1. Search Engines, history and different types

In the beginning there was **Archie** (1990, indexed computer files) and **Gopher** (1991, indexed plain text documents). **Lycos** (1994) and **AltaVista** (1995) were amongst the early popular Search Engines. **Google** (1998) appeared a little later. **MSN Search** débuted in 2005.

Google and **AltaVista** are examples of indexing search engines, which use varying criteria to rank relevant search engine results pages (SERPs).

Clusty, a meta-search engine, clusters results based on textual and linguistic similarity. Clusty attempts to provide relevant results more quickly than with a general-purpose search engine, discover relationships between items and view deeper results that would be buried in a purely ranked list.

Dogpile, also a meta-search engine, combines the results of several search engines, thus giving potentially greater coverage of the Web, given the assumption that no search engine covers the entire web.

Ask Jeeves (now simply **Ask**), specialises in natural language queries (NLQs).

Yahoo! is an example of a directory rather than a search engine, populated by human editors. In the early days of the web this proved an ideal way to navigate the web, but does not scale so well as the web assumes huge proportions.

Some search engines have worked on a pay-per-view basis, with companies paying for clickthroughs, or visits to their websites. **Overture** (originally GoTo.com) is an example of this, and is now part of *Yahoo! Search Marketing*.

A potential extension to this concept is for companies to pay search engines only for completed sales. There would need to be a way to verify ‘sales’, and a far higher payment rate could be applied. This is the holy grail of targeted advertising.

2. Challenges faced by Search Engines

- The web is growing faster than search engines can index (especially directory-based engines such as Yahoo!).
- Many pages are updated frequently (e.g. News providers) which forces search engines to revisit periodically.
- Queries are mostly limited to *key word* searches. Proximity searches offer an improvement, where search terms are limited to same-sentences or paragraphs.
- Dynamically-generated pages are difficult to index, since they do not persist.
- The **deep web** cannot be indexed.
- Link farms, Link selling (from sites with a high PageRank) and other manipulations attempt to subvert SERPs.
- A “google bomb” is a specific attempt to influence the ranking of a web page for a given phrase by adding links to the page with the phrase as the anchor text (e.g. “failure” ranks the biography of G.W. Bush highest).

3. How does PageRank work, and how can it be personalised?

$$PR(W) = T/N + (1-T)(\ PR(W_1)/O(W_1) + PR(W_2)/O(W_2) + \ldots + PR(W_n)/O(W_n)\ )$$

where *T* is the teleportation probability (if *T* = 1 most of the equation disappears). The PageRank for a given page *W* is related to the sum of the PageRanks of all the pages linking to *W* each divided by the number of outlinks for those pages (only a portion of each page's PageRank contributes to *W*'s PageRank).

PageRank assigns a user-independent 'importance' to pages. It can be personalised by biasing the algorithm to prefer certain kinds of pages – those preferred by the user (e.g. by using features of the URL such as the domain). This will raise the PageRank of pages preferred by the user, thus *personalising* it.

4. What is the idea behind the "random surfer" model?

PageRank is based on the random surfer model. The model has a parameter $0 \leq a \leq 1$. More precisely, in each time step a surfer, currently visiting a document with probability *a* follows a link or, with the remaining probability 1 – *a*, randomly jumps to another page.

The traditional Random Surfer model *does not* reflect some very common aspects of Web surfing i.e. *reverting* to a page previously visited. Reverting to previous pages constitutes a substantial part of Web surfing behaviour and should be reflected in ranking schemes.

5. What is the PageRank 'rank sink' problem and how is it overcome?

A *rank sink* is a loop between web pages, e.g. A→ B→ C→A which could cause problems when generating PageRank. This problem can be overcome by using a *rank source*, or the concept of *random jumps* with a probability. This circumvents the infinite loop scenario. This is described in Page, Brin & Winograd's paper The PageRank Citation Ranking: Bringing Order to the Web (1998).

6. Calculating PageRank

PageRank is an iterative algorithm which is repeated until the PageRanks stabilise. The teleportation (or *random surfer*) element has an effect on PageRanks – the higher the teleportation probability *T*, the more the PageRanks tend towards the same value. When *T* = 1, all PageRanks are equal. However, when *T*=0, there is the possibility of a rank sink, which is to be avoided. Page & Brin suggested a value of 0.15 (i.e. 85% of the time a user will traverse by following links).

