Computer Science 572 Exam
Prof. Horowitz
Wednesday, February 22, 2017, 8:00am – 8:50am

Name: Siddhesh Rajiv Karekar **Student Id Number:**

1. This is a closed book exam.
2. Please answer all questions.
3. There are a total of 25 questions. Each question is worth 4 points.
4. **Place your answer immediately below the question. Limit answers to ONE SENTENCE.**

1. If *tp* represents true positive, *fp* represents false positive, *fn* represents false negative and *tn* represents true negative, define Precision and Recall

Answer: Precision = tp / (tp + fp)
Recall = tp / (tp + fn)

2. The harmonic mean of precision and recall is sometimes called the *F* measure. If *R* represents Recall and *P* represents Precision, define *F*

Answer: F = 2PR / (P + R)
or
F = (b^2 + 1)PR / (b^2*P + R)

3. Define the three main properties of cryptographic hash functions

Answer: 1) should be difficult to reverse
2) even a small change in input text must generate significantly different hash
3) should be difficult to find two different input strings must generate the same hash
4) should be easy to compute

4. Name one cryptographic hash function

Answer: SHA-1, SHA-2, MD5

5. Given two sets A and B, define their Jaccard similarity

Answer: size of A intersect B / size of A union B

6. If *df(t)* is the document frequency of term *t* out of *N* documents, what is the range of values *df(t)* can take on

Answer: 0 - N

7. Does *idf* (inverse document frequency) have an effect on ranking for one term queries?

Answer: No

8. One alternative to the use of document frequency is collection frequency. Define collection frequency for a term *t*.

Answer: The total number of time a term t appears across all documents in a collection.

9. if *tf(t,d)* is the term frequency of term *t* in document *d*, out of a total of *N* documents, and *df(t)* is the document frequency of term *t*, define the *tf-idf* weight of term *t* in document *d*

Answer: td.idf(t,d) = tf(t,d) * log (N / df(t))

10. Given a query *q* and a set of documents *D*, and the *tf-idf(t,d)* the weighted score of term *t* in document *d* in *D*, what is the score of the query *q* with respect to document *d*

Answer: Sum ((1 + log tf(qi,d)) * log (|D| / idf (qi)))

11. State Heap's Law

Answer:

$V = Kn^{beta}$
V = vocabulary / set of unique words
K = constant
n = set of all words
beta usually in the range 0.4 - 0.6

The law states that the number of distinct words in a document is a function of the document length.

12. According to Zipf's law how often will the most frequent word occur as compared to the second most frequent word? How often will the most frequent word occur as compared to the third most frequent word?

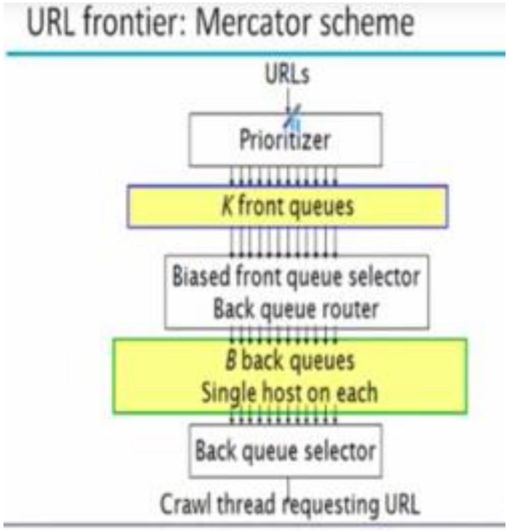
Answer: Let the most frequent, second most frequent and third most frequent words be a, b, c

Approximately,
a = 2b
a = 3c

13. The rank/frequency data that satisfies Zipf's law is generally drawn using a log-log scale. Why?

Answer: It appears in a straight line

Below is a diagram that explains part of the Mercator crawler and it was the subject of one of the videos shown in class. Please answer the following questions about the diagram



URL frontier: Mercator scheme

14. Which queues control freshness, the front queues or the back queues?

Answer: Front

15. Which queues control politeness, the front queues or the back queues?

Answer: Back

16. Which queues make use of a min-heap data structure?

Answer: Back

17 In Google if the query is as shown in the next line
February President OR “Abe Lincoln” dies
show how Google interprets the query by fully parenthesizing the query and insert all Boolean operators

Answer: February AND (President OR "Abe Lincoln") AND dies

18. Google offers a set of special operators that can be used to modify queries. Mention four:

Answer: inurl, inanchor, daterange, intext, allinanchor, filetype, cache

19. What is the technique used to speed up the merging of postings in an inverted index?

Answer: Skip pointers

20. Name four criteria YouTube uses to rank its search results.

Answer: Subtitles, Upload date, Number of views, number of likes, quality

21. To come up with related videos, YouTube forms a co-visitation count for each pair of videos. What is the co-visitation count?

Answer: co-visitation count for (x,y)
The total number of times a video y was watched after x across all user sessions

22. An inverted index is often split into two parts, name them.

Answer: Dictionary and postings list

23. In one sentence explain who uses ContentID and what it does.

Answer:
- YouTube uses ContentID
- It is a system to generate a digital fingerprint of a file, such as a song or a video
- Companies send their original content to YouTube, which generates a ContentID for each
- If the fingerprint for any user-uploaded video matches an existing ContentID, the company can decide what to do

24. The Java program below is part of crawler4j and defines which pages to crawl. In particular crawler4j will not crawl css, js, gif, jpg, png, mp3, gz, and zip files. However there are two lines that include XXXXXXX denoting code that is missing. Fill in the missing code.

```
                                        Crawler
public class MyCrawler extends XXXXXXX {
   private final static Pattern FILTERS =
Pattern.compile(".*(\\.(XXXXXXXX))$");
    @Override      css|js|gif|jpg|png|mp3|gz|zip
    public boolean shouldVisit(Page referringPage, WebURL url) {
       String href = url.getURL().toLowerCase();
       return !FILTERS.matcher(href).matches()
            && href.startsWith("http://www.cnn.com/");
    }
```

Answer:

25. The Java program below is part your crawler4j homework exercise. However there are two lines that include XXXXXXX denoting code that is missing. Fill in the missing code.

```
               Controller
public class XXXXXXX {
   public static void main(String[] args) throws Exception {
      String crawlStorageFolder = "/data/crawl";
      int numberOfCrawlers = 7;
      CrawlConfig config = new CrawlConfig();
      config.setCrawlStorageFolder(crawlStorageFolder);
      PageFetcher pageFetcher = new PageFetcher(config);
      RobotstxtConfig robotstxtConfig = new RobotstxtConfig();
      RobotstxtServer robotstxtServer = new RobotstxtServer(robotstxtConfig, pageFetcher);
      CrawlController controller = new XXXXXXX;
      controller.addSeed("http://www.cnn.com/”);
     controller.start(MyCrawler.class, numberOfCrawlers);
   }
}
```

Answer: