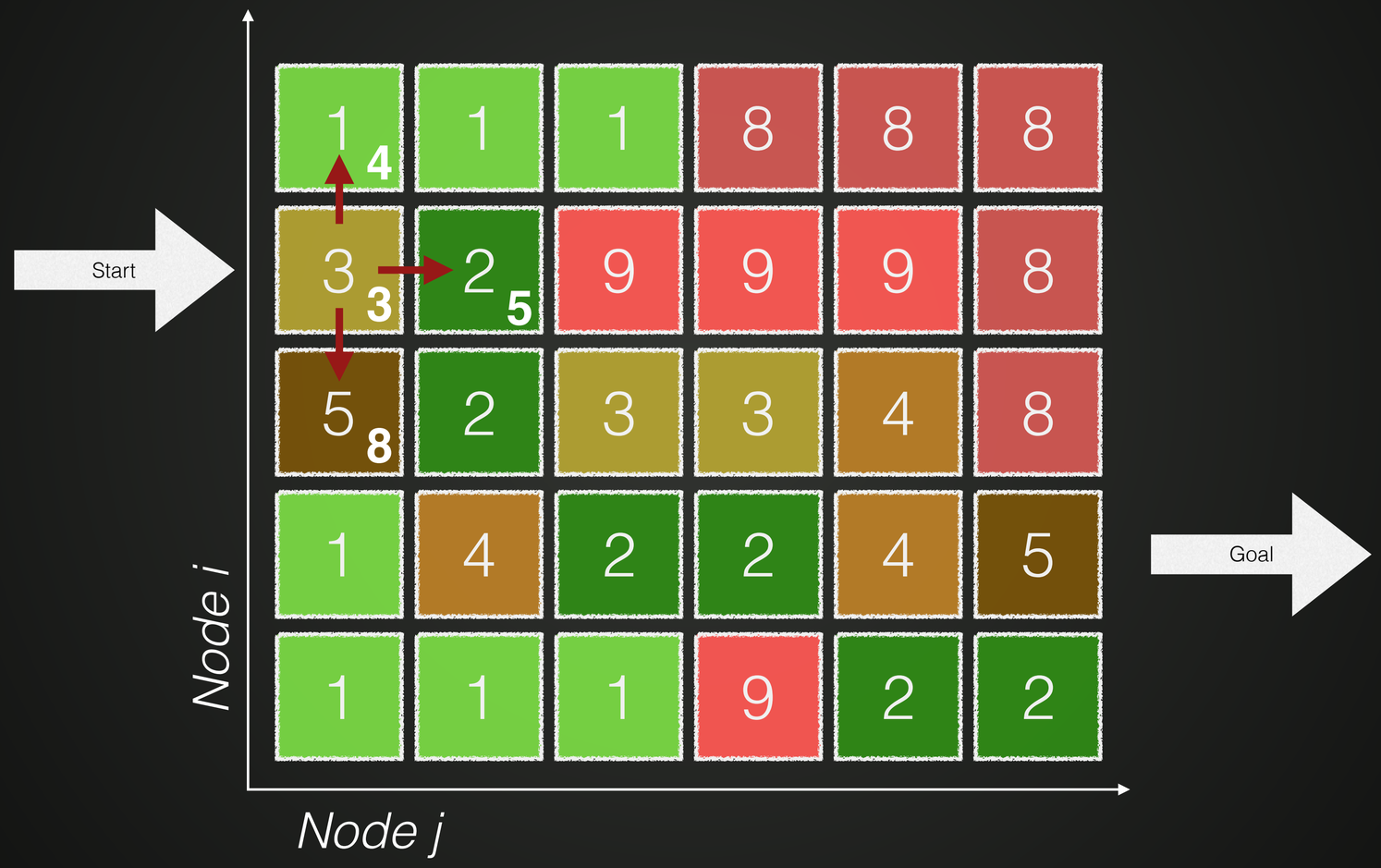
Greedy algorithm



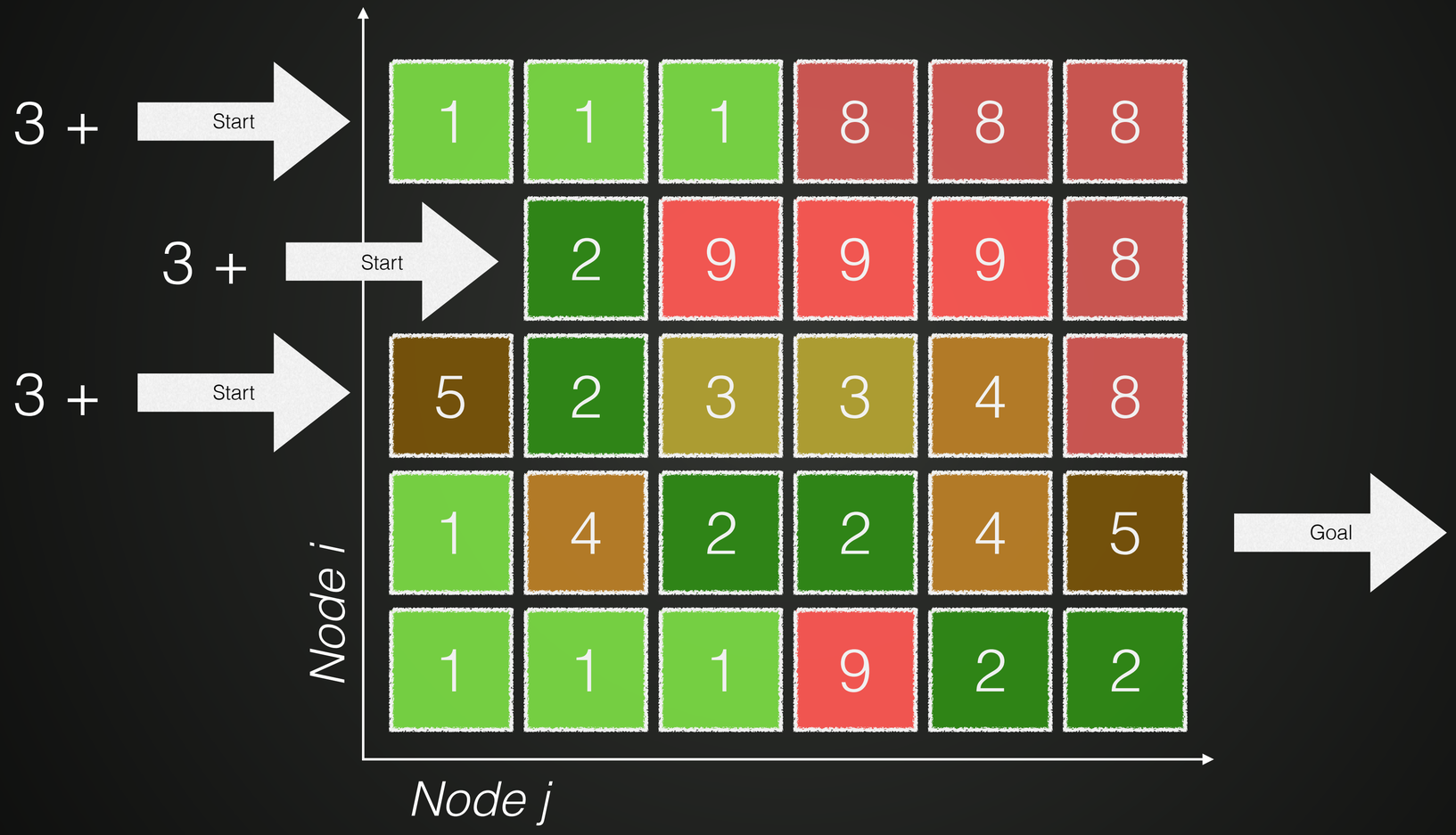
Dijkstra's algorithm



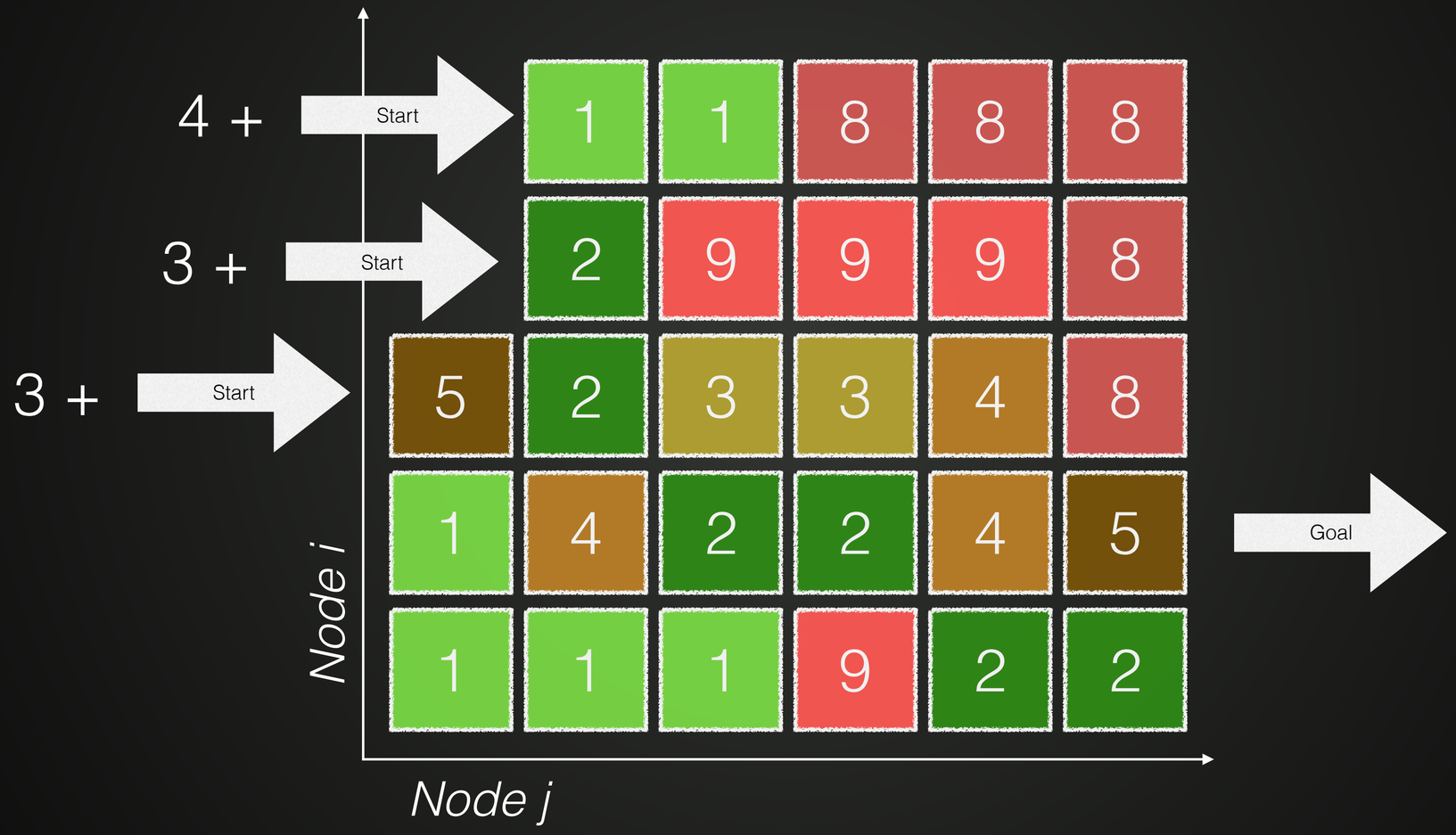
Dijkstra's algorithm



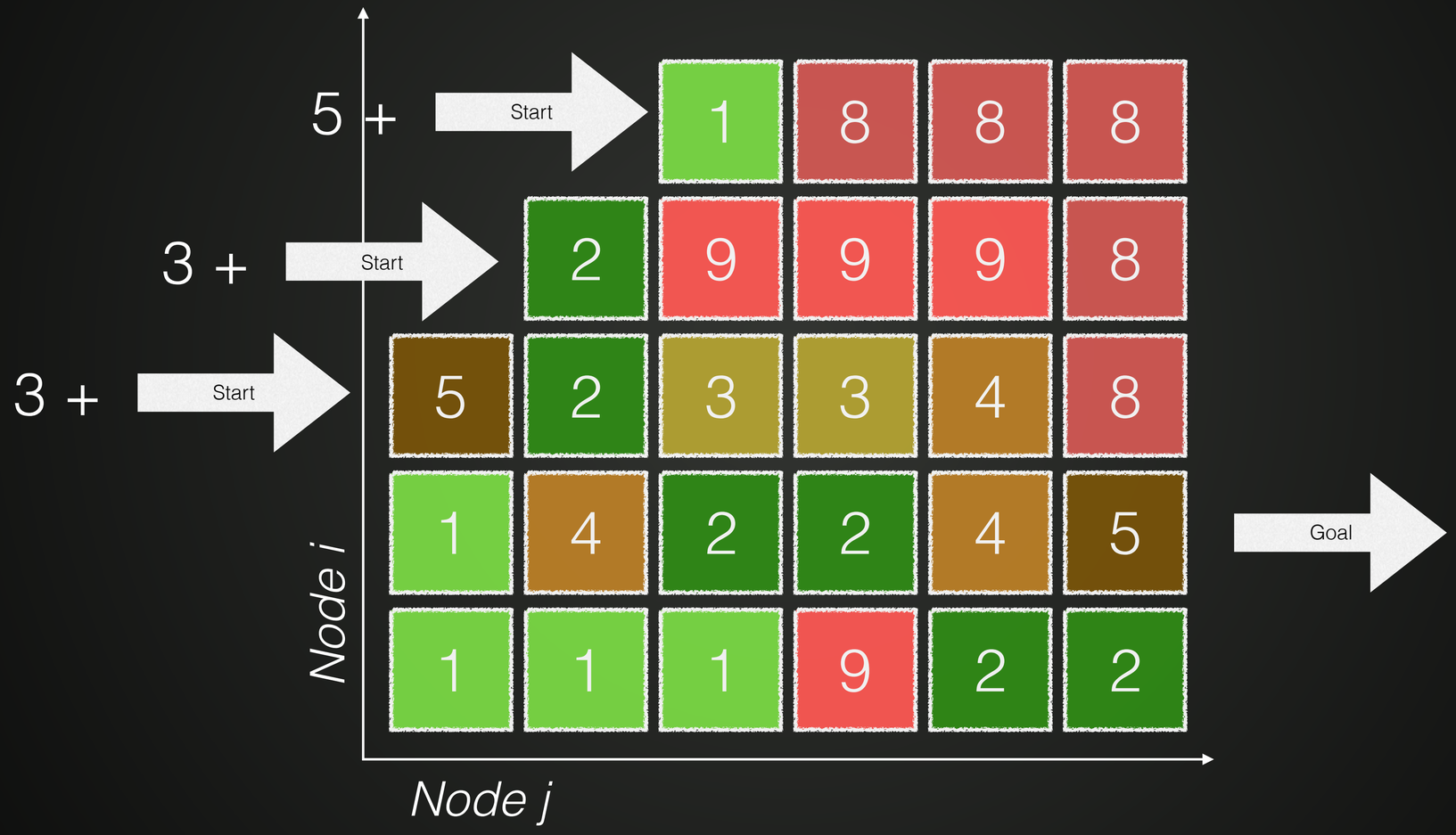
Dijkstra's algorithm



Dijkstra's algorithm



