**PAGE RANGE / WEB SEARCH:**

**Overview:**

View web as directed graph

If we have pages p and q, we have an edge from p to q if p has a link to q

Usually given a web graph G = (V, E)

V is the set of all pages

Often have multiple edges and self-loops

Several Steps:

1. Crawling: Visit all web-pages and gather information
2. Rank the web pages
3. Create an index of web pages to enable efficient retrieval when a query arrives

Graph traversal algorithms are BFS and DFS

BFS is used more in web searches

**BFS:**

Start at root

Visit all vertices at distance 1 from root

Visit all vertices at distance 2 from root

Etc…

*Algorithm:*

*Input: G = (V, E)*

*Visited = {root}*

*Q = {root}*

*While Q is not empty:*

*v = first element in Q*

*remove first element in Q*

*For every edges <v, u>:*

*If u is not in visited:*

*Add u to visited*

*Add u to Q*

**Web Search:**

Main bottleneck for web crawl is sending a request and waiting for the response

Politeness policy is not to send too many requests to a server within a short time