(i) calculate adjacency matrix

		Citing page					
		P_1	P_2	P_3	P_4	P_5	P_6
Cited page	P_1		1	1		1	
	P_2			1			
	P_3						1
	P_4	1		1	1	1	
	P_5	1					
	P_6			1			1
	Number	2	1	4	1	2	2

(ii) normalise by number of links

		Citing page					
		P_1	P_2	P_3	P_4	P_5	P_6
Cited page	P_1		1	0.25		0.5	
	P_2			0.25			
	P_3						0.5
	P_4	0.5		0.25	1	0.5	
	P_5	0.5					
	P_6			0.25			0.5
	Number	2	1	4	1	2	2

(iii) initially all pages have weight = 1

$$w_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

(iv) recalculate weights

$$w_2 = Bw_1 = \begin{bmatrix} 1.75 \\ 0.25 \\ 0.5 \\ 2.25 \\ 0.5 \\ 0.75 \end{bmatrix}$$

(v) iterate until $w_k = Bw_{k-1}$

7. What is TF-IDF? Compare PageRank and TF-IDF.

Term frequency (TF): A term that appears many times within a document is likely to be more important than a term that appears only once.
Inverse Document Frequency (IDF): A term that occurs in a few documents is likely to be a better discriminator that a term that appears in most or all documents.
The TF-IDF weight is a statistical measure used to evaluate how important a word is to a document. A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights tend to filter out common terms.
e.g. if the word *Web* appears 8 times in a 200 word document, its TF is 8/200 (0.04). If only 2 documents out of 100 contain the word *Web*, its IDF is 2/100 (0.02). TD-IDF is therefore 0.04/0.02 = 2.

PageRank: The rank of a web page is higher if many pages link to it. Links from highly ranked pages are given greater weight than links from less highly ranked pages.

With TF-IDF documents are ranked depending on how well they match a specific query. With PageRank, the pages are ranked in order of importance, with no reference to a specific query.

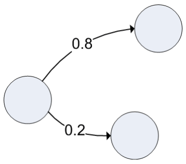
8. What is a Markov chain? Why are they useful?

A Markov Chain (named after Russian mathematician Andrey Markov), is a network with a probability function imposed on it. It is a statistical model concerning states and transitions with the property that it is *memoryless*, i.e. it does not remember its previous states (however, real life surfing is not as idealised as this).

A Markov Chain has two components: (i) a network structure where each node is called a state; (ii) a transition probability of traversing a link from one state to another.

The sum of outgoing probabilities from a state is always 1, i.e. $\sum_i p_i = 1$

e.g.



A *random walk* is a path through the network. The probability of the path is the **product** of all probabilities along the path (<1).

We can write a Markov Chain as a sparse matrix, with probabilities in each cell where there is a link. The sum of outgoing probabilities is always 1.

Support *s* in a chain accepts only paths whose initial probability is above *s*.
Confidence *c* accepts only paths whose probability is above *c*.

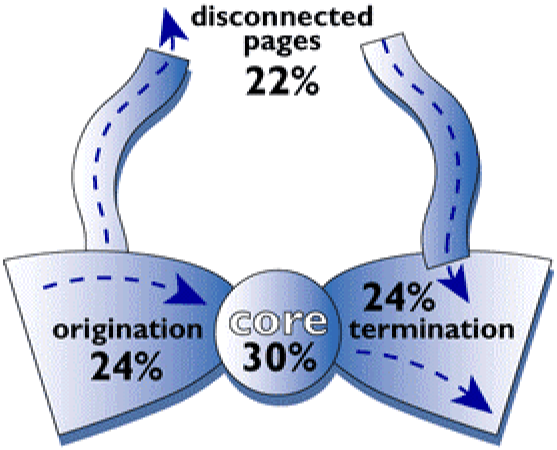
9. What are Power Law Distributions? Give Examples.

Power Laws are those which demonstrate exponential growth and can be seen as straight lines on a log-log graph.
A power law relationship is of the form $y = ax^k$ which can be expressed as $\log(y) = k \cdot \log(x) + \log(a)$ which itself is of the straight line form $y = mx + c$.
Examples include scale-free networks such as the web (a few highly-connected nodes, many low-connected nodes), Newtonian gravity (inverse-square law), the Pareto principle (wealth distribution *aka* '80-20 rule'), earthquake magnitudes (the Richter Scale), storm strengths (the Beaufort scale) and city sizes.

Power Laws often have "Long Tails" – the low-frequency, low-amplitude portion of the population. The term was coined in a 2004 article by Chris Anderson in *Wired*.

Hubs & Authorities – some nodes are popular (many inlinks, e.g. hubs), others are focal points (many outlinks, e.g. authorities). Many are both.
Preferential Attachment – *aka* 'the rich get richer'. Popular sites tend to become more popular, thus accentuating the power law distribution.

10. The 'Bow-Tie' theory



This model suggests that only a portion of the Web (the core) is well-connected.

There are many pages which link *into* the core, but are not linked *to* (e.g. new pages) and many which are linked *to* but do not link *back* to the core (e.g. corporate websites).

There are also *tubes* and *tendrils* which avoid the core, and *disconnected* pages.

11. What are recall/precision in the context of Search Engines?

Recall is the ability of a search engine to obtain all of the relevant documents.
Precision is the ratio or fraction of a search output that is relevant for a particular query.

There is a certain amount of subjectivity with these terms, and it is generally not possible to know the total number of relevant documents in the case of calculating *recall*.

12. Recommender Systems and Collaborative Filtering. What is collaborative filtering? What is the difference between user-based and item-based collaborative filtering?

Recommender systems are programs which make predictions, often implemented using *collaborative filtering* algorithms. Data can be gathered implicitly (e.g. recording purchases or page impressions) or explicitly (asking for ratings – e.g. IMDB – or relative rankings).

Collaborative filtering (CF) is the method of making automatic predictions (filtering) about the interests of a user by collecting taste information from many users (collaborating). The underlying assumption of CF approach is that those who agreed in the past tend to agree again in the future.

User-based collaborative filtering predicts a user's interests based on information from similar user profiles. A user-item matrix is constructed, and correlation between near-neighbours (users) can be calculated (treating the user information as vectors).

Item-based collaborative filtering (e.g. Amazon.com, "users who bought x also bought y") works by building an item-item matrix which determines relationships between pairs of items, and using the matrix and the data on the current user, makes inferences.

Difficulties with CF include sparsity (many users, but few ratings), scalability (huge numbers of users and items, e.g. customers and books) and the first-rater problem (i.e cold-starting).

13. Web Data Mining – contrast content mining, link mining and usage mining.

- Content mining concerns mining useful information or knowledge from web page contents.
- Usage mining refers to the discovery of user access patterns from Web usage logs.
- Web Link mining tries to discover useful knowledge from the structure of hyperlinks.

Content mining retrieves and analyses web content, classifying web objects (pages, websites, etc). Objects can be clustered and web crawlers can use a focused search to reveal patterns. DMOZ (the Open Directory Project) is an example of clustering by classification.

Applications of usage mining include website reorganisation, caching oft-accessed pages, recommendation and personalisation.
Users can be identified by IP address (unreliable), cookies (security/deleted cookie issues), logins.
Sessions can be identified by time-stamping or by navigation connections. Sessions illustrate *user trails* through a website. Markov chains can be constructed from user trails. By adding start and end states, the chains are termed *absorbing* (i.e. self-contained). Pattern discovery tools can identify complex patterns in usage.

PageRank and **Hubs and Authorities** are the two most well-known applications of link mining.

Other forms of mining include association rule mining, market basket analysis and cluster analysis.

14. What is capture/recapture?

This is a simple mechanism for estimating the size of a population. We begin by capturing a sample of size *c*, tagging them, and then returning them to the pond. After allowing some time for mixing, a second recaptured sample of size *r* is taken and the number *t* of them which are tagged is recorded. An estimate of the number of fish in the pond is

$$N = \frac{c \cdot r}{t}$$

Put another way, the ratio of recaptured to captured (*t/c*) is estimated to be the same as the ratio of recaptured sample size to total population (*r/N*).

In the context of the size of the web, we can use two search engines' results to simulate the capture/recapture, with the overlap providing the recapture value. We need to know the size of the web as reported by one of the search engines. The equation thus becomes

$$Size\,of\;web = \frac{size\,of\;\; SE1 \cdot \#\; pages\; returned\; by\; SE2}{overlap\; between\; SE1 \,\&\, SE2}$$

This is becoming a less precise metric as more of the web consists of dynamic pages. It is no longer possible to determine the '*size*' of the web in terms of pages.

15. RSS (Really Simple Syndication)

RSS is a way to publish and receive information, *without using e-mail*. Advantages are that recipients opt-in to receiving an RSS feed, it is not filtered (like e-mail) and can be updated as often as required. RSS feeds can be read using an RSS Feed Aggregator. RSS is *pulled* by a recipient (or its feed aggregator), not *pushed* by the publisher.

RSS is based on XML. The RSS specification describes the structure of the XML necessary to deliver RSS information. A minimal structure is

```
<rss version="2.0">
  <channel>
    <title>RSS feed title</title>
    <link>http://www.myfeed.com/</link>
    <description>short description of feed</description>
    <item>
      <title>story headline</title>
      <description>summary of story</description>
      <link>http://www.myfeed.com/story</link>
    </item>
  </channel>
</rss>
```

Channels can also include an image, published date, editor, etc. There can be many *items* within a channel. Items can also include author, category, published date, etc.