How do search engines work?

Search engines work in several stages:

- Crawling - think of this as gathering all of the books within a library
- Indexing - think of this as reading the books and making a structured list of the information within the books
- Ranking - think of this as recommending books to the reader

Crawling

- Web pages are discovered by search engines through **software programs called crawlers** (or spiders, bots, or robots)
- Crawlers **follow links from one webpage to another**, systematically visiting pages on the web
- They start from a set of seed URLs and visit other pages linked from those URLs
- Website crawlers **follow rules and guidelines established by website owners**, using mechanisms like the robots.txt file. These guidelines **direct crawlers on which areas of a website to explore or avoid**, respecting website preferences and ensuring privacy
- Once a crawler reaches a webpage, it **fetches the HTML content** of that page
- The crawler **examines the HTML structure and retrieves information**, such as text content, headings, links, and **metadata**
- To understand the structure of the webpage, **the HTML that was retrieved is broken down into individual components**
- This process involves **identifying elements, tags, and attributes** that hold valuable information like titles and headings

Indexing

- The data extracted from the webpage is indexed, which involves storing the collected information in a structured manner within a search engine's database
- Each word in the document is included in the page's index as an entry, along with the word’s position on the page
- The **index allows for quick retrieval and ranking of relevant web pages** in response to user queries

Ranking

- When a user enters a query, the search engine searches the index for matching pages and returns the results they believe are the highest quality and most relevant to the user's query

Benefits of search engine crawling & indexing

- The process of search engine **indexing** is **essential for search engines** to collect, examine and arrange online content
- It involves **collecting and storing information from web pages** in a searchable index

Improved search results

- Indexing webpages means search engines can:
  - Provide users with **relevant and up-to-date search results**
  - Match user searches with content which **increases the chances of accurate and valuable results**
- This means the user is **more likely to find what they're looking for quickly**, ideally on the first page of search results, without having to go to additional pages

Efficient retrieval

- Indexing enables **efficient retrieval of information**
- Search engines don't need to scan the entire web for every search query. They can just **search their indexed data to produce search results quickly**

Ranking & relevance

- Indexing enables search engines to **assess the relevance and quality of web pages**
- Search result **rankings are determined by various ranking algorithms** that analyse indexed data. These algorithms consider factors such as **keyword relevance, backlinks, and user engagement**

Freshness & updates

- Search engine crawlers **periodically revisit indexed web pages to detect updates and changes**
- This process guarantees that the search results **display the latest content** that is currently accessible on the Internet
- If a webpage has been updated and not re-crawled, the page may no longer be relevant for the user's search

PageRank Algorithm

- A crucial element of search ranking algorithms is the Page Rank algorithm
  - The algorithm was developed by Larry Page and Sergey Brin
  - Many search engines rely on it, particularly Google
- **Web pages are evaluated and ranked by the algorithm** based on their perceived relevance and importance

Why is the PageRank algorithm important?

- The PageRank algorithm was created to **tackle the difficulty of determining the importance of web pages** with the **immense amount of information available**
- The purpose of the algorithm is to **provide better search results that are more precise and related** by taking into account various factors beyond just matching keywords

Key elements of the PageRank algorithm

Link analysis

- The PageRank algorithm **analyses the structure of links between pages** on the web
- **Web pages are given importance** by the algorithm, which considers the **quantity and quality of inbound links from other pages**
- Each **link acts as a "vote" for the target page**, with the voting weight determined by the importance of the linking page
- Websites that have **more high-quality links pointing towards them** are deemed to be more valuable and pertinent and have a higher weight
- Webpages with a **higher weight will score more highly** and have a higher ranking

Link weight distribution

- **The importance of a webpage is calculated by PageRank**, which takes into account the total number of "votes" it has received
- The algorithm distributes the importance of a page to the pages it links to by s**haring a portion of its importance with each outgoing link**
- By following this process, **pages of superior quality are given greater importance** and make a larger impact in determining the ranking of other pages

Iterative calculation

- The PageRank algorithm uses a **repetitive calculation** process. At the beginning, every webpage is given the same value to start with
- In subsequent iterations, **the significance of each page is re-evaluated** by considering the weighted impact of inbound links
- The process continues until the rankings become stable

Damping factor

- In order to avoid infinite loops, an algorithm introduces a **damping factor that ranges between 0 and 1 (usually set at 0.85)**
- The damping factor is the **likelihood of a user clicking on a link at random** rather than following the links on the current page
- The damping factor ensures that the **ranking calculation includes user behaviour** and maintains harmony between discovering new links and staying on the current page

Factors influencing PageRank

Although the initial PageRank algorithm mainly concentrated on link analysis, present-day search engines consider many factors to improve search results rankings.

