The Future is open

An annual open source gathering
by **Open Source Community Africa.**

#OSCAFEST22

Open Source
Festival 2022

DSAs in Software Systems.

Open Source
Festival 2022

About Me.

- Occasional writer
- Managing [at]sysdsgn on Twitter
- [at]_alternatewolf on Twitter
- **SRE at Google Cloud**

Disclaimer: I don't represent Google in this talk. These are my opinions.

Introduction.

“DSAs” is short form for
Data Structure and Algorithms.

Data Structures

E.g. Trees, LinkedLists, Graphs

A structure that helps us **organize data** in a particular way in a computer's memory.

Algorithms

E.g. Binary Search, DFS

Formally, an algorithm is a **finite sequence of instructions** executed by a computer. These instructions take in an input and produce some output.

Open Source
Festival 2022

Themes.

1

DSAs as a tool to **solve problems directly.**

2

DSAs as a way to **reason about existing systems.**

3

DSAs as a tool **for novelty.**

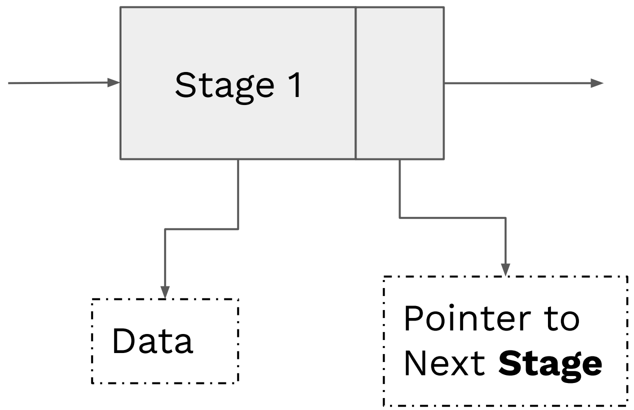
DSAs as a tool to solve problems directly.

Recruitment Platform for Non-profit.

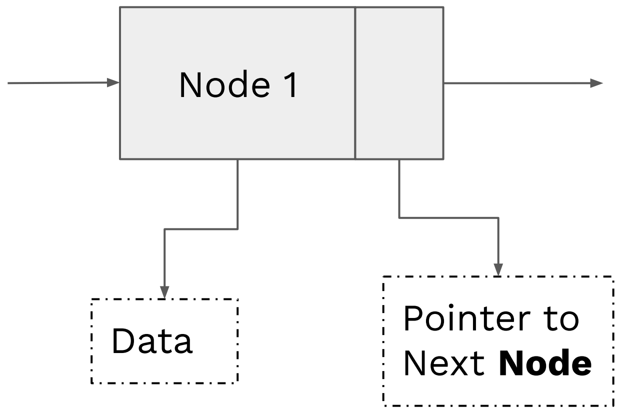
DS: LinkedList

2 years ago I worked with a team of people to build a recruitment platform for a non-profit. They interview candidates for partner companies. I wrote code that implemented recruitment stages using a LinkedList.

