**2006 Practice Quiz 1 CS1501**

**Fill in the Blanks and True/False (24 points -- 2 points each).**

Complete the statements below with the MOST APPROPRIATE words/phrases.

a)         An algorithm has a run-time of **Theta(n2)** and, **for n = 4 runs in 4 seconds** on a computer. How long will a problem of double the size **(n = 8)** take to run on the same computer? 16

b)         Order the following growth rates from smallest (best) to greatest (worst): **n2 nlgn lgn n! 2n n3**   
logn, nlgn, n^2, n^3, 2^n, n!

c)         Assume a graph has N vertices. An upper bound on the worst case run-time of a **Hamiltonian Cycle** algorithm on the graph is Theta n!

d)         Given a **multiway radix search trie** in which **64-bit keys** are compared **8 bits at a time**, the **maximum height** of the tree is \_\_\_64/8=8\_\_\_\_\_\_\_ and **interior nodes** will each have up to \_\_2^8 = 256\_\_\_\_ children.

e)         Consider an empty **separate chaining hash table** of size **100**. If we hash **200 keys** into this table, the average chain length will be \_\_\_2\_\_\_\_\_\_ and the worst case chain length will be \_\_\_200\_\_\_\_\_.

Average chain length: (Number of Keys to Hash/Size of Table) = 200/100 = 2.  
Worst Case Chain Length: One index of the table holds all the keys.

f)          An example **text string** and **pattern string** that will produce the **worst case** for the **brute force string matching** algorithm are:

A (text) = \_\_\_ \_XXXXXXXXXXY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ P (pattern) = \_XY\_\_ \_\_\_\_\_\_\_\_\_\_

Indicate whether each of the following is TRUE or FALSE, **explaining why in an informative way for false answers.**

g)         **Pruning** is a technique that improves the **asymptotic run-times** of exhaustive search algorithms.   
False, pruning only improves the search time, but not the run time

h)         We **used DLBs** in Assignment 1 **rather than regular multiway tries** because DLBs allow for **faster searches.   
False, DLBs allow for better memory usage, not faster searches.**

i)           The Java String operator "+" appends one String to another **in constant time**.   
True

j)           Due to the **Pigeonhole Principle**, I cannot avoid collisions in hashing if the size of my **key space** is larger than the size of my **hash table**.   
True

k)         The Knuth Morris Pratt string matching algorithm **improves** over the brute force algorithm in the **worst case**, but **not in the normal case**.   
True

l)           In the **Rabin-Karp** string matching algorithm, if the **hash value of the pattern** matches that of a **substring of the text**, the pattern has been found.   
False, Different strings such as anagrams may have the same hash value, which could result in a hash collision. Rabin Karp avoids these hash collisions because of its secure, large number algorithm. ASCII \* 64^n + ASCII \* 54^n-1 + …

2)      **(16 points � 8 + 8)**Consider an **emtpy de la Briandais Tree**, which uses the **lower case letters** (plus a string termination character) as its alphabet, using the implementation that we discussed in lecture.� Also consider the following strings:run sunday runny sun sunny

r s

u u

n n

^ d

a

y

^