**Sheet 2 ( IR 3# IS CH.2)**

**Exercise 2.1**

**Are the following statements true or false?**

**a. In a Boolean retrieval system ,stemming never lowers precision.**

**b. In a Boolean retrieval system, stemming never lowers recall.**

**c. Stemming increases the size of the vocabulary.**

**d. Stemming should be invoked at indexing time but not while processing a query.**

**Exercise 2.2**

**Suggest what normalized form should be used for these words (including the word itself as a possibility):**

**a. ’Cos**

**b. Shi’ite**

**c. cont’d**

**d. Hawai’i**

**e. O’Rourke**

**Exercise 2.3**

**The following pairs of words are stemmed to the same form by the Porter stemmer. Which pairs would you argue shouldn’t be conﬂated. Give your reasoning.**

**a. abandon/abandonment**

**b. absorbency/absorbent**

**c. marketing/markets**

**d. university/universe**

**e. volume/volumes**

**Exercise 2.4**

**For the Porter stemmer rule group shown in (2.1):**

**a. What is the purpose of including an identity rule such as SS→SS?**

**b. Applying just this rule group, what will the following words be stemmed to?**

**circus canaries boss**

**c. What rule should be added to correctly stem pony?**

**d. The stemming for ponies and pony might seem strange. Does it have a deleterious effect on retrieval? Why or why not?**

**Exercise 2.5**

**Why are skip pointers not useful for queries of the form x OR y?**

**Exercise 2.6**

**We have a two-word query. For one term the postings list consists of the following16 entries:**

**[4,6,10,12,14,16,18,20,22,32,47,81,120,122,157,180]**

**and for the other it is the one entry postings list:**

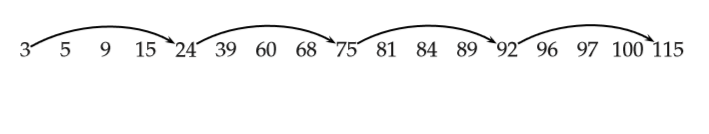
**[47].**

**Work out how many comparisons would be done to intersect the two postings lists with the following two strategies. Brieﬂy justify your answers:**

**a. Using standard postings lists**

**b. Using postings lists stored with skip pointers, with a skip length of √P, as suggested in Section 2.3.**

**Exercise 2.7**

**Consider a postings intersection between this postings list, with skip poin**

**And the following intermediate result postings list (which hence has no skip pointers) :**

**3 5 89 95 97 99 100 101**

**Trace through the postings intersection algorithm in Figure2.10 (page 37).**

**a. How often is a skip pointer followed (i.e., p1 is advanced to skip (p1))?**

**b. How many postings comparisons will be made by this algorithm while intersecting the two lists?**

**c. How many postings comparisons would be made if the postings lists are intersected without the use of skip pointers?**

**Exercise 2.8**

**Assume a biword index. Give an example of a document which will be returned for a query of New York University but is actually a false positive which should not be returned.**

**Exercise 2.9**

**Shown below is a portion of a positional index in the format: term: doc1: (position1, position2, ...); doc2: (position1, position2, ...); etc.**

**angels: 2: (36,174,252,651); 4: (12,22,102,432); 7: (17);**

**fools: 2: (1,17,74,222); 4: (8,78,108,458); 7: (3,13,23,193);**

**fear: 2: (87,704,722,901); 4: (13,43,113,433); 7: (18,328,528);**

**in: 2: (3,37,76,444,851); 4: (10,20,110,470,500); 7: (5,15,25,195);**

**rush: 2: (2,66,194,321,702); 4: (9,69,149,429,569); 7: (4,14,404);**

**to: 2: (47,86,234,999); 4: (14,24,774,944); 7: (199,319,599,709);**

**tread: 2: (57,94,333); 4: (15,35,155); 7: (20,320);**

**where: 2: (67,124,393,1001); 4: (11,41,101,421,431); 7: (16,36,736);**

**Which document(s) If any match each of the following queries, where each expression within quotes is a phrase query?**

**a. “fools rush in”**

**b. “fools rush in” AND “angels fear to tread**

**Exercise 2.10**

**Consider the following fragment of a positional index with the format:**

**word: document: (position, position, ...); document: (position, ...)**

**...**

**Gates: 1: (3); 2: (6); 3: (2,17); 4: (1);**

**IBM: 4: (3); 7: (14);**

**Microsoft: 1: (1); 2: (1, 21); 3: (3); 5: (16,22,51);**

**The /k operator, word1 /k word2 ﬁnds occurrences of word1 within k words of word2 (on either side), where k is a positive integer argument. Thus k = 1 demands that word1 be adjacent to word2.**

**a. Describe the set of documents that satisfy the query Gates /2 Microsoft.**

**b. Describe each set of values for k for which the query Gates /k Microsoft returns a different set of documents as the answer.**

**Exercise 2.11**

**Consider the general procedure for merging two positional postings lists for a given document, to determine the document positions where a document satisﬁes a /k clause (in general there can be multiple positions at which each term occurs in a single document). We begin with a pointer to the position of occurrence of each term and move each pointer along the list of occurrences in the document, checking as we do so whether we have a hit for /k. Each move of either pointer counts as a step. Let L denote the total number of occurrences of the two terms in the document. What is the big-O complexity of the merge procedure, if we wish to have postings including positions in the result?**

**Exercise 2.12**

**Consider the adaptation of the basic algorithm for intersection of two postings lists (Figure 1.6, page 11) to the one in Figure 2.12 (page 42), which handles proximity queries. A naive algorithm for this operation could be O(PLmax2), where P is the sum of the lengths of the postings lists (i.e., the sum of document frequencies) and Lmax is the maximum length of a document (in tokens).**

**a. Go through this algorithm carefully and explain how it works.**

**b. What is the complexity of this algorithm? Justify your answer carefully.**

**c. For certain queries and data distributions, would another algorithm be more efﬁcient? What complexity does it have?**

**Exercise 2.13**

**Suppose we wish to use a postings intersection procedure to determine simply the list of documents that satisfy a /k clause, rather than returning the list of positions, as in Figure 2.12 (page 42). For simplicity, assume k ≥ 2. Let L denote the total number of occurrences of the two terms in the document collection (i.e., the sum of their collection frequencies). Which of the following is true? Justify your answer.**

**a. The merge can be accomplished in a number of steps linear in L and independent of k, and we can ensure that each pointer moves only to the right.**

**b. The merge can be accomplished in a number of steps linear in L and independent of k, but a pointer may be forced to move non-monotonically (i.e., to sometimes back up)**

**c. The merge can require kL steps in some cases.**

**Exercise 2.14**

**How could an IR system combine use of a positional index and use of stop words? What is the potential problem, and how could it be handled?**

**With my best wishes**

**Eng : Sara Salama**