

Worker Threads Crash Course

Rich Trott
he/him
@Trott

An Introduction To The Things I Like/Use/Want To Talk About In Worker Threads

Rich Trott
he/him
@Trott

I Was Doing A Thing And I Wanted To Talk About The Thing And Ended Up Talking About Worker Threads

Rich Trott
he/him
@Trott



Photo by Vladislav Klapin on Unsplash

@Trott

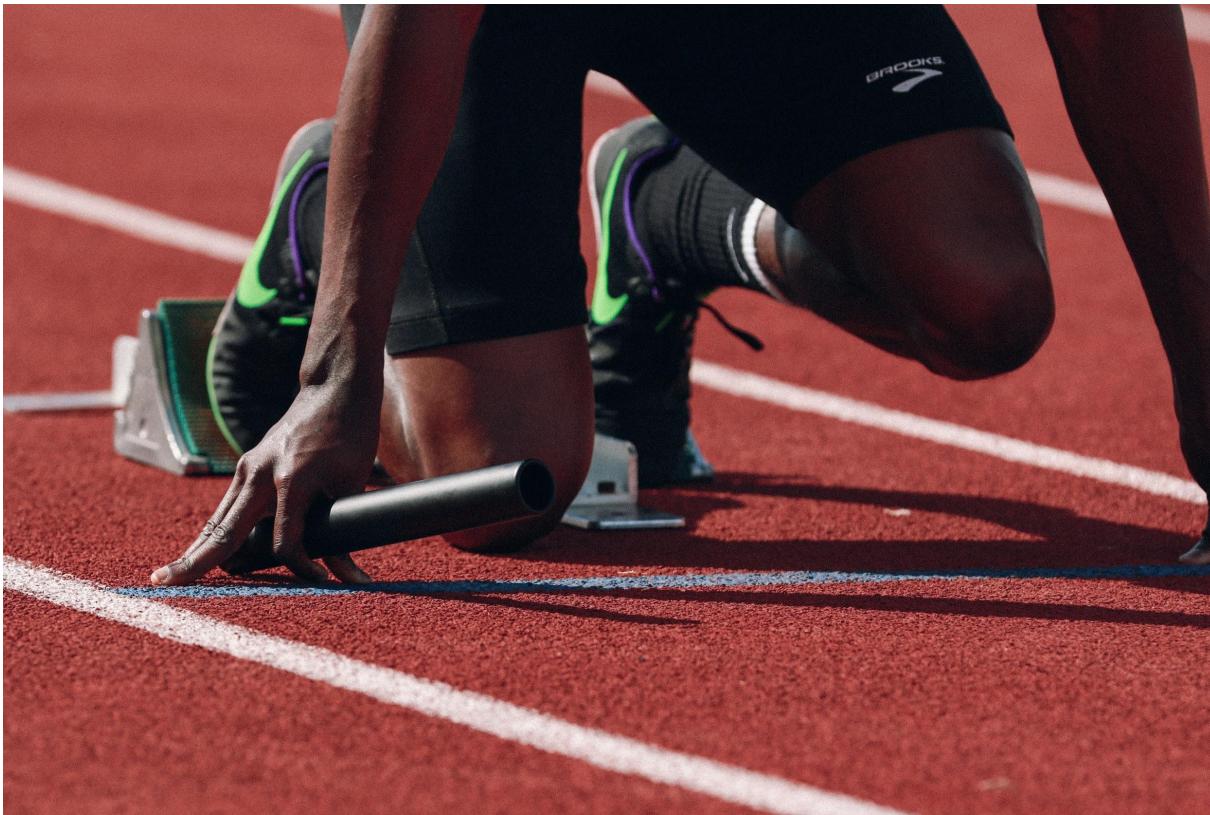
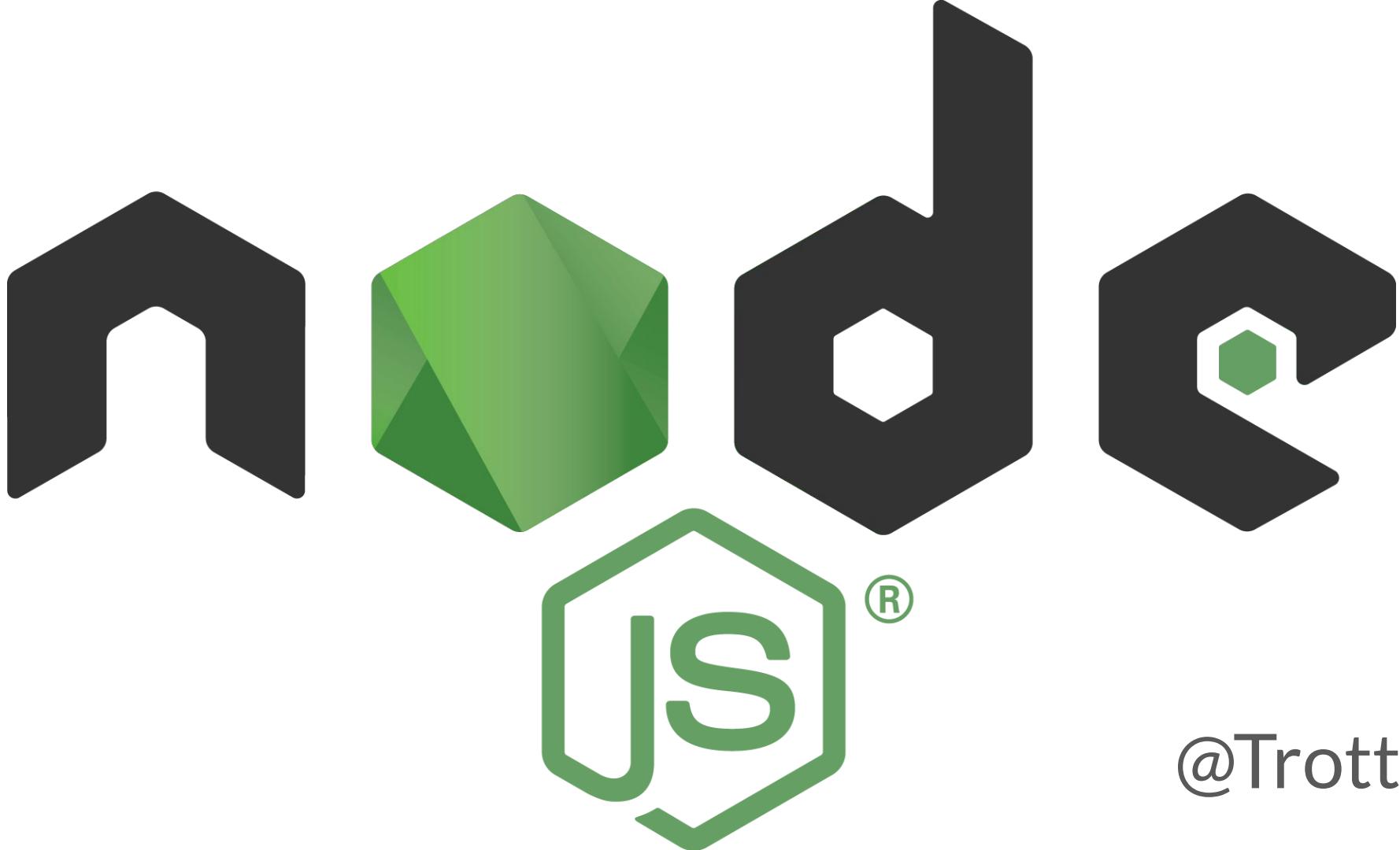


Photo by Braden Collum on Unsplash

@Trott

UCSF Library

@Trott



@Trott

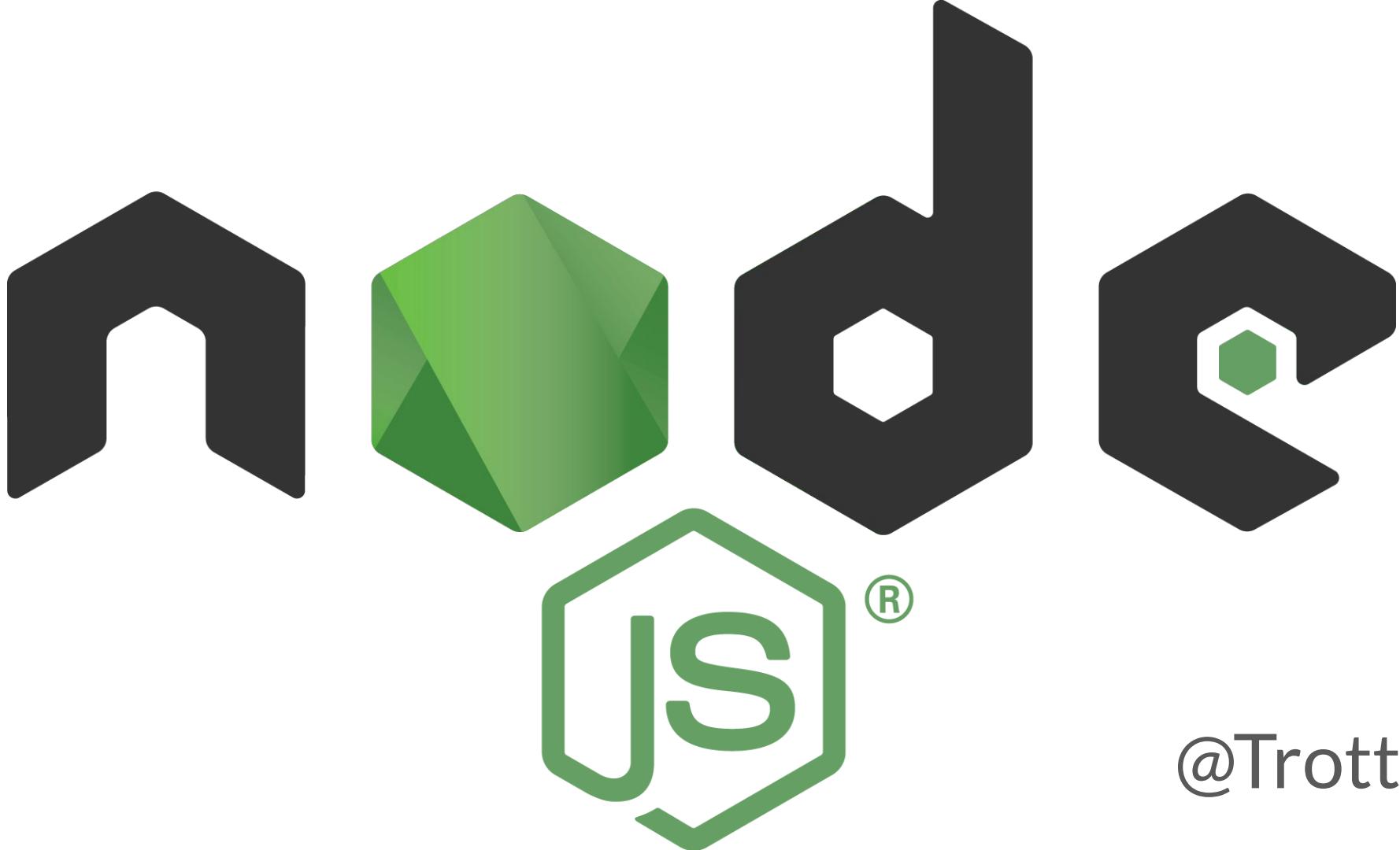


PALACE FAMILY STEAK HOUSE

@Trott

<https://www.PalaceFamilySteakHouse.com/>

@Trott



@Trott

Disclaimers

@Trott

The views expressed are
my own and not
necessarily those of my
employer.

@Trott

The views expressed are
my own and not
necessarily those of
Node.js.

@Trott



@Trott

worker_threads

@Trott

Node.js 10.5.0

(with a command line flag)

@Trott

Node.js 12

(without a command line flag)

(because Node.js 11.x is not supported anymore)

@Trott

Node.js 12.11.0

worker_threads are stable!

@Trott

Node.js 12.15.0

@Trott

Node.js 13.8.0

@Trott

WTF are they?

@Trott

Web Workers

@Trott

threads

@Trott

JavaScript is
single-threaded

@Trott

JavaScript is
single-threaded
(some people argue this)

@Trott

```
var keepGoing = true

setTimeout(() => { keepGoing = false }, 100)

while (keepGoing) {
  console.log('Keep going!')
}
```

@Trott

Blocking the Event Loop

@Trott



► 26:53

What the heck is the event loop anyway? | Philip Roberts | JSConf EU ...

YouTube · JSConf

@Trott

<https://www.PalaceFamilySteakHouse.com/>

@Trott

Asynchronous

@Trott

Asynchronous I/O

@Trott

CPU, not so much

@Trott

cluster

@Trott

Multiple processes

@Trott

Lightweight

@Trott

Data sharing

@Trott

Hello, threads!

@Trott

```
const { Worker, isMainThread, parentPort } = require('worker_threads');
if (isMainThread) {
    // This code is executed in the main thread and not in the worker.

    // Create the worker.
    const worker = new Worker(__filename);
    // Listen for messages from the worker and print them.
    worker.on('message', (msg) => { console.log(msg); });
} else {
    // This code is executed in the worker and not in the main thread.

    // Send a message to the main thread.
    parentPort.postMessage('Hello world!');
}
```

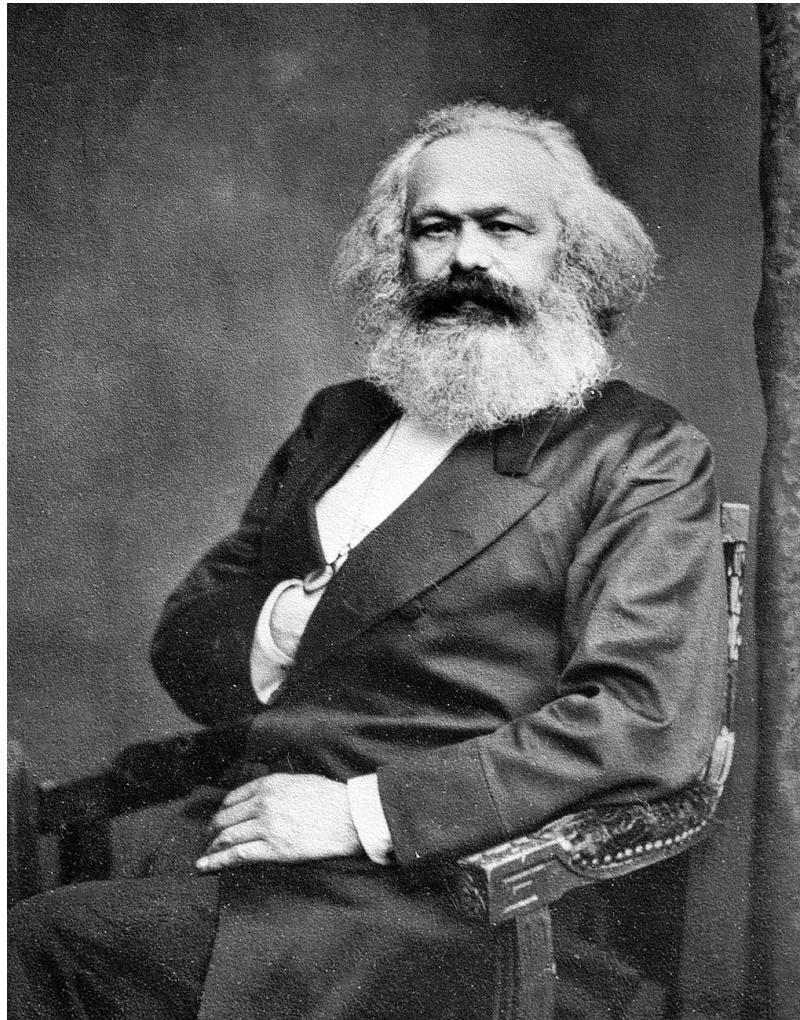
@Trott

```
const { Worker, isMainThread, parentPort } = require('worker_threads');
if (isMainThread) {
    // This code is executed in the main thread and not in the worker.

    // Create the worker.
    const worker = new Worker(__filename);
    // Listen for messages from the worker and print them.
    worker.on('message', (msg) => { console.log(msg); });
} else {
    // This code is executed in the worker and not in the main thread.

    // Send a message to the main thread.
    parentPort.postMessage('Hello world!');
}
```