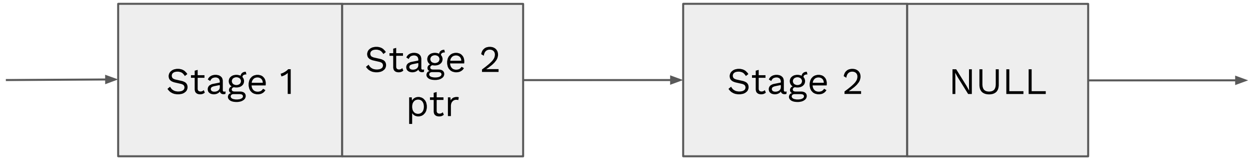
Interview Stage



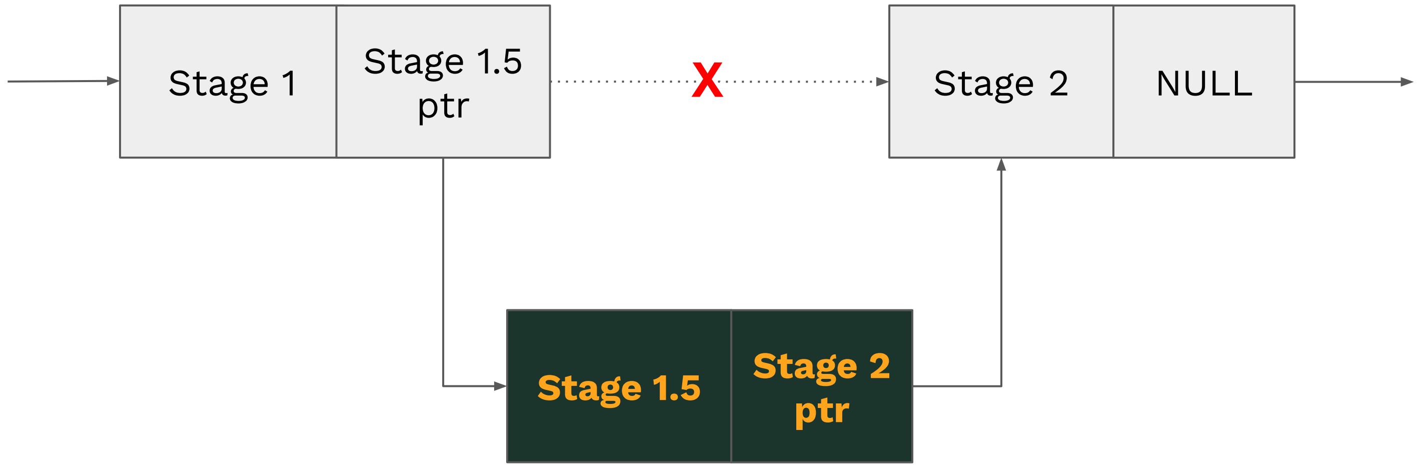
Node in LinkedList



Say, we created these interview stages...



We can insert a new stage between 1 and 2 like so...



...which is exactly how we'd insert a new node in LinkedList.

Lesson:

Knowing about the LinkedList beforehand made it easy to solve this problem. After listing my requirements, I saw that they fit perfectly into how a LinkedList stores and manipulates data.

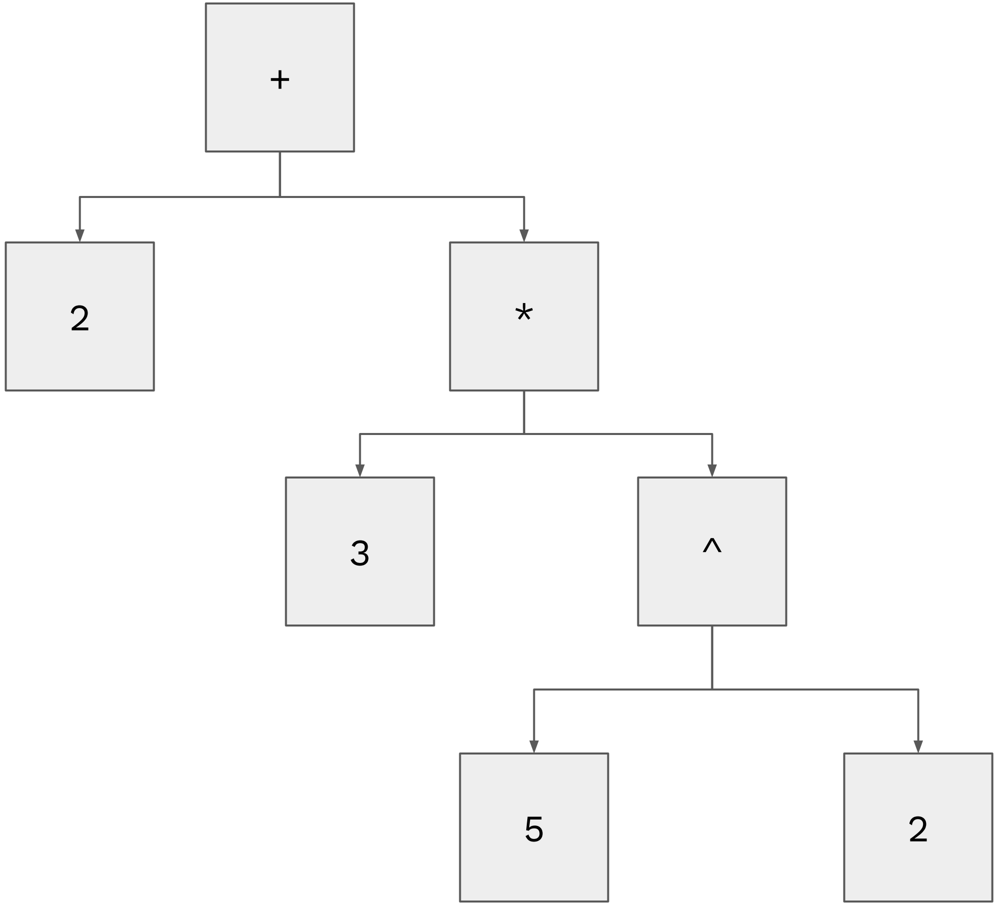
Expression Evaluation

DS: Trees; A: Post-order Traversal

I’ve been intrigued about what it’d take to build a program that can evaluate a math expression, taking precedence into account. I found trees to be a revelation.

Tree Representation for **2 + 3 * 5 ^ 2.**

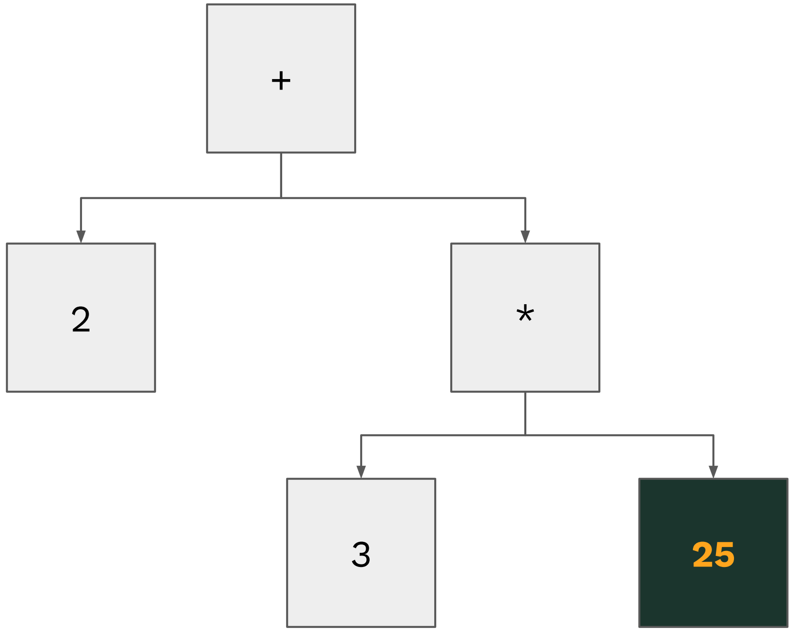
We can solve using **post-order traversal** i.e. Solve deepest nodes first.