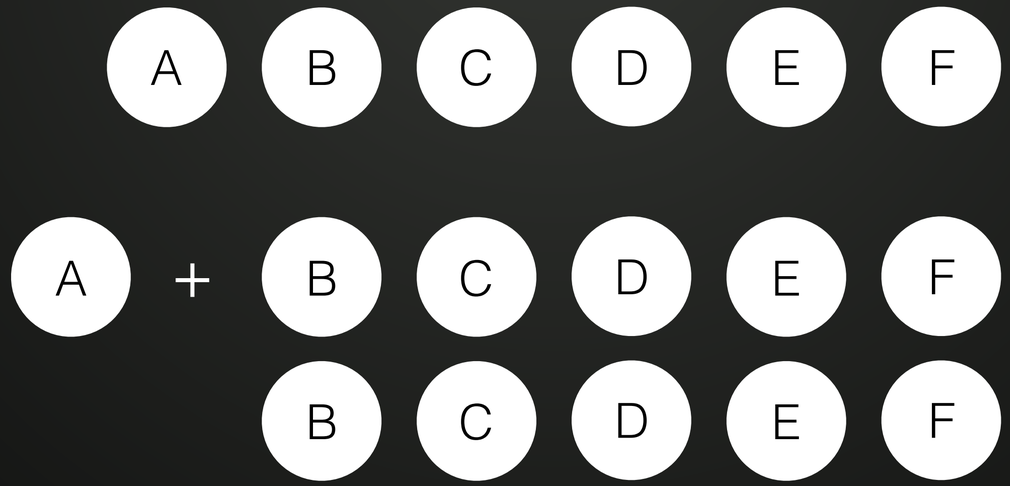
Dijkstra's algorithm



Dynamic programming

Let $V_{(n)} = \{v_1, v_2, \ldots, v_n\}$

Optimal solution $O_{(n)} = \min \left| \begin{matrix} v_n + O_{(n-1)} \\ O_{(n-1)} \end{matrix} \right|$



Dynamic programming

Let $V_{(n)} = \{v_1, v_2, \ldots, v_n\}$

Optimal solution $O_{(n)} = \min \left| \begin{matrix} v_n + O_{(n-1)} \\ O_{(n-1)} \end{matrix} \right|$

Dynamic programming

Let $V_{(n)} = \{v_1, v_2, \ldots, v_n\}$

Optimal solution $O_{(n)} = \min \left| \begin{matrix} v_n + O_{(n-1)} \\ O_{(n-1)} \end{matrix} \right|$