@Trott



@Trott

```
const { Worker, isMainThread, parentPort } = require('worker_threads');
if (isMainThread) {
    // This code is executed in the main thread and not in the worker.

    // Create the worker.
    const worker = new Worker(__filename);
    // Listen for messages from the worker and print them.
    worker.on('message', (msg) => { console.log(msg); });
} else {
    // This code is executed in the worker and not in the main thread.

    // Send a message to the main thread.
    parentPort.postMessage('Hello world!');
}
```

@Trott

```
const { Worker, isMainThread, parentPort } = require('worker_threads');
if (isMainThread) {
    // This code is executed in the main thread and not in the worker.

    // Create the worker.
    const worker = new Worker(__filename);
    // Listen for messages from the worker and print them.
    worker.on('message', (msg) => { console.log(msg); });
} else {
    // This code is executed in the worker and not in the main thread.

    // Send a message to the main thread.
    parentPort.postMessage('Hello world!');
}
```

@Trott

```
const { Worker, isMainThread, parentPort } = require('worker_threads');
if (isMainThread) {
    // This code is executed in the main thread and not in the worker.

    // Create the worker.
    const worker = new Worker(__filename);
    // Listen for messages from the worker and print them.
    worker.on('message', (msg) => { console.log(msg); });
} else {
    // This code is executed in the worker and not in the main thread.

    // Send a message to the main thread.
    parentPort.postMessage('Hello world!');
}
```



@Trott

```
const { Worker, isMainThread, parentPort } = require('worker_threads');
if (isMainThread) {
    // This code is executed in the main thread and not in the worker.

    // Create the worker.
    const worker = new Worker(__filename);
    // Listen for messages from the worker and print them.
    worker.on('message', (msg) => { console.log(msg); });
} else {
    // This code is executed in the worker and not in the main thread.

    // Send a message to the main thread.
    parentPort.postMessage('Hello world!');
}
```



@Trott

```
const { Worker, isMainThread, parentPort } = require('worker_threads');
if (isMainThread) {
    // This code is executed in the main thread and not in the worker.

    // Create the worker.
    const worker = new Worker(__filename);
    // Listen for messages from the worker and print them.
    worker.on('message', (msg) => { console.log(msg); });
} else {
    // This code is executed in the worker and not in the main thread.

    // Send a message to the main thread.
    parentPort.postMessage('Hello world!');
}
```

@Trott

```
const { Worker, isMainThread, parentPort } = require('worker_threads');
if (isMainThread) {
    // This code is executed in the main thread and not in the worker.

    // Create the worker.
    const worker = new Worker(__filename);
    // Listen for messages from the worker and print them.
    worker.on('message', (msg) => { console.log(msg); });
} else {
    // This code is executed in the worker and not in the main thread.

    // Send a message to the main thread.
    parentPort.postMessage('Hello world!');
}
```



@Trott

```
const { Worker, isMainThread, parentPort } = require('worker_threads');
if (isMainThread) {
    // This code is executed in the main thread and not in the worker.

    // Create the worker.
    const worker = new Worker(__filename);
    // Listen for messages from the worker and print them.
    worker.on('message', (msg) => { console.log(msg); });
} else {
    // This code is executed in the worker and not in the main thread.

    // Send a message to the main thread.
    parentPort.postMessage('Hello world!');
}
```

@Trott

Six Degrees of Kevin Bacon

@Trott

Katy Perry

@Trott

Katy Perry -> Zoolander 2 -> John Malkovich

@Trott

Katy Perry -> Zoolander 2 -> John Malkovich
John Malkovich -> *Queens Logic* -> Kevin Bacon

@Trott

Music Routes

@Trott

What musicians play on what tracks?

@Trott

AllMusic

@Trott

MusicBrainz

@Trott

Discogs

@Trott

Wikidata

@Trott

<https://github.com/Trott/music-routes-data>

@Trott

<https://trott.github.io/music-routes-visualization/>

@Trott

Breadth-First Search

@Trott

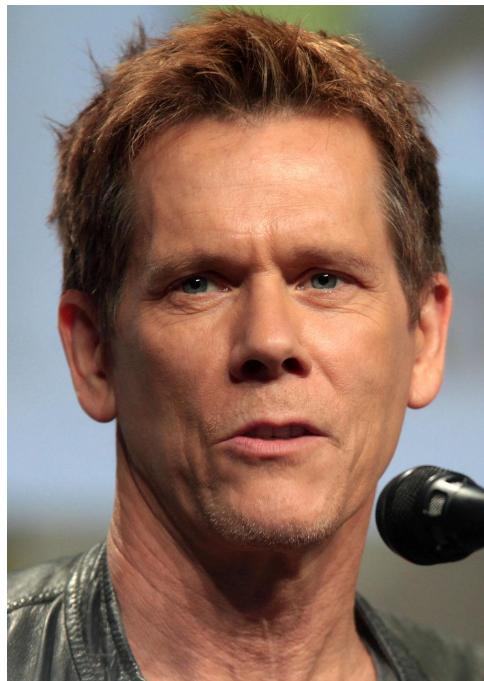
Connect Katy Perry to Kevin Bacon

@Trott

Are Katy Perry and Kevin Bacon the same person?



====



Katy Perry photo: © Glenn Francis, www.PacificProDigital.com
Kevin Bacon photo: Gage Skidmore

@Trott

NO!!!

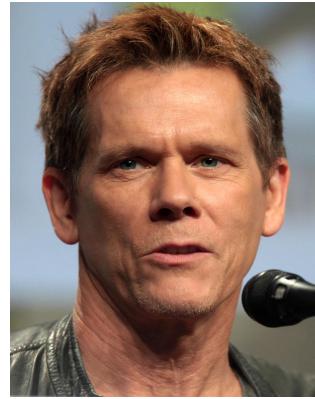
@Trott



+



includes



Find everyone that was in a movie with Katy Perry.

@Trott



+

 +


includes



Find everyone that was in a movie with anyone
that was in a movie with Katy Perry.

@Trott

The Worst!

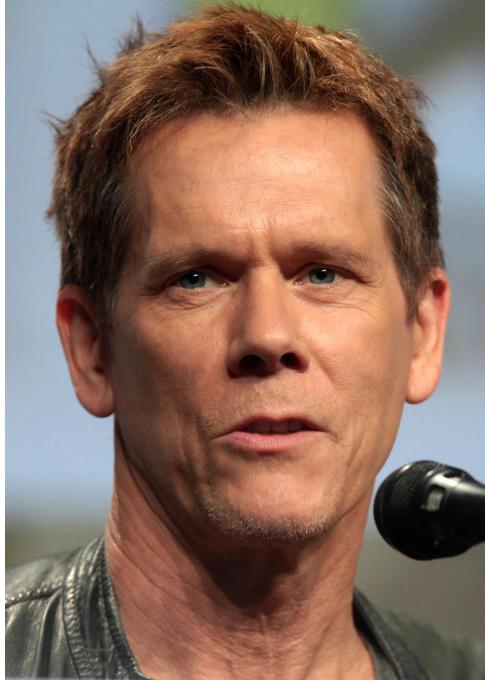
@Trott

Next: The Second Worst!