Open Source
Festival 2022

Step 1

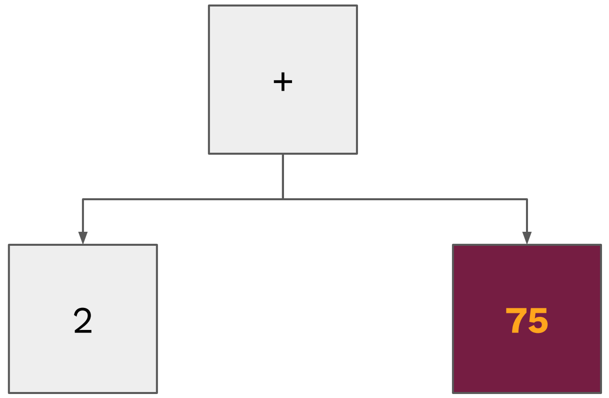
Solve **5 ^ 2** to get **25**.



Open Source
Festival 2022

Step 2

Solve **3 * 25** to get **75**.



Step 3

Solve **2 + 75** to get **77**.

Lesson:

Modeling the problem as a tree made solving this problem easy. You'll find that for a lot of problems using the right data structure makes the solution easy.

But how can you use the right data structure if you don't know it?

DSAs as a way to reason about existing systems.

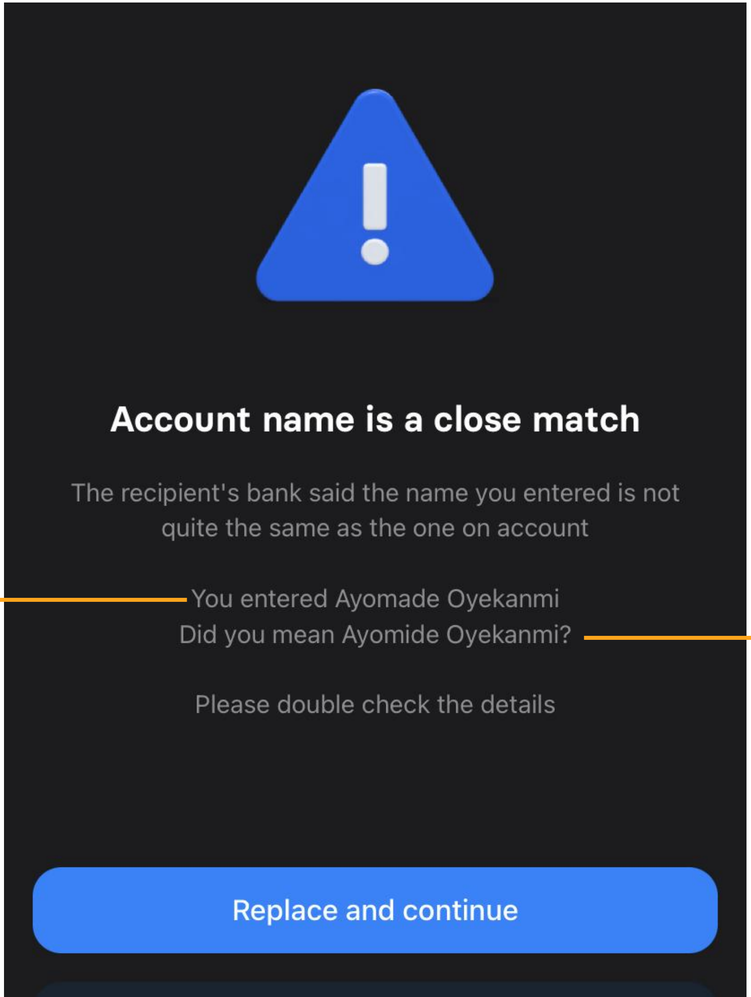
Revolut Transfer

A: Fuzzy String Matching Algorithms

To transfer money on Revolut, you need to enter account number + full name. Revolut allows you to transfer to the account when the name you input is a “close match” to the name they have stored internally for the account number.

You entered Ayomade

Did you mean Ayomide?



Account name is a close match

The recipient's bank said the name you entered is not quite the same as the one on account

You entered Ayomade Oyekanmi
Did you mean Ayomide Oyekanmi?

Please double check the details

Fuzzy String Matching Algorithms

A class of algorithms that tell you how closely on string matches another by counting the number of **inserts**, **deletes** and **replacements** to convert one string to the other.