Pause

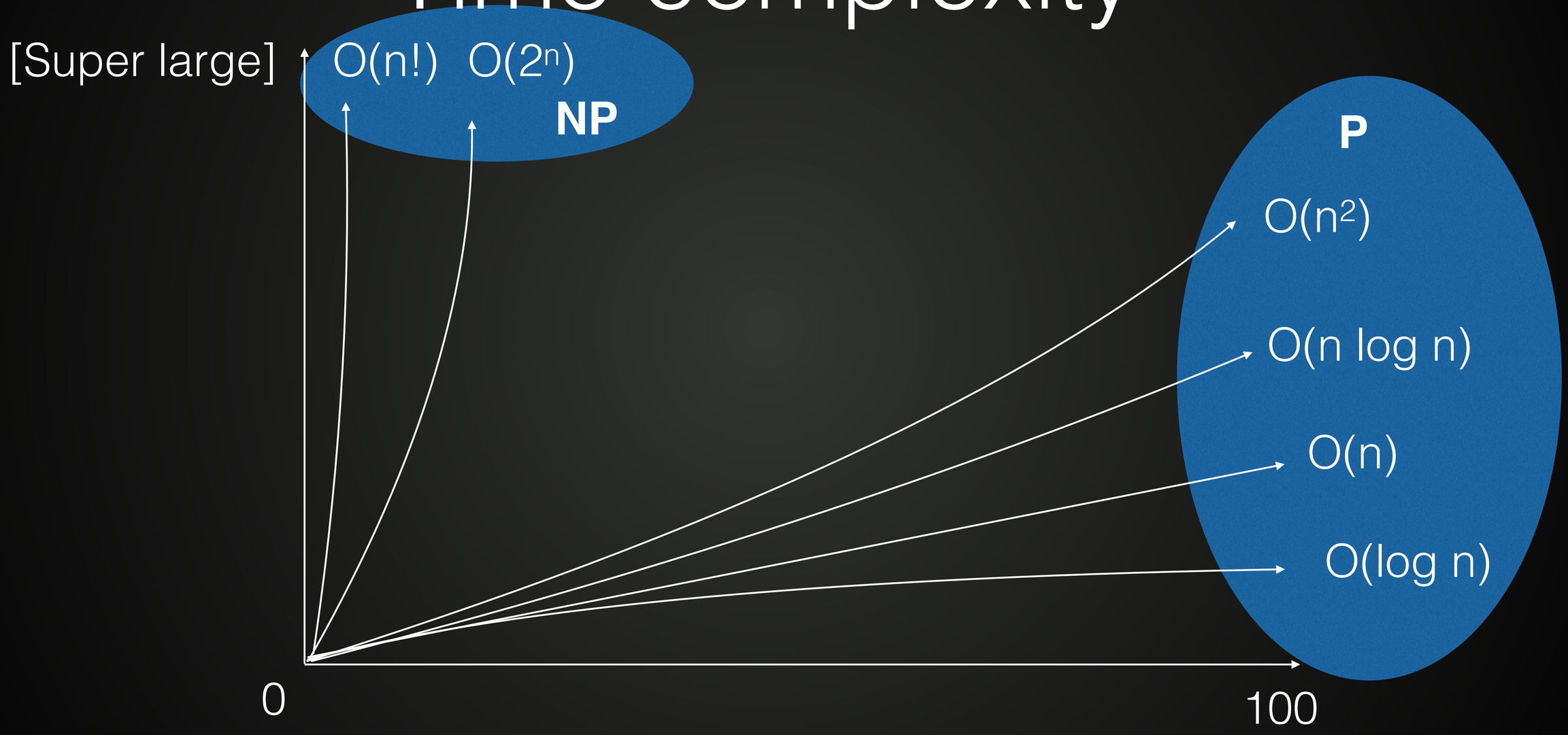
$1.000.000

Time complexity

Breadth-first search	$O(\|V\| + \|E\|)$	~ O(n)
Depth-first search	$O(\|V\| + \|E\|)$	~ O(n)
Greedy algorithm	$O(\|V\| + \|E\|)$	~ O(n)
Dijkstra's algorithm	$O(\|V\| \log(\|V\|) + \|E\|)$	~ O(n log n)
Bellman-Held-Karp	$O(2^n n^2)$	