@Trott



<=>



Katy Perry photo: © Glenn Francis, www.PacificProDigital.com
Kevin Bacon photo: Gage Skidmore

@Trott

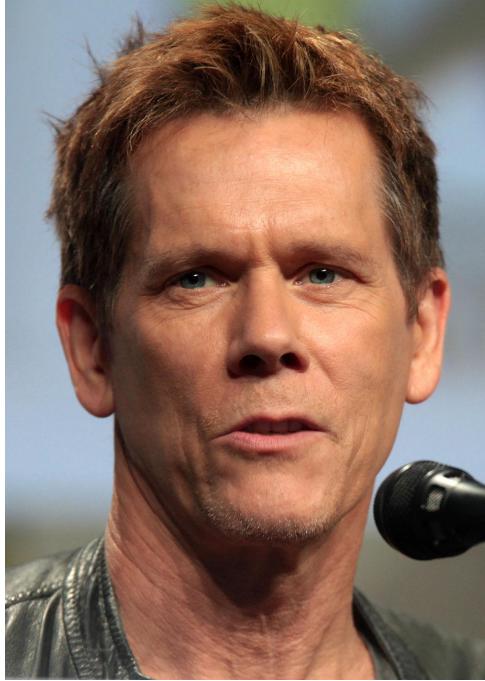


Photo: Wikipedia, attributed to "Author"

@Trott



<=>



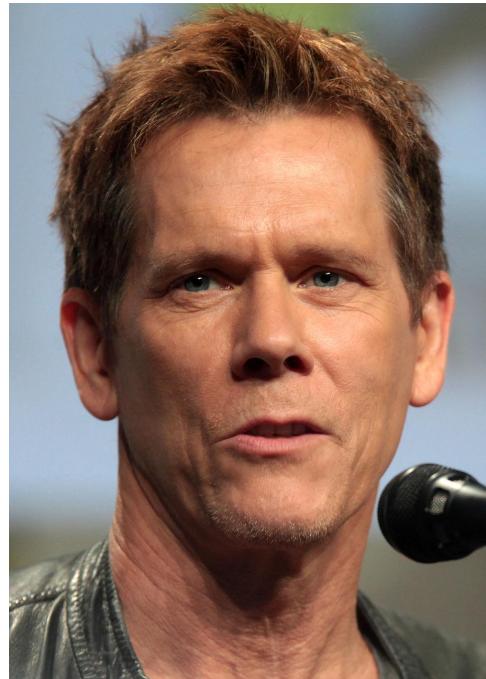
Katy Perry photo: © Glenn Francis, www.PacificProDigital.com
Kevin Bacon photo: Gage Skidmore

@Trott

Are Katy Perry and Kevin Bacon the same person?



====



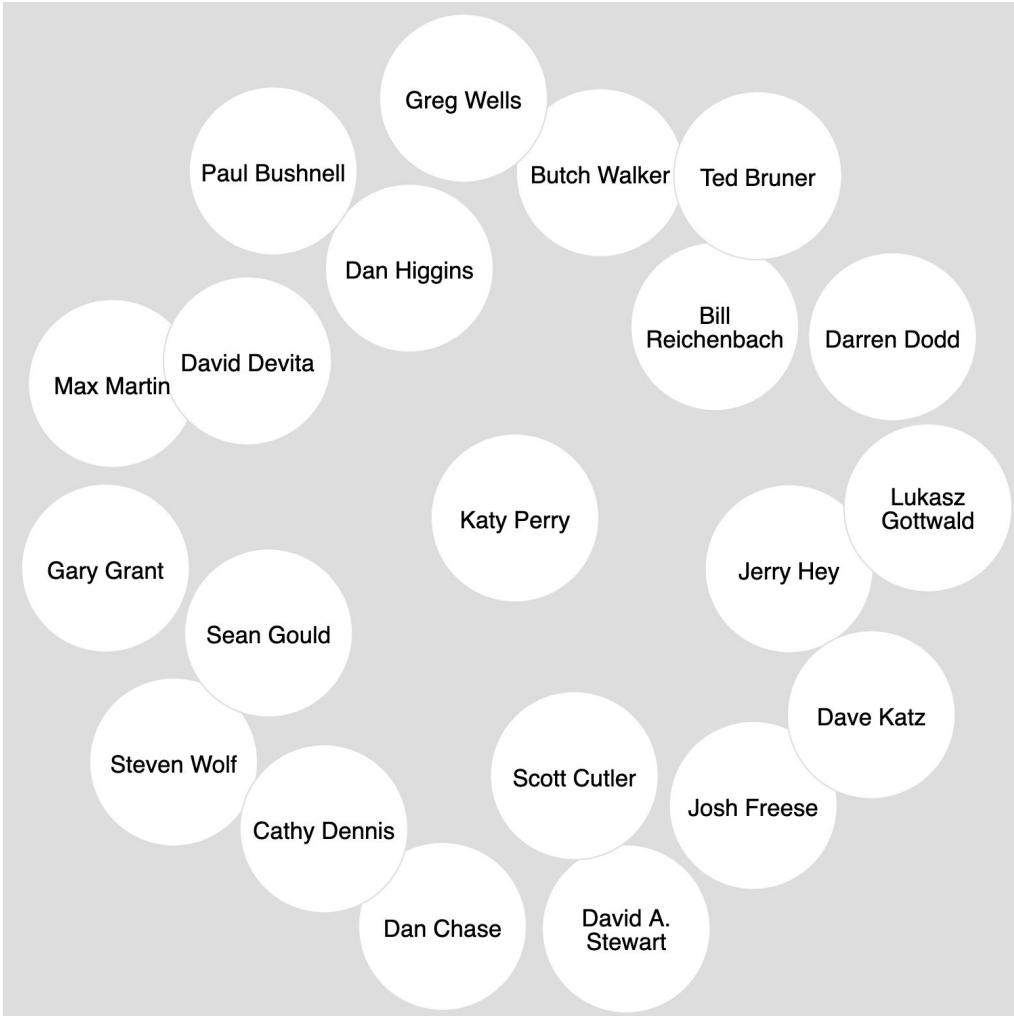
Katy Perry photo: © Glenn Francis, www.PacificProDigital.com

Kevin Bacon photo: Gage Skidmore

@Trott

NO!!!

@Trott



@Trott

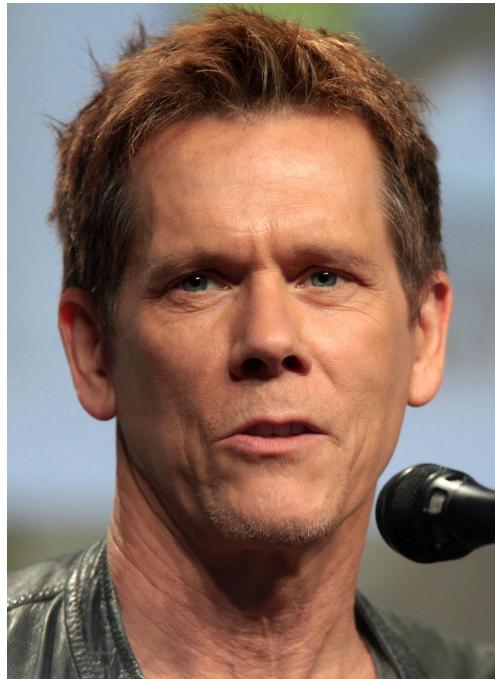


Photo: Gage Skidmore

@Trott



@Trott



@Trott

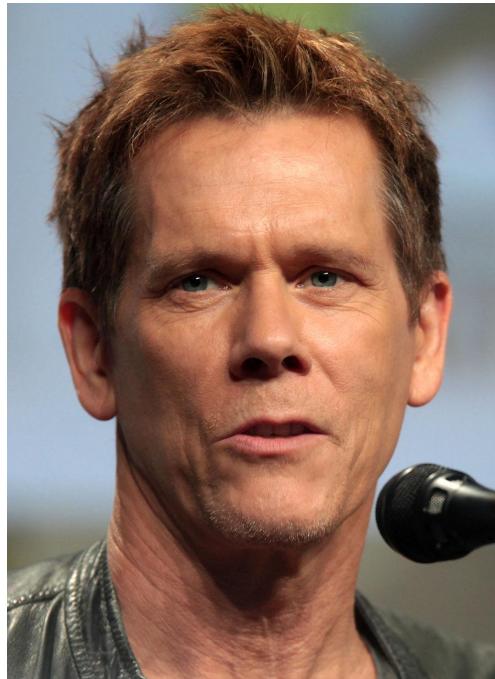


Photo: Gage Skidmore

@Trott

Katy Perry played on "I'm Still Breathing" with David A. Stewart

David A. Stewart played on "Queen Of New Orleans" with Jon Bon Jovi

Jon Bon Jovi played on "Boys In Bars" with Kevin Bacon

@Trott

Breadth-First Search

@Trott

Let's implement a
Breadth-First Search

@Trott

Let's not implement a
Breadth-First Search

@Trott

<https://github.com/Trott/music-routing>

@Trott

```
// The index in these arrays represent how many steps away from the start
// individual. Initialize at zero steps.
startTracksBfsResults[0] = getTracksForIndividual(startId)
startIndividualsBfsResults[0] = new Set([startId])

// The index here is how many steps away from the target individual.
targetTracksBfsResults[0] = getTracksForIndividual(targetId)
targetIndividualsBfsResults[0] = new Set([targetId])

// Check if the two individuals have any tracks in common.
let matches = matchFound(startTracksBfsResults, targetTracksBfsResults)

// Keep getting more individuals/tracks until there is a match.
while (!matches.length) {
  getNextBfsStepResults(startTracksBfsResults, startIndividualsBfsResults)
  matches = matchFound(startTracksBfsResults, targetTracksBfsResults)
}
```

@Trott

```
// The index in these arrays represent how many steps away from the start
// individual. Initialize at zero steps.
startTracksBfsResults[0] = getTracksForIndividual(startId)
startIndividualsBfsResults[0] = new Set([startId])

// The index here is how many steps away from the target individual.
targetTracksBfsResults[0] = getTracksForIndividual(targetId)
targetIndividualsBfsResults[0] = new Set([targetId])

// Check if the two individuals have any tracks in common.
let matches = matchFound(startTracksBfsResults, targetTracksBfsResults)

// Keep getting more individuals/tracks until there is a match.
while (!matches.length) {
  getNextBfsStepResults(startTracksBfsResults, startIndividualsBfsResults)
  matches = matchFound(startTracksBfsResults, targetTracksBfsResults)
}
```



@Trott

```
// The index in these arrays represent how many steps away from the start
// individual. Initialize at zero steps.
startTracksBfsResults[0] = getTracksForIndividual(startId)
startIndividualsBfsResults[0] = new Set([startId])

// The index here is how many steps away from the target individual.
targetTracksBfsResults[0] = getTracksForIndividual(targetId)
targetIndividualsBfsResults[0] = new Set([targetId])

// Check if the two individuals have any tracks in common.
let matches = matchFound(startTracksBfsResults, targetTracksBfsResults)

// Keep getting more individuals/tracks until there is a match.
while (!matches.length) {
  getNextBfsStepResults(startTracksBfsResults, startIndividualsBfsResults)
  matches = matchFound(startTracksBfsResults, targetTracksBfsResults)
}
```



@Trott

```
// The index in these arrays represent how many steps away from the start
// individual. Initialize at zero steps.
startTracksBfsResults[0] = getTracksForIndividual(startId)
startIndividualsBfsResults[0] = new Set([startId])

// The index here is how many steps away from the target individual.
targetTracksBfsResults[0] = getTracksForIndividual(targetId)
targetIndividualsBfsResults[0] = new Set([targetId])

// Check if the two individuals have any tracks in common.
let matches = matchFound(startTracksBfsResults, targetTracksBfsResults)

// Keep getting more individuals/tracks until there is a match.
while (!matches.length) {
  getNextBfsStepResults(startTracksBfsResults, startIndividualsBfsResults)
  matches = matchFound(startTracksBfsResults, targetTracksBfsResults)
}
```



@Trott

```
// The index in these arrays represent how many steps away from the start
// individual. Initialize at zero steps.
startTracksBfsResults[0] = getTracksForIndividual(startId)
startIndividualsBfsResults[0] = new Set([startId])

// The index here is how many steps away from the target individual.
targetTracksBfsResults[0] = getTracksForIndividual(targetId)
targetIndividualsBfsResults[0] = new Set([targetId])

// Check if the two individuals have any tracks in common.
let matches = matchFound(startTracksBfsResults, targetTracksBfsResults)

// Keep getting more individuals/tracks until there is a match.
while (!matches.length) {
  getNextBfsStepResults(startTracksBfsResults, startIndividualsBfsResults)
  matches = matchFound(startTracksBfsResults, targetTracksBfsResults)
}
```



@Trott

```
// The index in these arrays represent how many steps away from the start
// individual. Initialize at zero steps.
startTracksBfsResults[0] = getTracksForIndividual(startId)
startIndividualsBfsResults[0] = new Set([startId])

// The index here is how many steps away from the target individual.
targetTracksBfsResults[0] = getTracksForIndividual(targetId)
targetIndividualsBfsResults[0] = new Set([targetId])

// Check if the two individuals have any tracks in common.
let matches = matchFound(startTracksBfsResults, targetTracksBfsResults)

// Keep getting more individuals/tracks until there is a match.
while (!matches.length) {
  getNextBfsStepResults(startTracksBfsResults, startIndividualsBfsResults)
  matches = matchFound(startTracksBfsResults, targetTracksBfsResults)
}
```



@Trott

```
// The index in these arrays represent how many steps away from the start
// individual. Initialize at zero steps.
startTracksBfsResults[0] = getTracksForIndividual(startId)
startIndividualsBfsResults[0] = new Set([startId])

// The index here is how many steps away from the target individual.
targetTracksBfsResults[0] = getTracksForIndividual(targetId)
targetIndividualsBfsResults[0] = new Set([targetId])

// Check if the two individuals have any tracks in common.
let matches = matchFound(startTracksBfsResults, targetTracksBfsResults)

// Keep getting more individuals/tracks until there is a match.
while (!matches.length) {
    getNextBfsStepResults(startTracksBfsResults, startIndividualsBfsResults)
    matches = matchFound(startTracksBfsResults, targetTracksBfsResults)
}
```



@Trott

```
// The index in these arrays represent how many steps away from the start
// individual. Initialize at zero steps.
startTracksBfsResults[0] = getTracksForIndividual(startId)
startIndividualsBfsResults[0] = new Set([startId])

// The index here is how many steps away from the target individual.
targetTracksBfsResults[0] = getTracksForIndividual(targetId)
targetIndividualsBfsResults[0] = new Set([targetId])

// Check if the two individuals have any tracks in common.
let matches = matchFound(startTracksBfsResults, targetTracksBfsResults)

// Keep getting more individuals/tracks until there is a match.
while (!matches.length) {
  getNextBfsStepResults(startTracksBfsResults, startIndividualsBfsResults)
  matches = matchFound(startTracksBfsResults, targetTracksBfsResults)
}
```



@Trott



@Trott

```
$ node bfs-single-thread.js
```

```
search duration: 14022.502ms
```

```
Aretha Franklin played on "Testimony" with Céline Dion
```

```
Céline Dion played on "We Are The World 25 For Haiti" with Jason Mraz
```

```
Jason Mraz played on "Did You Get My Message?" with Rachael Yamagata
```

```
Rachael Yamagata played on "Clairaudients (Kill Or Be Killed)" with Janet Weiss
```

```
Janet Weiss played on "Endless Talk" with Carrie Brownstein
```

```
$
```

@Trott

Bidirectional Search

@Trott



Photo by Charles on Unsplash

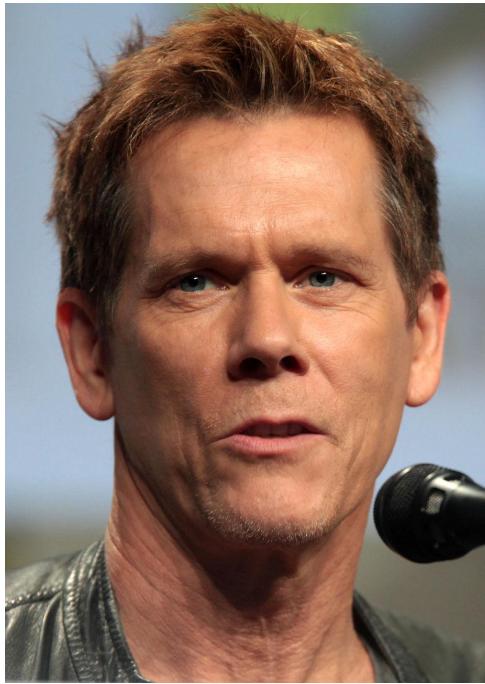
@Trott

Bidirectional Search

@Trott



====



Katy Perry photo: © Glenn Francis, www.PacificProDigital.com
Kevin Bacon photo: Gage Skidmore

