Brandon Garza

CPTS 315

HW1

Analytical Part

Q1:

1. 4
2. 4/6 = 66%
3. Support {A, B} / support {A} = 4/6

Q2:

1. The formula to find the index of a pair is (i – 1)(n – i/2) + j – i. Plugging 7 in for i and 8 in for j, we get an index of 100.
2. The tabular method would be the preferred choice because it would save space as only the pairs that have a nonzero count will be stored which is only 10% of the pairs. Tabular method takes requires 12 bytes per entry while triangular matrix takes up 4 bytes, but the triangular matrix must store all pairs even if they have a zero count.

Q3:

1. 1 -> 4

2 -> 6

3 -> 8

4 -> 8

5 -> 6

6 -> 4

{1,2} -> 2

{1,3} -> 3

{1,4} -> 2

{1,5} -> 1

{1,6} -> 0

{2,3} -> 3

{2,4} -> 4

{2,5} -> 2

{2,6} -> 1

{3,4} -> 4

{3,5} -> 4

{3,6} -> 2

{4,5} -> 3

{4,6} -> 3

{5,6} -> 2

|  |  |  |
| --- | --- | --- |
| Bucket | Pairs | Count |
| 0 |  | 0 |
| 1 | {2,6}, {3,4} | 5 |
| 2 | {1,2}, {4,6} | 5 |
| 3 | {1,3} | 3 |
| 4 | {1,4}, {3,5} | 6 |
| 5 | {1,5} | 1 |
| 6 | {1,6}, {2,3} | 3 |
| 7 | {3,6} | 2 |
| 8 | {2,4}, {5,6} | 6 |
| 9 | {4,5} | 3 |
| 10 | {2,5} | 2 |

1. Buckets 1, 2, 4, and 8 are frequent with because they pass the support threshold of 4.
2. {2,6}, {3,4}, {1,2}, {4,6}, {1,4}, {3,5}, {2,4}, {5,6} because they are in the buckets

Q4:

The purpose of this study was to develop an algorithm that efficiently detected copies of documents or parts of documents by using identifiers. The identifier for a document is referred to as a “fingerprint”, which is a collection of bits of strings or k-grams that are assumed to be unique to the document. This paper argues for the use of the local algorithm, an algorithm that seeks to solve the problem that some fingerprinting algorithms have of choosing k-grams from a document that are too far apart. An algorithm is said to be local if there is a k-gram found for every window of a user defined size.

The paper argues that three of the most important properties of a copy detection algorithm are whitespace insensitivity, noise suppression, position independence. Whitespace insensitivity is the ability of the algorithm to ignore irrelevant information in text such as whitespace, capitols, and variable names when it comes to computer programs. Noise suppression is the ability of the algorithm to ignore common words such as ‘the’. And lastly position independence is the ability of the algorithm to detect similarities even if the paragraphs in a document are in a different order. The paper describes a variety of methods that were used and considered to address these three properties when developing the algorithm.

The local algorithm that is shown in the paper is presented as the winnowing algorithm and is optimized to address the above important properties and be efficient by changing aspects such as the window size, size of the k-gram, and number of fingerprints collected. Overall, they found that their method of winnowing was an efficient way to detect copies of documents and portions of documents.