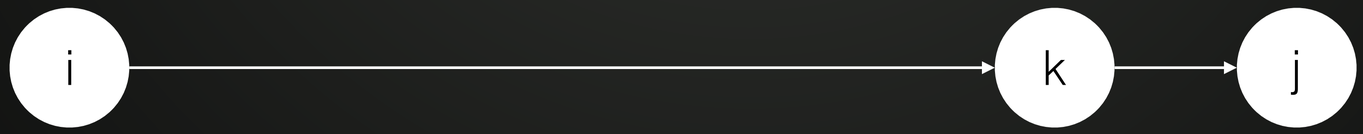
@tobiasnyholm

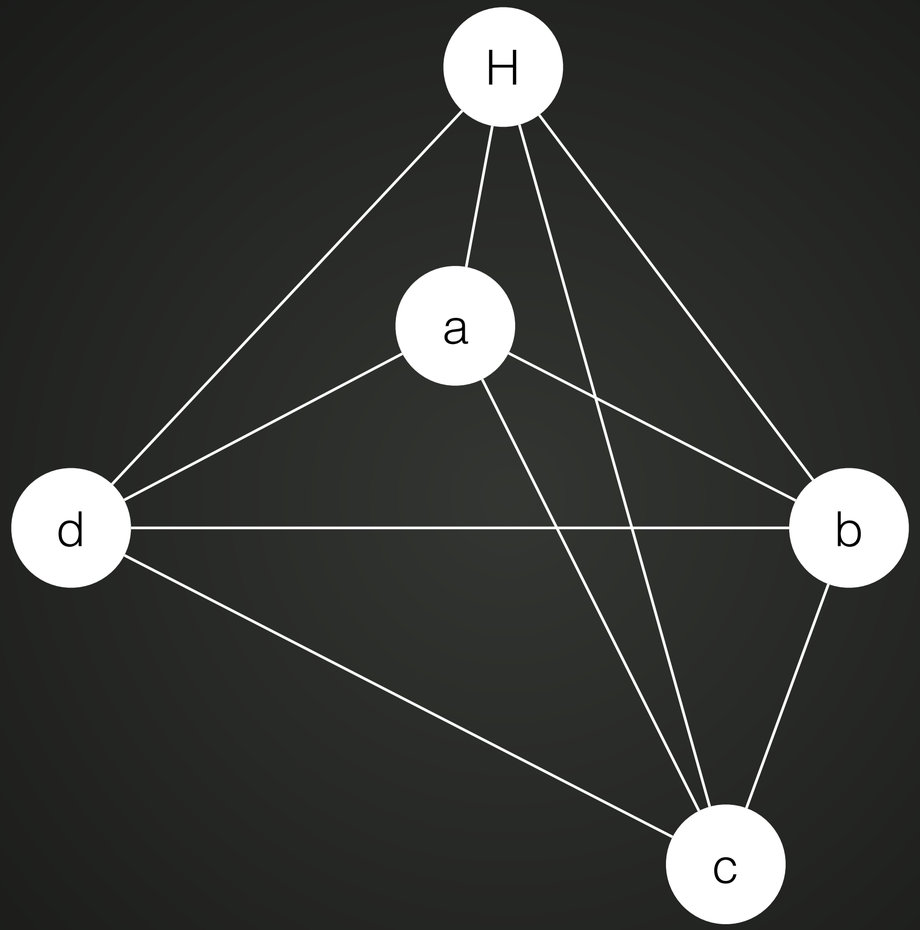
Time complexity

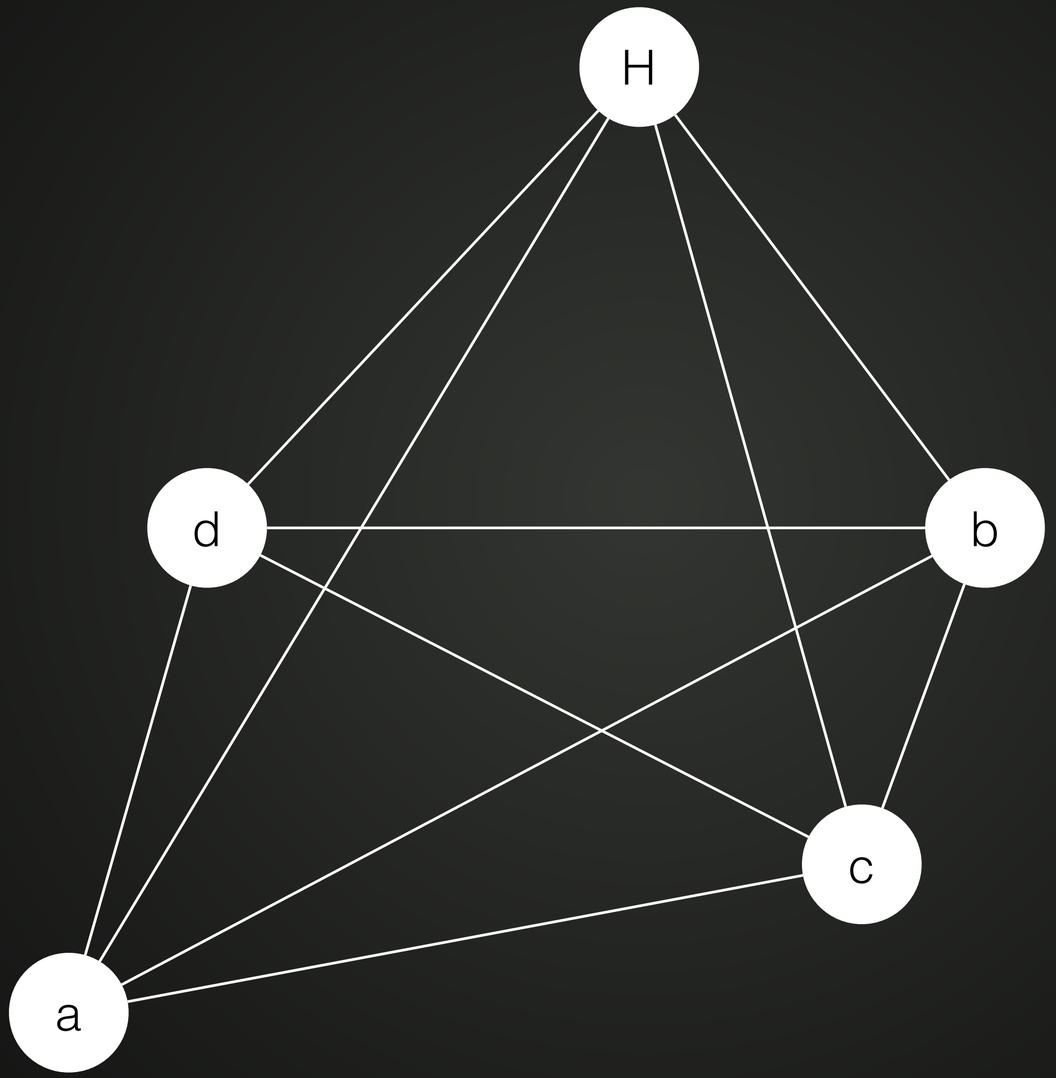


Traveling salesman

- Isn’t traveling salesman just a edge case of Dijkstra’s algorithm for shortest path?
- Could we use Dijkstra’s algoritm and require all length of the solution?
- If we do not allow duplicates?



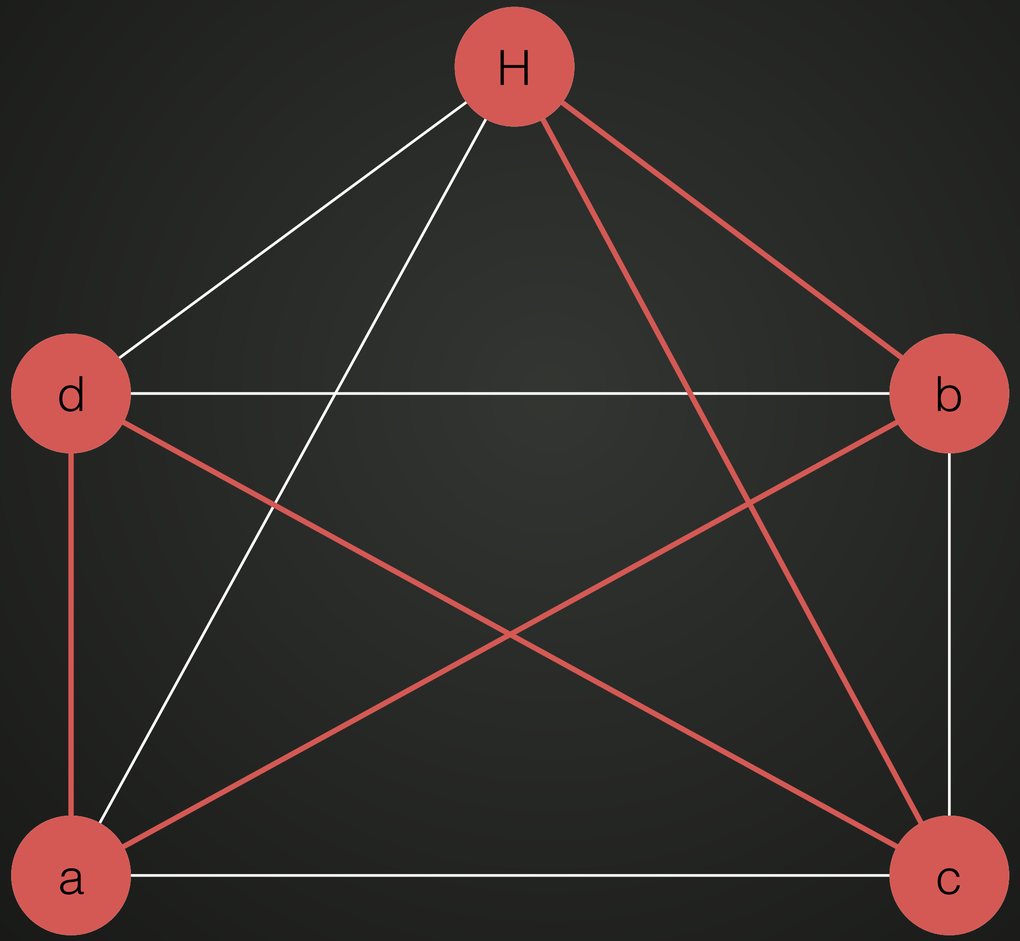

H
a
d
b
c


H
d
b
c
a

“Why do we spend so much time on TSP?”