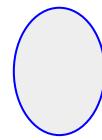
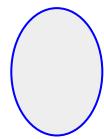
@Trott



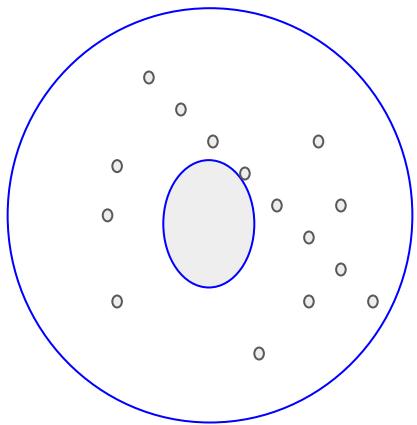
@Trott



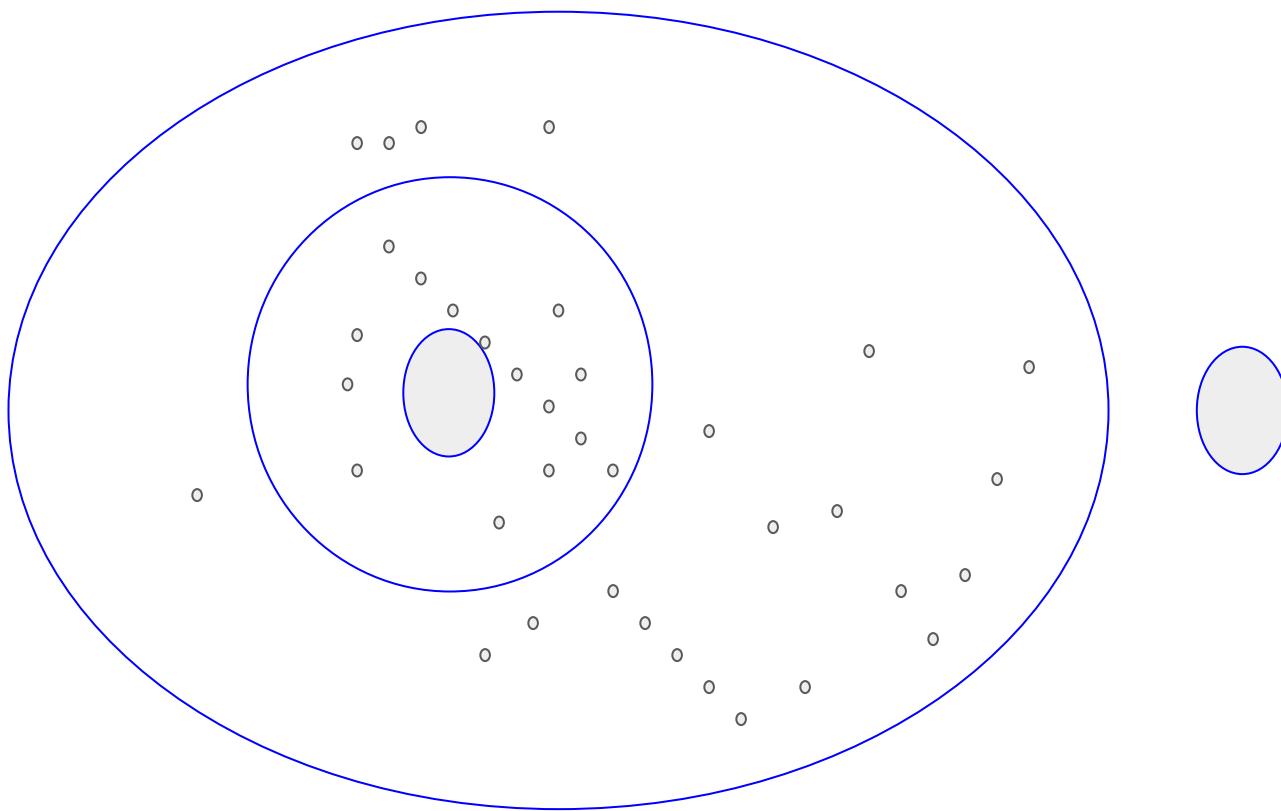
@Trott



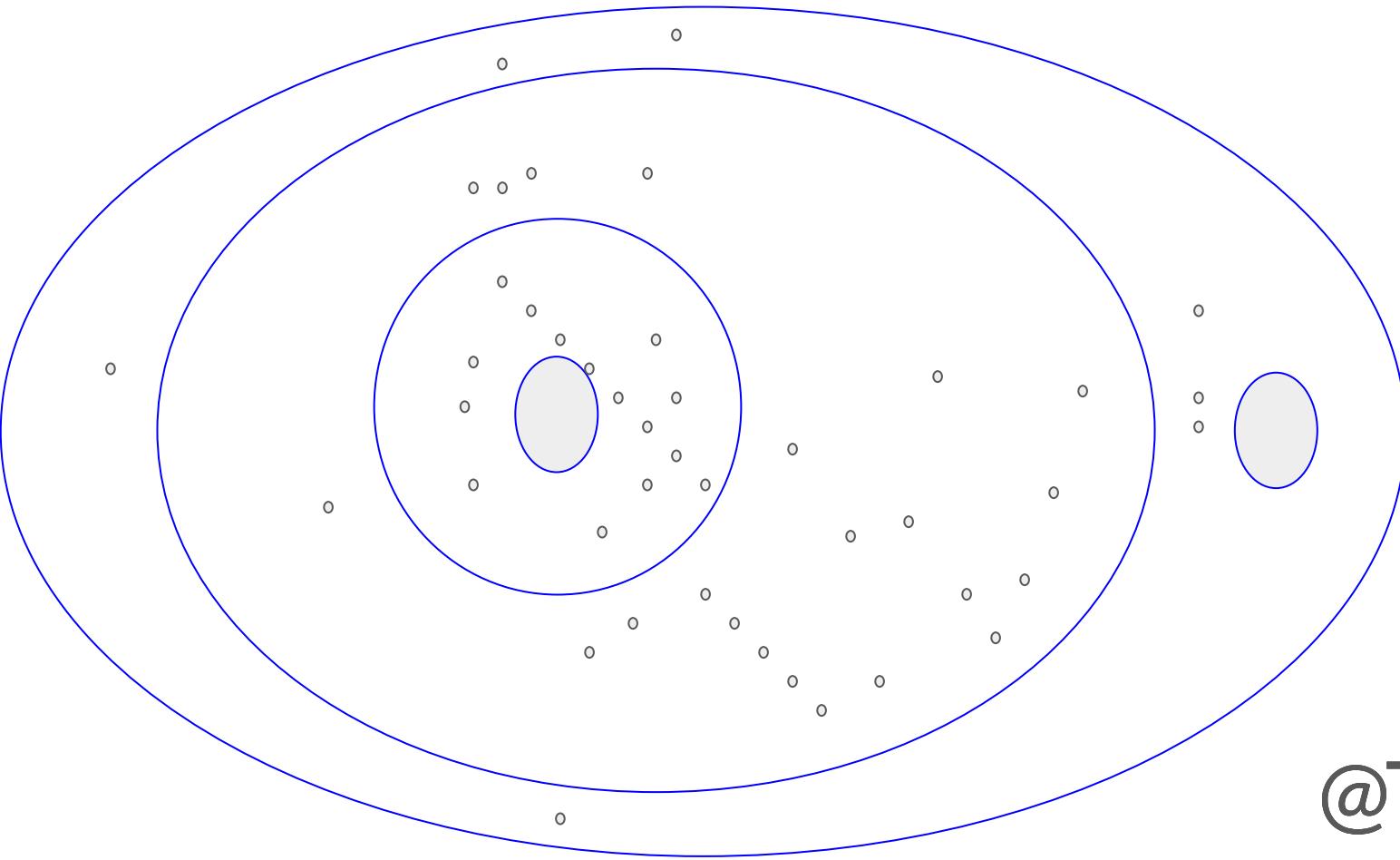
@Trott



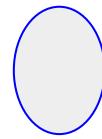
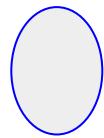
@Trott



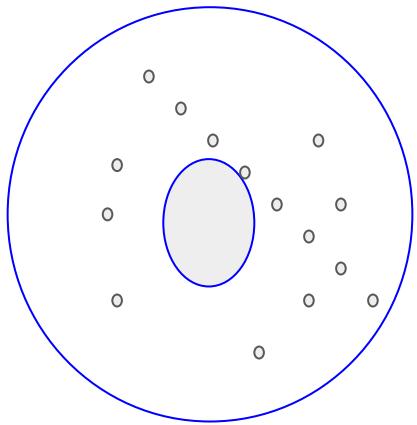
@Trott



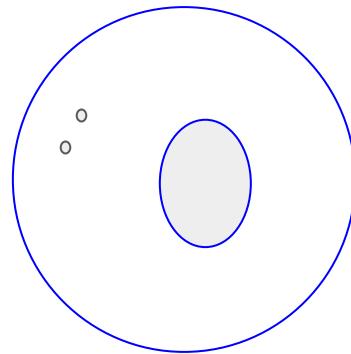
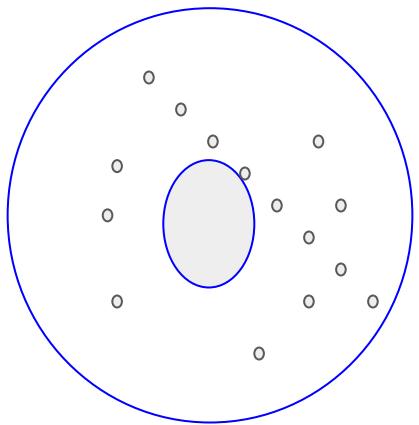
@Trott



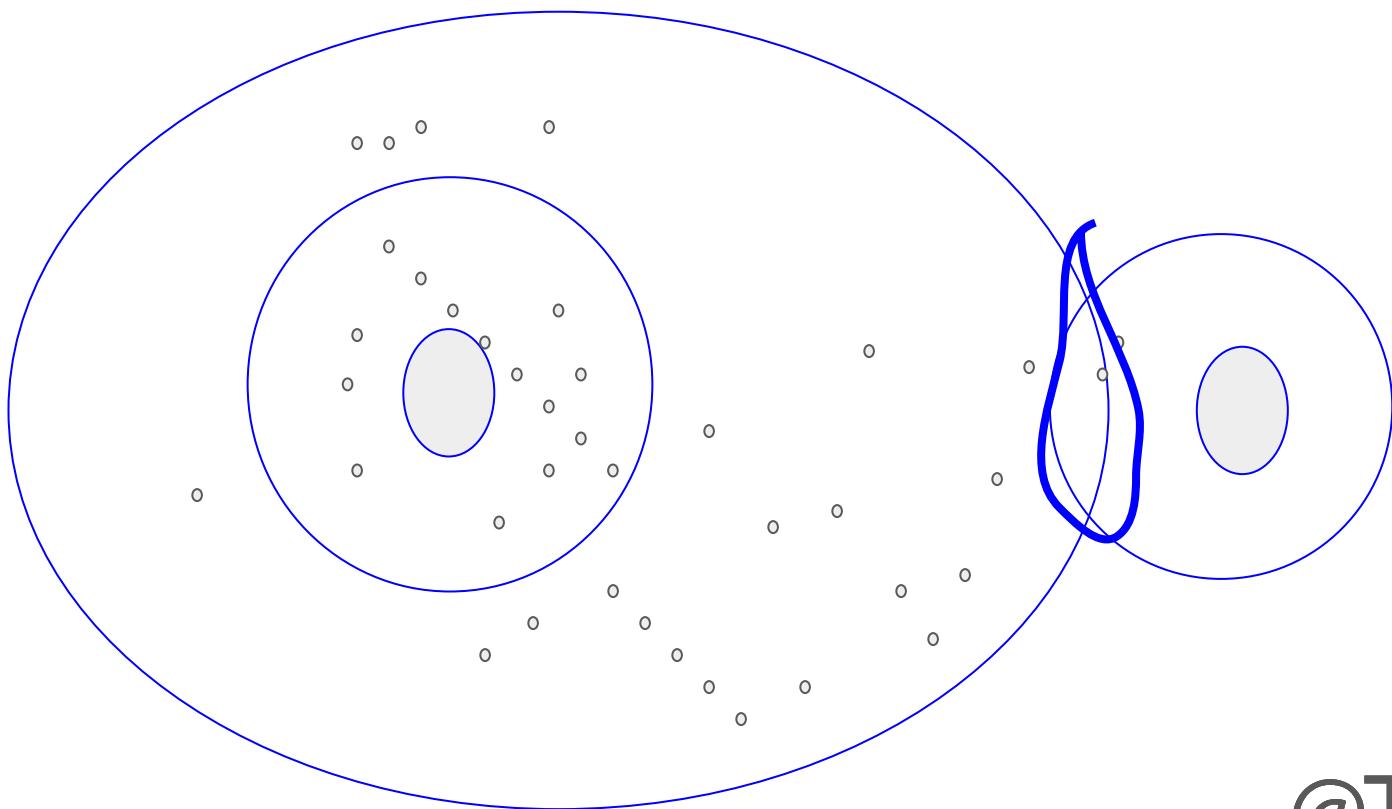
@Trott



@Trott



@Trott



@Trott

https://en.wikipedia.org/wiki/Bidirectional_search

@Trott

```
// The index in these arrays represent how many steps away from the start
// individual. Initialize at zero steps.
startTracksBfsResults[0] = getTracksForIndividual(id1)
const targetTracksBfsResults = [] set([id1])
const targetTracksBfsResults: any[] way from the target individual.
targetTracksBfsResults[0] = getTracksForIndividual(id2)
targetIndividualsBfsResults[0] = new Set([id2])

// Check if the two individuals have any tracks in common.
let matches = matchFound(startTracksBfsResults, targetTracksBfsResults)

// Keep getting more individuals/tracks until there is a match.
while (!matches.length) {
  getNextBfsStepResults(startTracksBfsResults, startIndividualsBfsResults)
  matches = matchFound(startTracksBfsResults, targetTracksBfsResults)

  if (matches.length) {
    continue
  }
  getNextBfsStepResults(targetTracksBfsResults, targetIndividualsBfsResults)
  matches = matchFound(startTracksBfsResults, targetTracksBfsResults)
}
```

@Trott

```
// Keep getting more individuals/tracks until there is a match.  
while (!matches.length) {  
    getNextBfsStepResults(startTracksBfsResults, startIndividualsBfsResults)  
    matches = matchFound(startTracksBfsResults, targetTracksBfsResults)  
  
    if (matches.length) {  
        continue  
    }  
    getNextBfsStepResults(targetTracksBfsResults, targetIndividualsBfsResults)  
    matches = matchFound(startTracksBfsResults, targetTracksBfsResults)  
}
```

@Trott

```
$ node bidirectional-single-thread.js
```

```
search duration: 2821.076ms
```

```
Aretha Franklin played on "Jumpin' Jack Flash" with Randy Jackson
```

```
Randy Jackson played on "We Are The World 25 For Haiti" with Jason Mraz
```

```
Jason Mraz played on "Did You Get My Message?" with Rachael Yamagata
```

```
Rachael Yamagata played on "Clairaudients (Kill Or Be Killed)" with Janet Weiss
```

```
Janet Weiss played on "Dance Song '97" with Carrie Brownstein
```

```
$
```

@Trott

Bidirectional search with worker_threads!

@Trott

main.js:

```
function createWorker (id, index) {
  const worker = new Worker('./worker.js', { workerData: { id } })
  worker.on('error', (err) => { throw err })
  worker.on('message', callback.bind(worker, index))
  return worker
}
```



@Trott



Photo by Herson Rodriguez on Unsplash

@Trott



Photo by Etienne Girardet on Unsplash

@Trott

main.js:

```
function createWorker (id, index) {
  const worker = new Worker('./worker.js', { workerData: { id } })
  worker.on('error', (err) => { throw err })
  worker.on('message', callback.bind(worker, index))
  return worker
}
```

worker.js:

```
const { parentPort, workerData } = require('worker_threads')
const { id } = workerData
```

@Trott

main.js:

```
function createWorker (id, index) {
  const worker = new Worker('./worker.js', { workerData: { id } })
  worker.on('error', (err) => { throw err })
  worker.on('message', callback.bind(worker, index))
  return worker
}
```

worker.js:

```
const { parentPort, workerData } = require('worker_threads')
const { id } = workerData
```



@Trott

main.js:

```
function createWorker (id, index) {
  const worker = new Worker('./worker.js', { workerData: { id } })
  worker.on('error', (err) => { throw err })
  worker.on('message', callback.bind(worker, index))
  return worker
}
```



worker.js:

```
const { parentPort, workerData } = require('worker_threads')
const { id } = workerData
```

@Trott

main.js:

```
function createWorker (id, index) {
  const worker = new Worker('./worker.js', { workerData: { id } })
  worker.on('error', (err) => { throw err })
  worker.on('message', callback.bind(worker, index))
  return worker
}
```



worker.js:

```
const { parentPort, workerData } = require('worker_threads')
const { id } = workerData
```

@Trott

main.js:

```
function createWorker (id, index) {
  const worker = new Worker('./worker.js', { workerData: { id } })
  worker.on('error', (err) => { throw err })
  worker.on('message', callback.bind(worker, index))
  return worker
}
```

worker.js:

```
→ tracks[0] = getTracksForIndividual(id)
individuals[0] = new Set([id])
parentPort.postMessage({ tracks, individuals })
```

@Trott

main.js:

```
function createWorker (id, index) {
  const worker = new Worker('./worker.js', { workerData: { id } })
  worker.on('error', (err) => { throw err })
  worker.on('message', callback.bind(worker, index))
  return worker
}
```

worker.js:

```
tracks[0] = getTracksForIndividual(id)
individuals[0] = new Set([id])
parentPort.postMessage({ tracks, individuals })
```

@Trott

main.js:

```
function createWorker (id, index) {
  const worker = new Worker('./worker.js', { workerData: { id } })
  worker.on('error', (err) => { throw err })
  worker.on('message', callback.bind(worker, index))
  return worker
}
```

worker.js:

```
tracks[0] = getTracksForIndividual(id)
individuals[0] = new Set([id])
```

```
parentPort.postMessage({ tracks, individuals })
```

@Trott

main.js:

```
function createWorker (id, index) {
  const worker = new Worker('./worker.js', { workerData: { id } })
  worker.on('error', (err) => { throw err })
  worker.on('message', callback.bind(worker, index))
  return worker
}
```



worker.js:

```
tracks[0] = getTracksForIndividual(id)
individuals[0] = new Set([id])

parentPort.postMessage({ tracks, individuals })
```

@Trott

main.js:

```
→ function callback (index, data) {  
    tracks[index] = data.tracks  
    individuals[index] = data.individuals  
    matches = matchFound(tracks[0], tracks[1])  
    if (matches.length) {  
        done()  
    } else {  
        this.postMessage('next')  
    }  
}
```

@Trott

main.js:

```
function callback (index, data) {
  tracks[index] = data.tracks
  individuals[index] = data.individuals
  matches = matchFound(tracks[0], tracks[1])
  if (matches.length) {
    | done()
  } else {
    | this.postMessage('next')
  }
}
```



@Trott

main.js:

```
function callback (index, data) {
  tracks[index] = data.tracks
  individuals[index] = data.individuals
  matches = matchFound(tracks[0], tracks[1])
  if (matches.length) {
    | done()
  } else {
    | this.postMessage('next')
  }
}
```



@Trott

main.js:

```
function callback (index, data) {
  tracks[index] = data.tracks
  individuals[index] = data.individuals
  matches = matchFound(tracks[0], tracks[1])
  if (matches.length) {
    | done()
  } else {
    | this.postMessage('next')
  }
}
```



@Trott

main.js:

```
function callback (index, data) {
  tracks[index] = data.tracks
  individuals[index] = data.individuals
  matches = matchFound(tracks[0], tracks[1])
  if (matches.length) {
    | done()
  } else {
    | this.postMessage('next')
  }
}
```



@Trott

main.js:

```
function callback (index, data) {
  tracks[index] = data.tracks
  individuals[index] = data.individuals
  matches = matchFound(tracks[0], tracks[1])
  if (matches.length) {
    | done()
  } else {
    | this.postMessage('next')
  }
}
```



@Trott

main.js:

```
function callback (index, data) {  
    tracks[index] = data.tracks  
    individuals[index] = data.individuals  
    matches = matchFound(tracks[0], tracks[1])  
    if (matches.length) {  
        done()  
    } else {  
        this.postMessage('next')  
    }  
}
```



@Trott

main.js:

```
→ function done () {
    console.timeEnd('search duration')
    workerForStart.removeListener('message', callback)
    workerForTarget.removeListener('message', callback)
    workerForStart.terminate()
    workerForTarget.terminate()
    printResults()
}
```

@Trott

main.js:

```
function done () {  
    console.timeEnd('search duration')  
    workerForStart.removeListener('message', callback)  
    workerForTarget.removeListener('message', callback)  
    workerForStart.terminate()  
    workerForTarget.terminate()  
    printResults()  
}
```



@Trott

main.js:

```
function done () {  
    console.timeEnd('search duration')  
    workerForStart.removeListener('message', callback)  
    workerForTarget.removeListener('message', callback)  
    workerForStart.terminate()  
    workerForTarget.terminate()  
    printResults()  
}
```



@Trott

main.js:

```
function done () {  
    console.timeEnd('search duration')  
    workerForStart.removeListener('message', callback)  
    workerForTarget.removeListener('message', callback)  
    workerForStart.terminate()  
    workerForTarget.terminate()  
    printResults()  
}
```



@Trott

How does it perform?

@Trott

```
$ node bfs-single-thread.js  
search duration: 14022.502ms
```

@Trott

```
$ node bidirectional-single-thread.js  
search duration: 2821.076ms
```

@Trott

```
$ node bidirectional-with-workers.js
```

```
search duration: 653.384ms
```

```
Aretha Franklin played on "Don't Waste Your Time" with Mary J. Blige
```

```
Mary J. Blige played on "We Are The World 25 For Haiti" with Jason Mraz
```

```
Jason Mraz played on "Song For A Friend" with Rachael Yamagata
```

```
Rachael Yamagata played on "Clairaudients (Kill Or Be Killed)" with Janet Weiss
```

```
Janet Weiss played on "Black Tiles" with Carrie Brownstein
```

```
$
```

@Trott

Coming clean....

@Trott

Connect Lil Nas X to Palace Family Steak House



Lil Nas X photo: DiFronzo via Wikipedia, CC BY 2.0

@Trott

Six Degrees!

@Trott



@Trott

Lil Nas X photo: DiFronzo via Wikipedia, CC BY 2.0



@Trott

Lil Nas X photo: DiFronzo via Wikipedia, CC BY 2.0



Mary Chapin Carpenter photo by Kingkongphoto & www.celebrity-photos.com from Laurel Maryland, USA
CC BY-SA 2.0

@Trott



Photo by Curtis Hilbun, CC BY 3.0

@Trott



Dolly Parton

@DollyParton



@LilNasX twitter.com/LilNasX/status...



@Trott



@Trott

Lil Nas X played on "Old Town Road" with Billy Ray Cyrus

Billy Ray Cyrus played on "Romeo" by Dolly Parton with Mary Chapin Carpenter

Mary Chapin Carpenter played on "You Ain't Goin' Nowhere" with G.E. Smith

G.E. Smith played on "Downtown Train" with Tom Waits

Tom Waits played on "Filipino Box Spring Hog" with Chris Grady

Chris Grady played on ";Todo Es Mal!" by The Slow Poisoners with Rich Trott

Rich Trott played on "Palace Family Steak House (Theme)"

@Trott

<https://vintage-bean.glitch.me/>

@Trott

www.PalaceFamilySteakHouse.com



@Trott

Thanks!

@Trott