CS 345: Homework 3 (Due: Oct 7, 2014)

Rules. Individual homeworks; see Handout 1 (aka Syllabus). Hardcopies may be submitted no later than start of class the day they are due; electronic copies by NOON-time the same day.

Problem 1. (18 points) **(Estimate Corpus size of Google and Bing)**

Last time (2013) we tried this problem the best set of keywords to estimate reliably the corpus size of Google and Bing was `pasta` for `w1` and `physics` for `w2`: 24billion and 12billion were the two estimates respectively. The textbook's attempt of `tropical` and `Lincoln` does not work any more. And it is very likely that pasta and physics won't work either!

We ask you to try find three queries for `w1` , `w2` and `w1 AND w2` that will help you estimate the size of Google's and Bing's corpus. Estimates obtained MUST be higher than 50% but no more that 200% of estimates obtained in Homework 2 or in class. (This means we are forgiving some underestimation and some overestimation.)

In line 1 below include your `w1` results, in line 2 the `w2` and in line 3 the `w1 AND w2` for the choice of terms `w1, w2` that you pick. Nothing used in class may be reused including the keywords above.

No	Query Term(s)	GOOGLE No of hits	BING No of hits
1.	$\langle$ $\rangle$		
2.	$\langle$ $\rangle$		
3.	$\langle$ $\rangle$		
E1	What is the corpus size based on 1-3?		
2013.1	$\langle pasta \rangle$	237M	50.1M
2013.2	$\langle physics \rangle$	223M	63.4M
2013.3	$\langle pasta\ physics \rangle$	2.19M	262k
2013	What is the corpus size based on the 2013 data?	24G	12G

Table 1: Table for HW3.P1

Problem 2. (24 points) **(Paper by Brin and Page)**
Read the paper (pdf and HTML through link L1 in section C5 of the course web-page). For Hashin consult wikipedia or your favorite CS114 or CS435 textbook. Justify your answers based on the info of the papers.
`http://en.wikipedia.org/wiki/Hash_table`
might also help.

(a) According to the paper, what was the size (bytes) of Google's Lexicon around 1998?

(b) According to the paper how many bits for a `docID`? Quote the paper.

(c) Describe the data structures used by Google for indexing only (not all of them are listed in the architecture figure).

(d) In what data structured does Google use binary search?

(e) Does Google (1998) use a hash table for the dictionary? What do they use? Why ?

(f) How many bytes are assigned to each hit (of the hitlist)? How many types of hits? What are they (types of hits)?

Problem 3. (18 POINTS) **(Hash Table Design ala Google/ 1998)**
You have 2GiB of main memory of which 256MiB are being used by the operating system plus related programs such as those manipulating tables A and T below.

You are asked to organize the additional space to support a hash table along the lines of the paper where words are stored in a contiguous table which is an array A of characters delimited by a null character `\0`. A hash table T will store a `wordID` along with a reference (pointer or index) p to A. That way a `wordID` will be associated with a specific `word`.

The average length of a `word` is given as 7 characters stored in Unicode (1 Unicode uses two bytes). A `wordID` and p can only be in multiples of one byte (i.e. 1B, 2B, 3B but and thus 23bits won't be an option) for efficiency. Organize T and A for maximum efficiency.

In an efficient implementation

- (a) How many words n can the scheme support,
- (b) How big would the hash table size m (number of entries) be,
- (c) How many bits for a `wordID`,
- (d) How many bits for pointer p
- (e) How much space (bytes) will you scheme use for T,
- (f) How much space (bytes) will you scheme use for A,
- (g) What is the total space $A + T$ used by your scheme?

Justify your answers and choices. Round to nearest million for n, m, T, A but make sure you don't exceed the amount of available memory (i.e. $A + T$ should be accommodated easily by the available memory). Make sure that you fill the following table with data.

```
   n        =
   m        =
   wordID   =
   p        =
   T        =
   A        =
 A+T        =
```

Date Posted: 9/23/2014

Addendum (SI System and Units): The symbols for $10^3, 10^6, 10^9$ are k , M, and G respectively (and go on with P, T) for kilo,mega,giga,peta,tera. Note that the k is lower case. For megabyte, gigabyte etc avoid ambiguity by using kibi, mebi,gibi, tebi bytes respectively ie. KiB, MiB, GiB, TiB. Note that B is for `byte` but **bit** is for a bit. A lower-case b is nonsense. The kibi et al units are 2^{10}B, 2^{20}B, 2^{30}B and 2^{40}B respectively. (Whereas 1kB (and not 1KB) used to denote 1024B, the 1MB and 1GB terminology has been ambiguous; for main memory it denotes 1MiB and 1GiB respectively but disk drive manufacturers preferred them to refer to $10^6, 10^9$ bytes instead.)
YOU ARE INTO COMPUTING, SO USE your terminology for BYTES properly and correctly.