**Question**

Web Crawler:

If you were designing a web crawler, how would you avoid getting into infinite loops?

**Working of a web crawler**

1. Visits a set of pages (present in the form of a queue. These pages will be very frequently visited. Highest ranked pages)
2. From that page it will fetch all the details for crawling (mostly it will check the robots.txt) and determine the key words and all the pages links, add these links to the queue and then remove the page from the queue
3. Then it will proceed with the next in queue till it reaches a stage where there is nothing in the queue. i.e., no pages to visit.
4. While visiting each page it will create a HashMap and store its values (Keywords)

Now back to the question, how will an infinite loop be created?

As you can see in the 2 point that it will check for the links and add them to the queue, what if the page after leading to some three other pages is pointed back to the page. This has a potential to become serious as this will create an infinite loop.

**Solution**

We have to design a crawler in such a way that infinite loop is avoided.

**Let’s consider that HashMap is the way the crawler identifies the keywords associated. When searched in search engine**

**Links to subsequent visits will be added here**

**Queue**

**URL**

**List of all its links and keyword**

Now when the new URL’s discovered in the specific page is being added into the HashMap we can find if there is duplicate (i.e., find with hash if the URL - **Key** is already present), if so ignore them.

Finally stop Crawling once we reach the end

**One Assumption here is that the HashMap will have numerous space since this is usually how the spider (Web Crawler) works.**