Assignment 15

Alex Clemmer
Student number: u0458675

1

If there are 2^{36} pages on the internet and there are 2^4 links per page, then this is a dense graph, with 2^{40} edges.

An **adjacency matrix** for a graph with 2^{36} nodes would be $2^{36} \times 2^{36} = 2^{72}$—that is, one element for every possible edge (which in this case would be a URL). This matrix will never be larger or smaller, because it is a 2-dimensional matrix representing all possible edges, whether or not there is an edge there.

Each element is just 1 bit, so 2^{72} bits is the total size for the elements in the table. Since arrays in Java and C are implemented much the same way a jump table is, beyond that, you really just need a pointer or reference to the first element, at which point you can jump to any space by telling the computer exactly how many elements to jump.

If a terabyte is 2^{43} bits, then we need 2^{29} terabytes to hold this array.

An **adjacency list**, on the other hand, would require 2^{36} base nodes, each with an average of 16 64-bit elements, which means that the total required space for just the base nodes is $2^{36} \cdot 2^6 \cdot 2^4 \cdot 2^6 = 2^{52}$ bits = 512 terabytes.