CISC 3220 Homework Chapter 1

Rachel Friedman

February 2, 2020

Exercises 1.1

Question 1.1-2

Other measures of efficiency include:

- Storage requirement
- Memory requirement
- Hardware requirement
- Readability
- Maintainability
- Network utilization
- Database utilization
- Security concerns

Question 1.1-3

Binary Search Trees:

- Strength: Searching for an element is fast - $O(\log N)$
- Limitation: Insertions/Deletions are relatively slow - $O(\log N)$ time compared to $O(1)$ of Linked Lists.

Linked Lists:

- Strength: Can insert a new element at any place and they do not require sequential space in memory
- Limitation: Require additional memory for links

Question 1.1-4

How are the shortest-path and traveling salesman problems similar? How are they different?

They are similar in that each algorithm has to find the shortest path between nodes.

They are different in that the shortest-path requires a path between two nodes, while the traveling salesman requires a path between all nodes. In addition, the TSP must return to the starting node.

Question 1.1-5

Only the best solution will do when searching for a customer in a database. The algorithm must return that specific customer, and not one that is "close enough."

A "good enough" solution can be used when creating a recommendation algorithm for customers. It would be impossible to create a perfect recommendation system where all recommendations match each customer perfectly one hundred percent of the time, but if the majority of the recommendations fit the majority of the customers most of the time, the algorithm can be considered a viable solution.

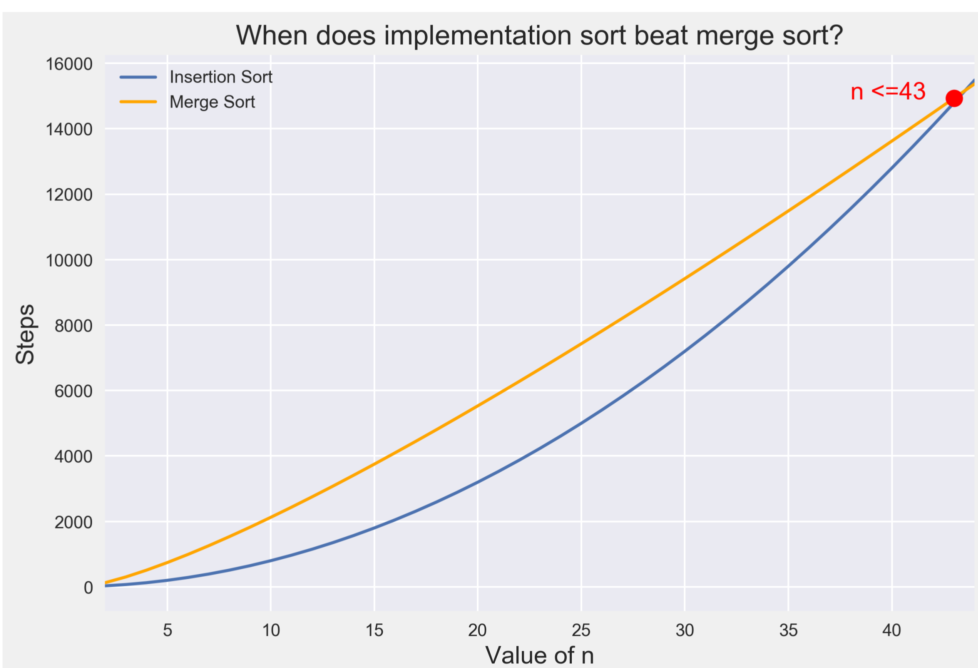
Exercises 1.2

Question 1.2-2

For inputs of size n, insertion sort runs in $8n^2$ steps, while merge sort runs in $64\ n \lg n$ steps. For which values of n does insertion sort beat merge sort?