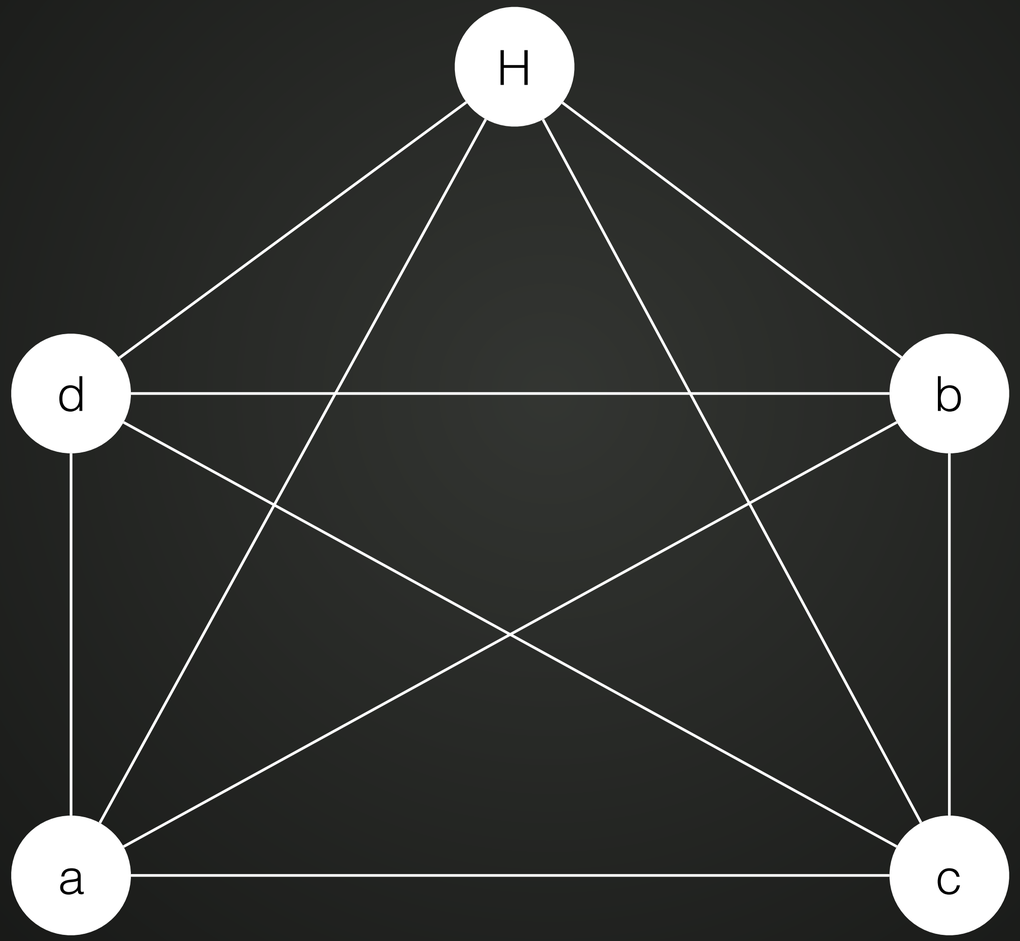
Let’s try TSP

Traveling salesman

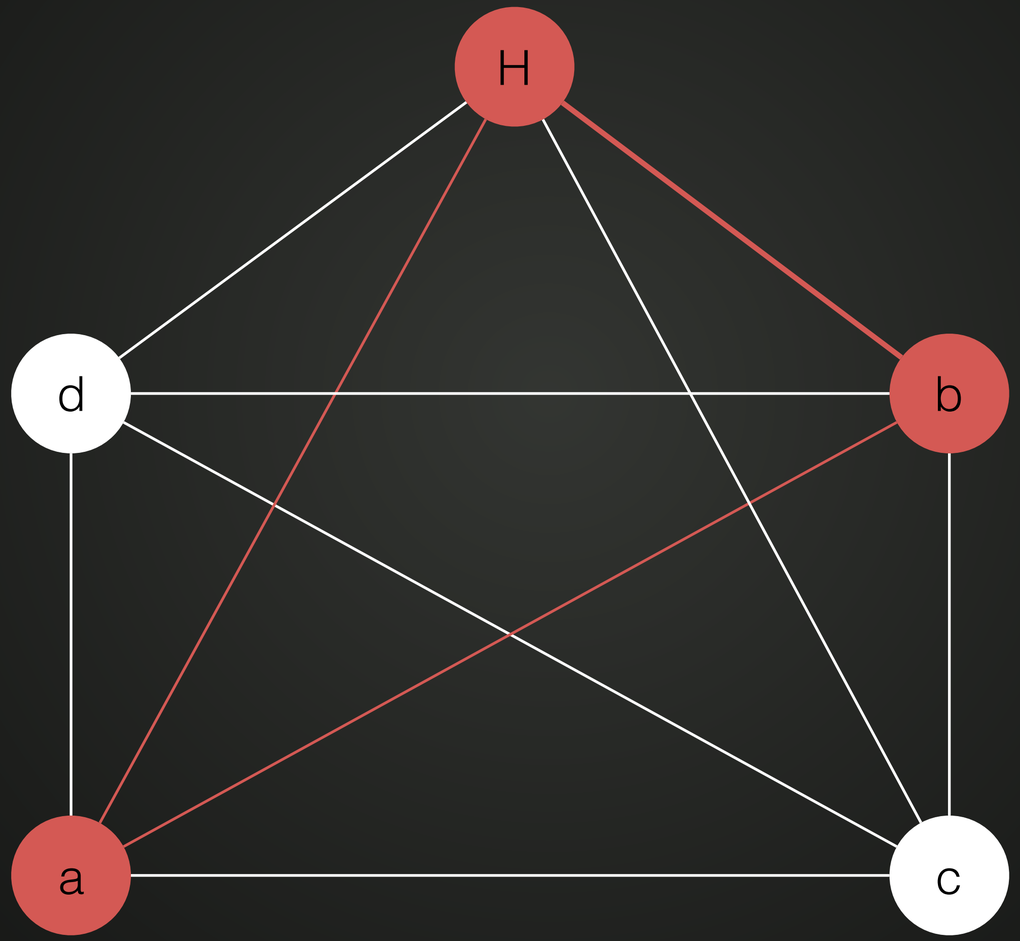


O(n!)