For example:

K I T T E N

S I T T **I** N **G**

3 operations.
Replace K, Replace E and Insert G.

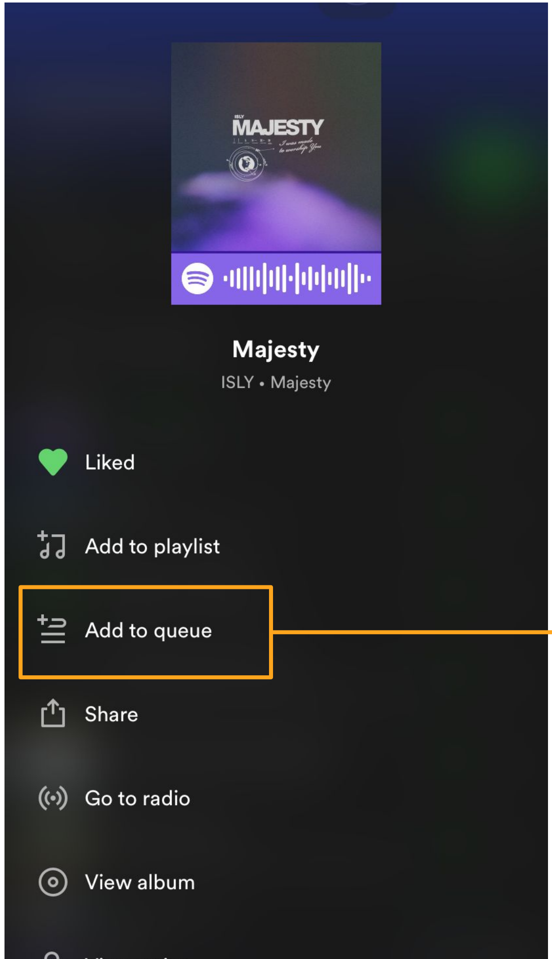
Lesson:

Knowing about the Fuzzy String Matching Algorithms helped me reason about how this problem might have been solved.

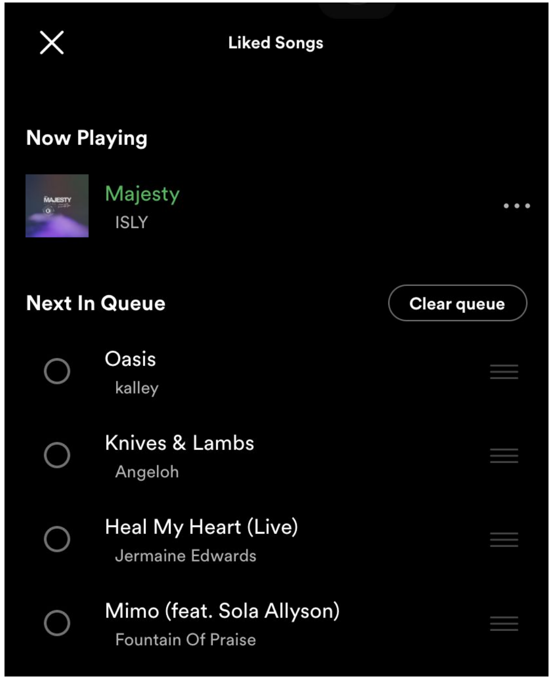
Spotify’s Music Player

DS: Queue (LinkedList)

In Spotify, you can add songs to the queue (which is really a LinkedList in disguise), you can delete songs from the queue and you can rearrange songs in the queue.



Read as **“Add to LinkedList”**, because a Queue is a LinkedList in disguise.



