CS 1501 Quiz 1

KEY
1. (4 points) Arrange the following orders of growth from least to greatest: $2^n, n^4, \log n, n!$, $n$

**ANSWER:**

- $\log n$
- $n$
- $n^4$
- $2^n$
- $n!$

2. (6 points) What is the worst-case runtime to insert into a DST?

What is the average-case runtime to insert into a DST?

**ANSWER:**

- $w$, a constant that is the bitlength of the key being inserted
- $\log n$
3. (7 points) Draw the result of inserting the following keys (as 4 bit integers) into a Radix search trie:

- 5 (0101)
- 10 (1010)
- 11 (1011)
- 3 (0011)
- 15 (1111)

**ANSWER:**

7 Trie is drawn as follows
4. (8 points) Assume that you want to build a directory application for a building. The building has 10 floors (numbered 0 - 9), and 25 office per floor (001-025, 101-125, 201-225, etc.). You goal is to design a data structure that will allow users to provide an office number and get a response of who is occupying that office (you do not need to support the reverse mapping). How would you solve this problem? Why did you choose this approach? State any assumptions that you make.

**ANSWER:**
Answers will vary. Valid assumptions to make: is the machine running this application memory limited or not? do the office occupants very frequently or not?

Example solutions:

- An RST as we would only need 10 bit to store a given key, this gives us constant time lookup.
- Assuming we are OK with wasted memory, we could allocate a 926 index array and store the name of each office user in their office’s index.
- An R-way trie with an alphabet of 0-9. This means only the leaves would have unused references, and they would still be half full. Wastes a bit of memory, but assuming this is OK, provides constant time lookup.
- Could try and use perfect hashing, but would need to provide a hash function, which would be tricky.