Relevance • The **content of a web page is a crucial factor** in determining its ranking in search results. This is influenced by the **keywords** used, the **quality** of the content, and **how relevant it is** to the search query	*User engagement* • **The way users interact with a website can be measured through metrics** like click-through rates, time spent on a page (dwell time), and bounce rates. These metrics can reveal the level of user engagement • Pages that receive **greater engagement from users may be deemed more valuable**	*Authority & trust* • The **reputation** and authority of a webpage or website **play a crucial role** • **Several factors can enhance a website's ranking**, including the **age** of the domain, **quality backlinks** from reputable sources e.g. government website or the BBC, and **trustworthy content**
Content freshness • Search engines value **fresh and up-to-date content** • Search queries may give **priority to web pages that are frequently updated** or have up-to-date information	*Mobile-friendliness* • As mobile devices became more prominent, **search engines started to factor in the mobile compatibility of web pages** when determining their ranking • Google primarily **uses the mobile version of a site's content to rank pages** from that site • Having a responsive design and **optimising the user experience on mobile devices can have a positive impact** on a website's rankings	Limitations & evolving nature • Although the **PageRank algorithm** is important in search engine rankings, it **is not the only factor** that determines them • **Search engines use different algorithms** and factors to guarantee that they provide varied, relevant, and top-quality search outcomes • Over time, the **details of the PageRank algorithm have undergone changes**. Search engines regularly enhance their ranking methods to cater to new challenges and meet user expectations

Internet – Server-side and Client Side processing

Server Side Processing

- Server side processing involves **running code and carrying out operations on the server** instead of on the client's device or browser
- Web development often involves utilising **server side programming languages like PHP, Python, Ruby, or Java** to handle incoming requests, process data, interact with databases, and generate dynamic content

Server side processing

Data retrieval & manipulation

- You can retrieve and manipulate data. PHP is capable of **interacting with databases, processing data, and generating dynamic content**

Server operations

- Perform server side operations that are **not accessible to the client**
- **Retrieving and displaying information from a database**

Form processing

- Handle form submissions, process the submitted data, and perform necessary validations or database operations on the server side

Client Side Processing

- Client side processing involves **carrying out code or processing tasks on the user's device**, usually within the web browser, instead of on the **server**
- This feature enables users to have **interactive and dynamic experiences** without constantly requesting data from the server
- Client side processing is **primarily done using JavaScript**, whereas server side processing is commonly carried out using **PHP**

Client side processing with JavaScript

JavaScript is a powerful scripting language that operates mainly on the client side. It provides developers with the ability to **modify web content, manage user interactions, and update the webpage dynamically** without requiring server requests. Here are a few examples of client side processing with JavaScript:

Form validation

- With JavaScript, it's possible to **validate user input in real time**, which means that **users can receive instant feedback** without the need for a server roundtrip
- E.g. when completing an online form, check that all required fields are filled out correctly and make sure the input meets the necessary format and length before sending the form to the server. If any areas are blank and need input (e.g. email address) the user will be notified before the form can be submitted

Benefits & drawbacks of server side processing

Benefits of Server Side Processing	**Drawbacks of Server Side Processing**
Improved security measures can be implemented through server side processing, ensuring the secure management of sensitive data, implementing access control measures, and guarding against common web vulnerabilities.	When multiple requests are made to a server, complex processing tasks can consume server resources and cause a decrease in overall server performance. This is known as **increased server load.**
Server side processing uses the resources of the server to perform advanced calculations, manipulate data, and interact with databases.	Using server side processing may cause latency because it involves communication with the server, which could lead to **slower response times** in comparison to client side processing.
Server side processing ensures **consistent behaviour across different devices and browsers**, as the processing logic is centralised on the server.	Server side processing **relies on the availability and reliability of the server** infrastructure. Downtime or performance issues can affect the functioning of the web application.
Server side processing can be **easily scaled** by adding more servers or optimising the server infrastructure to handle increasing traffic and user demands.	Server side processing typically requires a roundtrip to the server for each user action, **limiting real-time interactivity** and responsiveness.
	Server side processing may require more **complex development** and setup compared to client side processing, potentially increasing development time and effort.

Benefits & drawbacks of client side processing

Benefits of Client Side Processing	**Drawbacks of Client Side Processing**
Enhanced user experience is made possible through client side processing, creating interactive and dynamic user experiences. This eliminates the need for frequent server requests and page reloads.	There is a **potential security risk** with client side code as it can be seen by users, which may lead to sensitive information and operations being exposed or tampered with.
By offloading processing tasks to the client side, the **server load is reduced**, resulting in improved scalability and resource utilisation.	The **compatibility of devices and browsers** may vary, which can lead to issues with the client side code that depends on their capabilities and support.
User input can be instantly validated and **feedback can be provided in real-time**. This not only improves the user experience but also reduces the need for server roundtrips**.**	Client side processing can hurt **page load time**, particularly when dealing with large or complex operations that require substantial processing power.
With the use of JavaScript, web pages can have their content updated dynamically, resulting in a more seamless and **engaging browsing experience**.	Client side processing is heavily **dependent on JavaScript**. If the user's browser does not support or has disabled JavaScript, the functionality may become inaccessible or break.
Web applications can operate without an active internet connection by using client side technologies to provide **offline functionality.**	The accessibility of client side code to users can put **intellectual property at risk,** as it allows for easier viewing, copying, and modification of the code.

Choosing Server Side or Client Side Processing

The **choice** between client side and server side processing **depends on the specific requirements** of a task:

- Client side processing is better for tasks that require **immediate user feedback, real-time interactions, dynamic user interfaces, or data manipulation** within the browser. **JavaScript** is the primary language for such scenario
- Server side processing is better for tasks that involve **accessing databases, handling sensitive data, complex business logic, or server specific operations**. **PHP** and other server side languages are commonly used in these cases