Equation: $8n^2 \leq 64n \lg n$
Solution: $n \leq 43$ (see Python code)

```
   Insertion Sort  Merge Sort
n
2              32         128
3              72         304
4             128         512
5             200         743
    Insertion Sort  Merge Sort
n
40           12800       13624
41           13448       14058
42           14112       14494
43           14792       14933
44           15488       15373
```

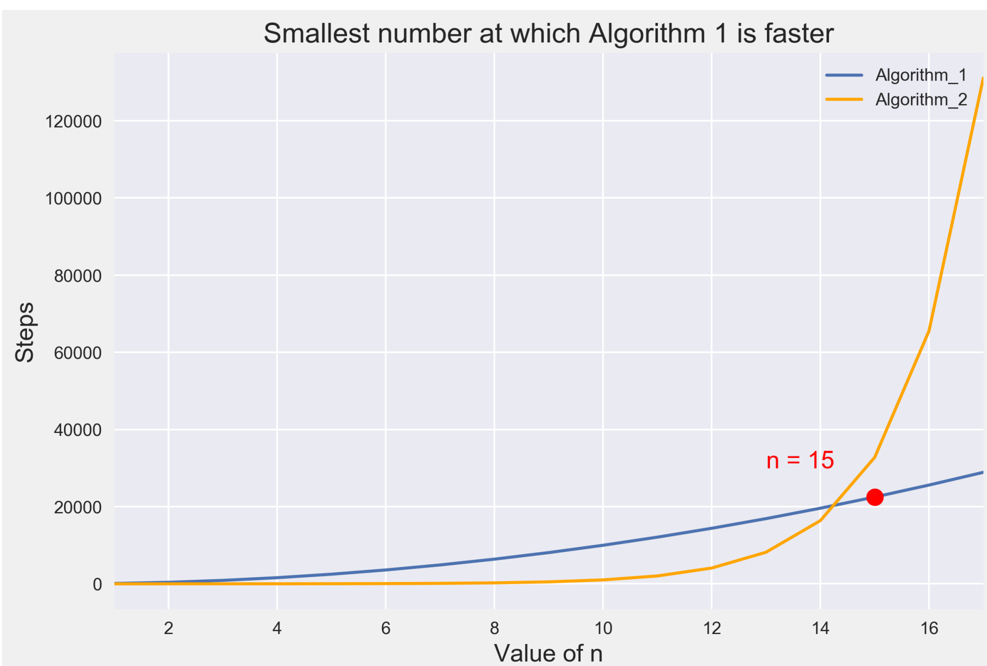


Question 1.2-3

What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n on the same machine?

Equation: $100n^2 \leq 2^n$
Solution: $n \geq 15$ (see Python code)

n	Algorithm_1	Algorithm_2
1	100	2
2	400	4
3	900	8
4	1600	16
5	2500	32
6	3600	64
7	4900	128
8	6400	256
9	8100	512
10	10000	1024
11	12100	2048
12	14400	4096
13	16900	8192
14	19600	16384
15	22500	32768
16	25600	65536
17	28900	131072



Problems

Problem 1-1

For each function $f(n)$ and time t in the following table, determine the largest size n of a problem that can be solved in time t, assuming that the algorithm to solve the problem takes $f(n)$ microseconds.

(For solutions to n log n and $n!$ refer to code in Python.)

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
lg n	2^{10^6}	$2^{6\cdot 10^7}$	$2^{36\cdot 10^8}$	$2^{864\cdot 10^8}$	$2^{2592\cdot 10^9}$	$2^{31536\cdot 10^9}$	$2^{3.1536\cdot 10^{15}}$
$\sqrt{n}$	10^{12}	$36 \cdot 10^{14}$	$1296 \cdot 10^{16}$	$746496 \cdot 10^{16}$	$6718464 \cdot 10^{18}$	$994519296 \cdot 10^{18}$	$9.9583 \cdot 10^{30}$
n	10^6	$6 \cdot 10^7$	$36 \cdot 10^8$	$864 \cdot 10^8$	$2592 \cdot 10^9$	$31536 \cdot 10^9$	$3.1536 \cdot 10^{15}$
$n \lg n$	62746	2801417	133378058	$2.7551 \cdot 10^9$	$7.1871 \cdot 10^{10}$	$7.9763 \cdot 10^{11}$	$6.8611 \cdot 10^{13}$
n^2	1000	7745	60000	293938	1609968	5615692	56176151
n^3	100	391	1532	4420	13736	31593	146645
2^n	19	25	31	36	41	44	51
$n!$	9	11	12	13	15	16	17