Liked Songs

Now Playing

Majesty
ISLY

Next In Queue

Clear queue

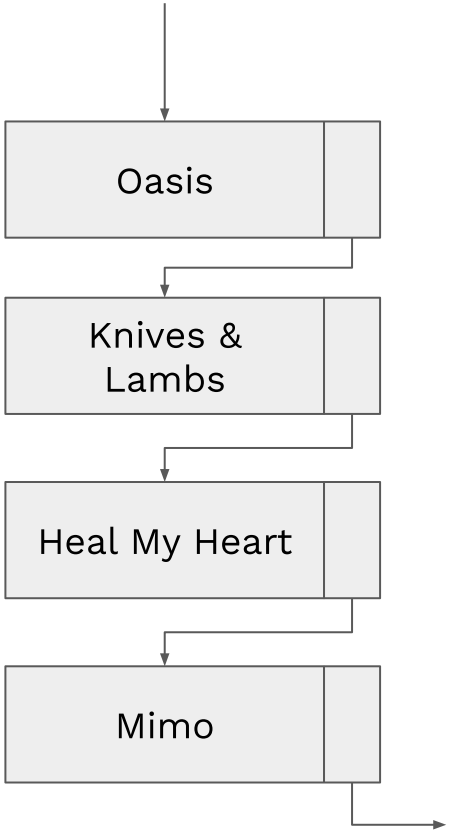
Oasis
kalley

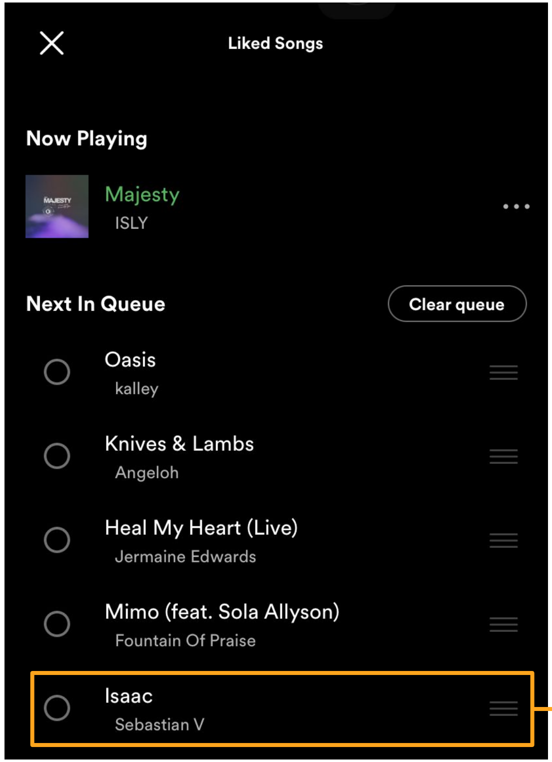
Knives & Lambs
Angeloh

Heal My Heart (Live)
Jermaine Edwards

Mimo (feat. Sola Allyson)
Fountain Of Praise

I can model the song queue as a **LinkedList.**





Liked Songs

Now Playing

Majesty
ISLY

Next In Queue

Clear queue

Oasis
kalley

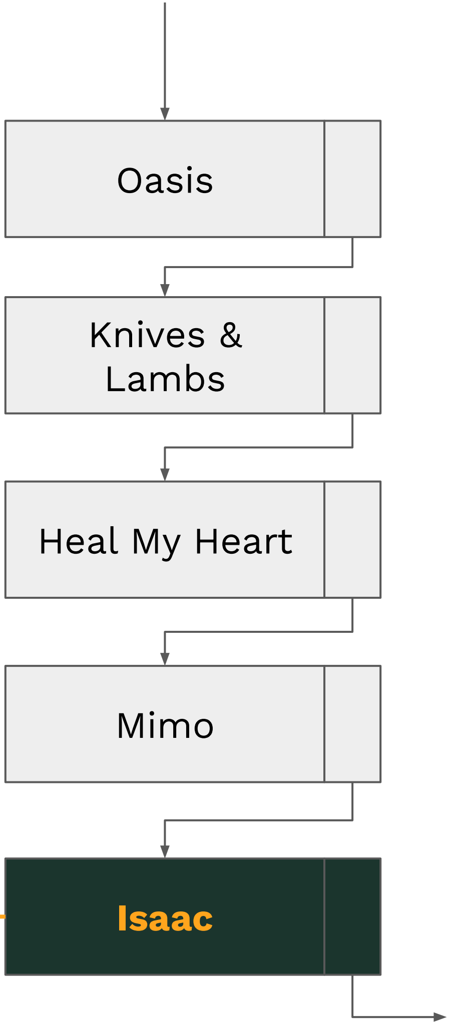
Knives & Lambs
Angeloh

Heal My Heart (Live)
Jermaine Edwards

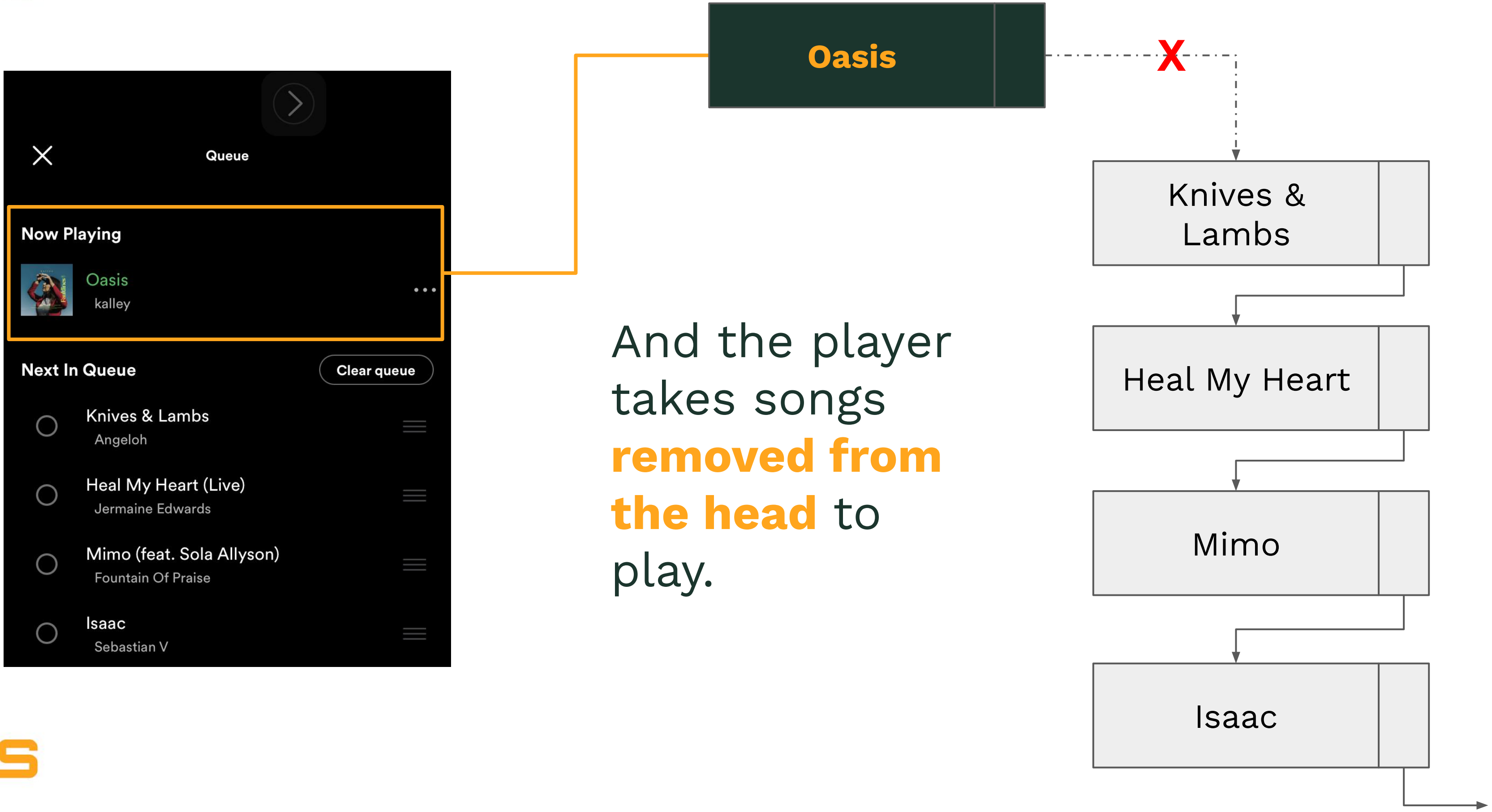
Mimo (feat. Sola Allyson)
Fountain Of Praise

Isaac
Sebastian V

New songs are **added to the tail.**



Open Source
Festival 2022


Queue

Now Playing

Oasis
kalley

Next In Queue

Clear queue

Knives & Lambs
Angeloh

Heal My Heart (Live)
Jermaine Edwards

Mimo (feat. Sola Allyson)
Fountain Of Praise

Isaac
Sebastian V

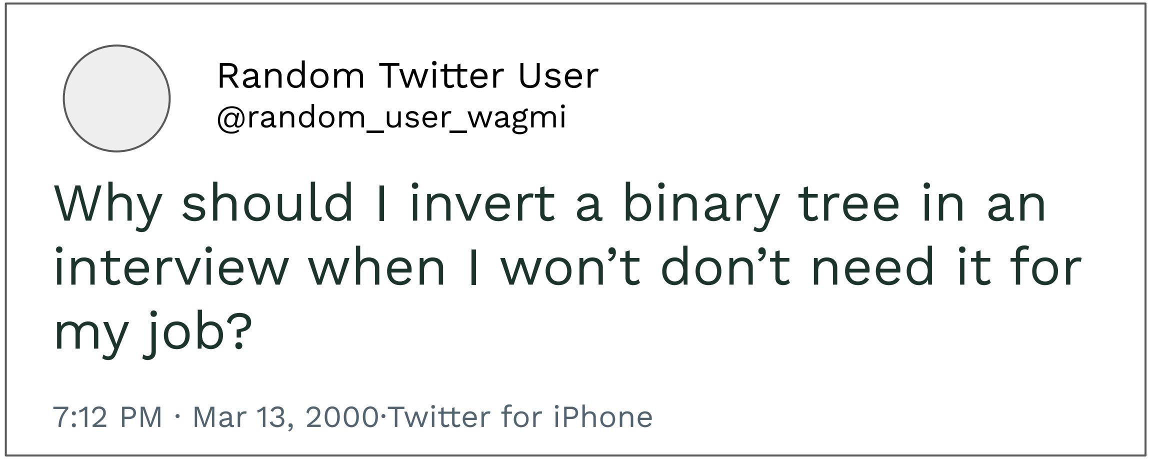
And the player takes songs **removed from the head** to play.

Question:

On Spotify’s player, you can rearrange and delete songs at any point in the LinkedList, but a using a LinkedList alone is slow for these two operations.

What data structure can we use in addition to a LinkedList to speed this up and how will it work?

DSAs as a tool for novelty.



Random Twitter User
@random_user_wagmi

Why should I invert a binary tree in an interview when I won’t don’t need it for my job?

7:12 PM · Mar 13, 2000·Twitter for iPhone

Open Source
Festival 2022

Binary Tree Inversion =
Pointer Chasing + Recursion

someone from HN.

Learning DSAs helps you learn important concepts in low impact settings.

And it sets up a critical foundation.

One day, you’ll need to solve a problem stack overflow can’t help with.

DSAs will help you come up with **new, creative solutions.**

Thank you 🤖

Open Source Festival 2022

+

Sustain Africa