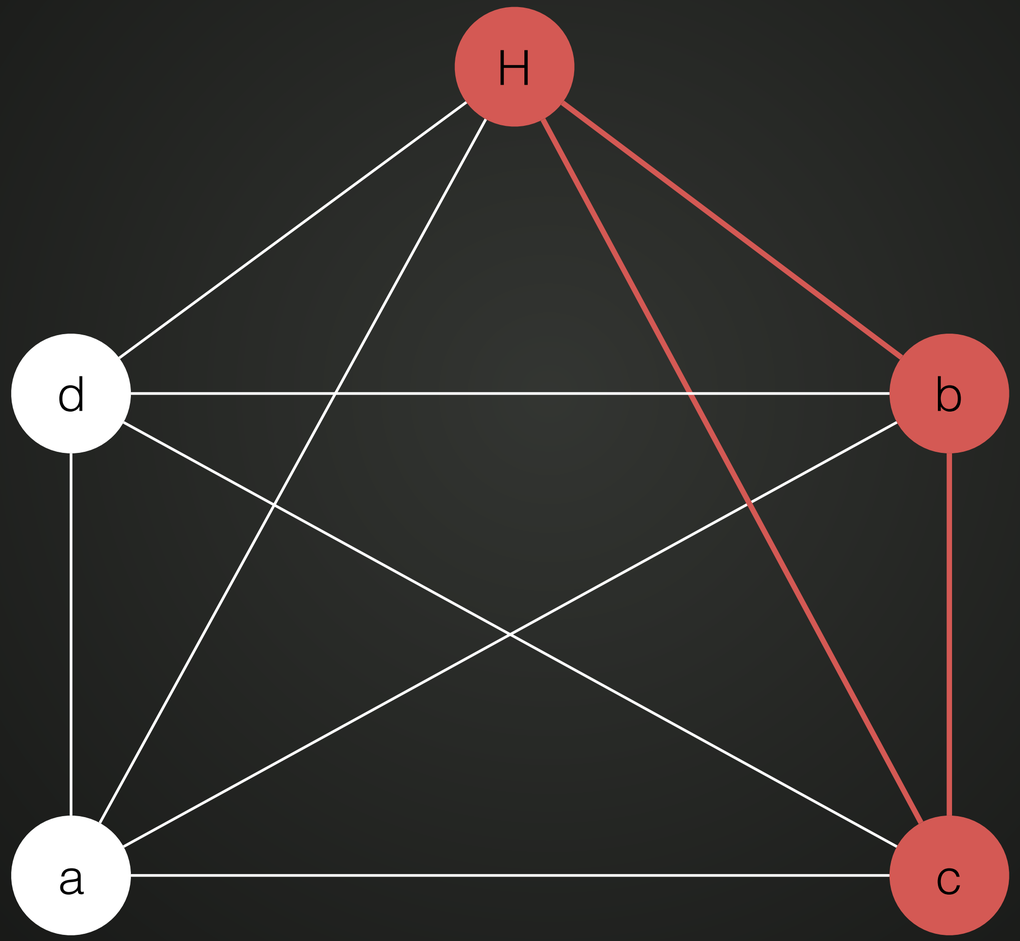
Traveling salesman



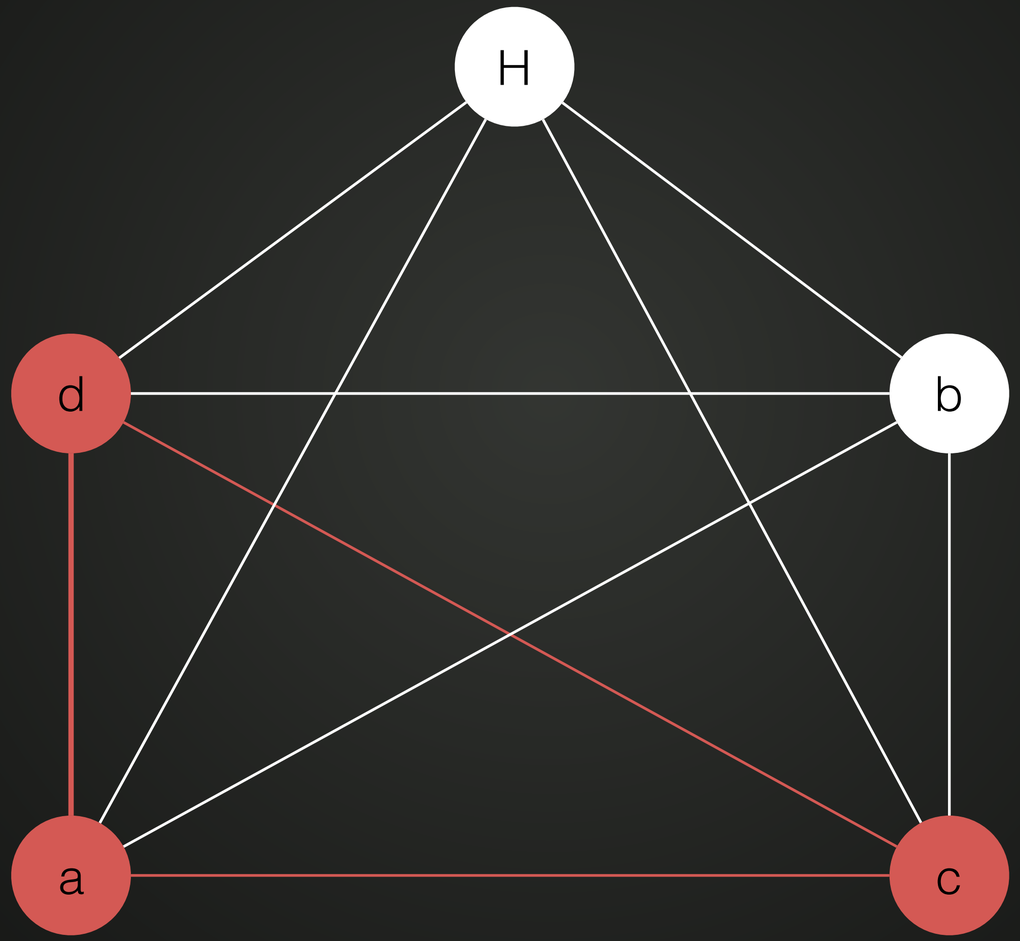
Traveling salesman



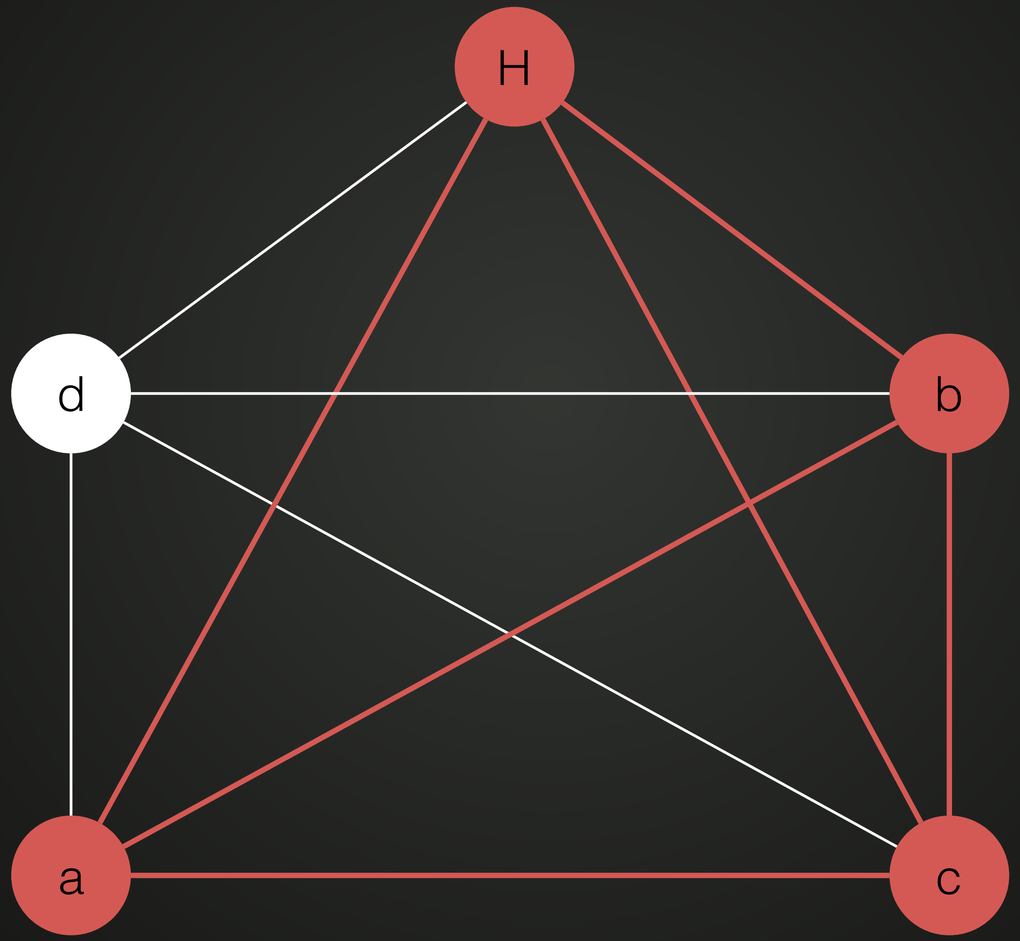
Traveling salesman



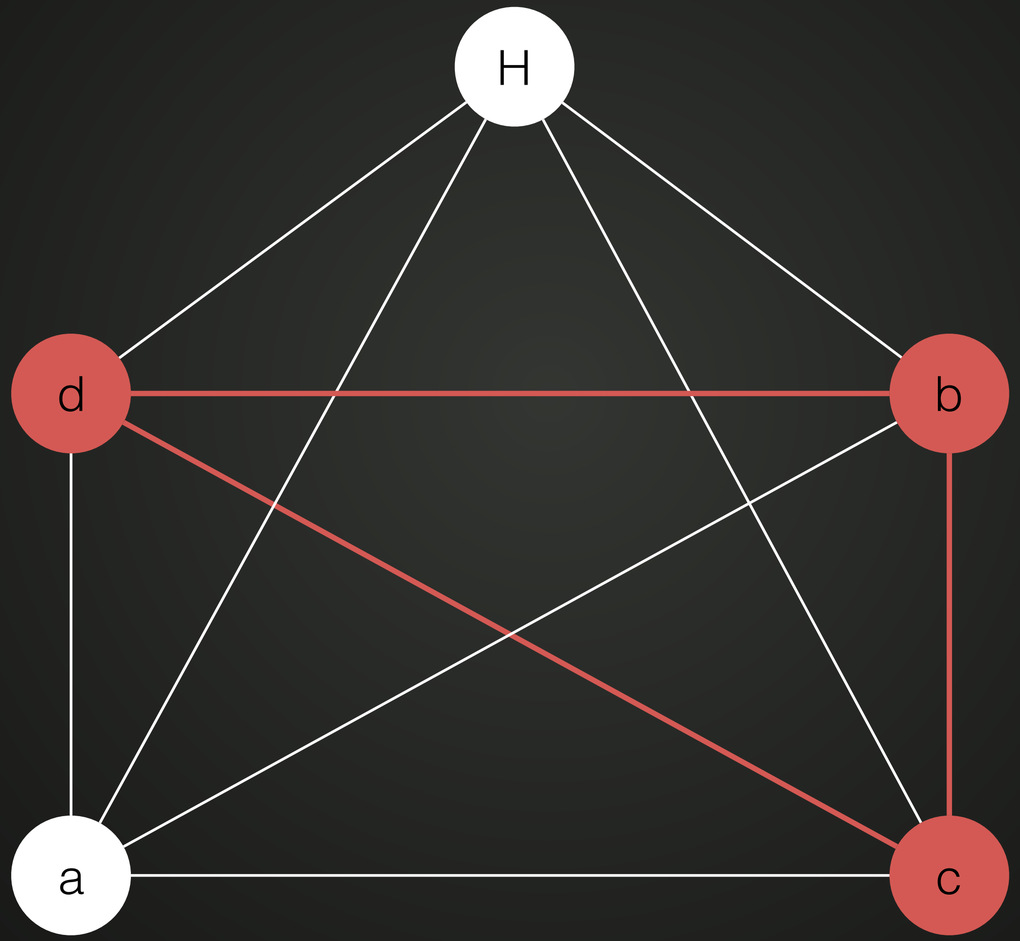
Traveling salesman



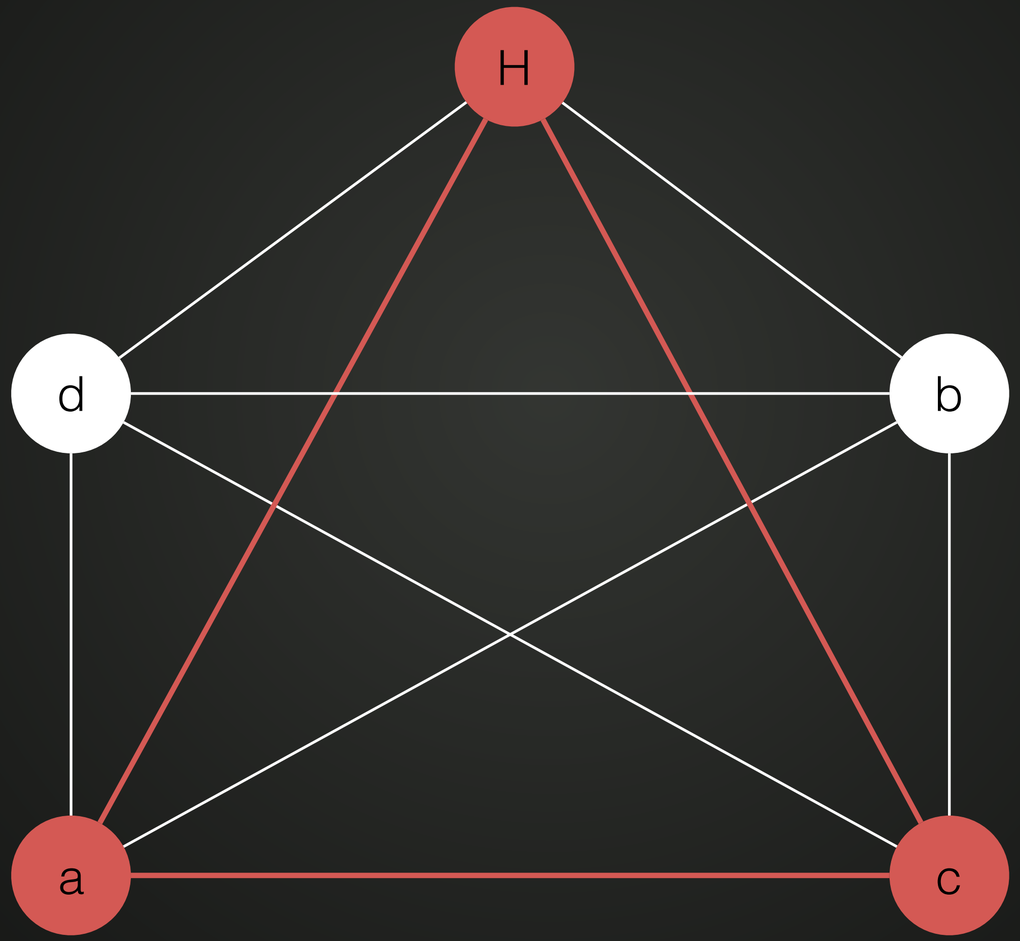
Traveling salesman



Traveling salesman



Traveling salesman



$O(2^n n^2)$

“This is good and all, but I’m building websites”

¯_(ツ)_/¯

Questions?

https://joind.in/talk/566f7