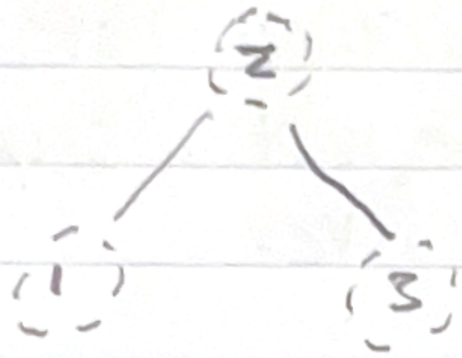
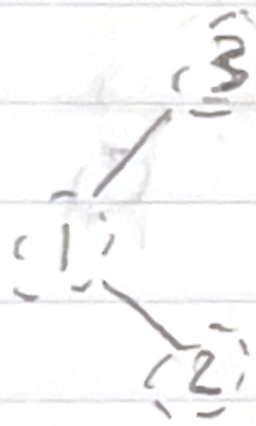
Insert

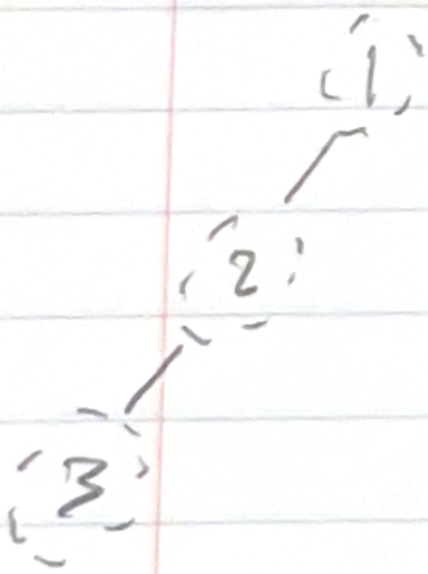
Best Case



Avg. Case



Worst Case



1) Form a balanced BST containing 1000 #'s

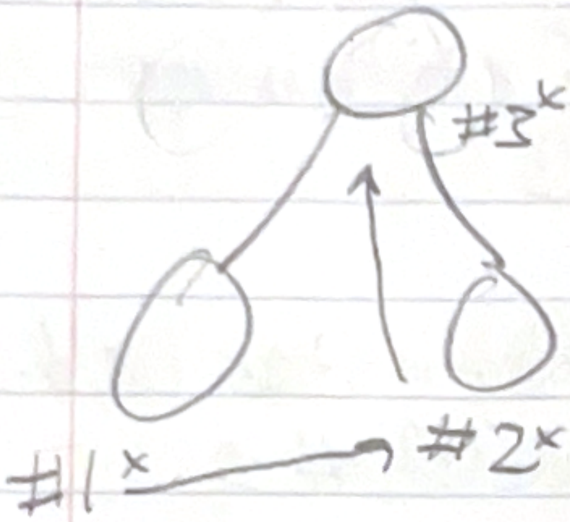
2) Make a tree with shuffled #'s

3) Generate a linked list tree with 1000 #s

Compare elapsed time. Graph accordingly with intervals increasing by powers of 10.

Delete

Best Case



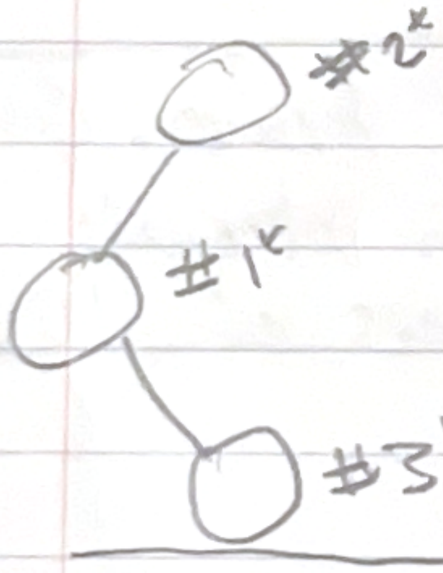
Delete order:

- Delete leaves first so as to not disrupt BST structure

Generation:

- Balanced

Average Case



Delete Order:

- Random

Generation:

- Shuffled

Delete order:

- sorted ascending

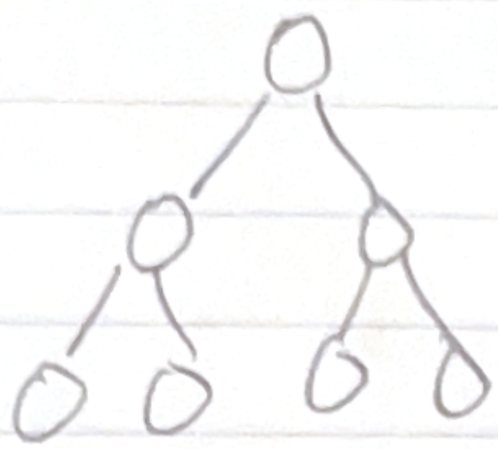
Generation:

- numerical order

* Tree generation must be done prior

Search (no hit)

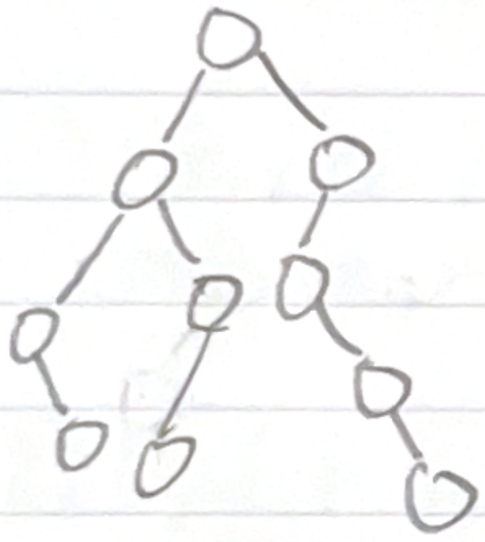
Best Case



1) pick a number to "search" for that is not in the tree

2) Perform 4000 total searches and compare elapsed time between cases.

Avg Case



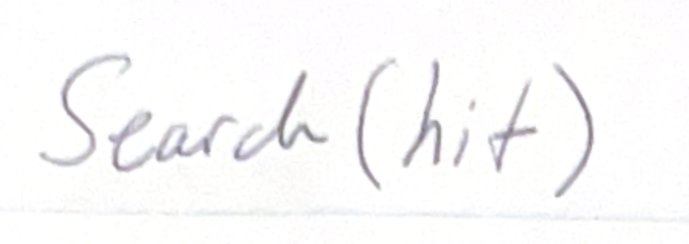
*Random

Worst Case

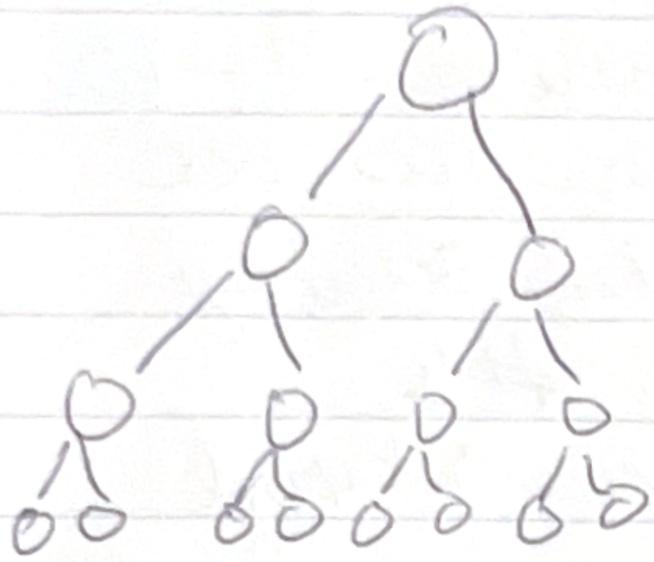
*Linked List

Big O #1

Search (hit)



Balanced (Best)



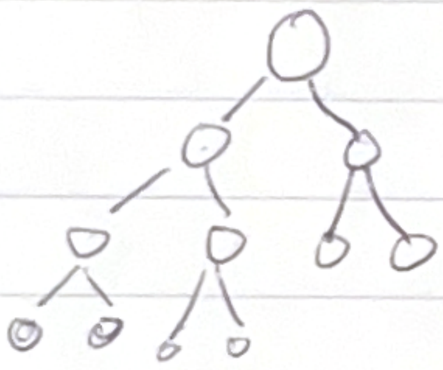
1) perform 100,000 searches randomly selecting a tree #

2) increase tree size by using powers of 10

3) graph the results

Random

(average)



Shuffle #'s

1) [illegible]

2) [illegible]

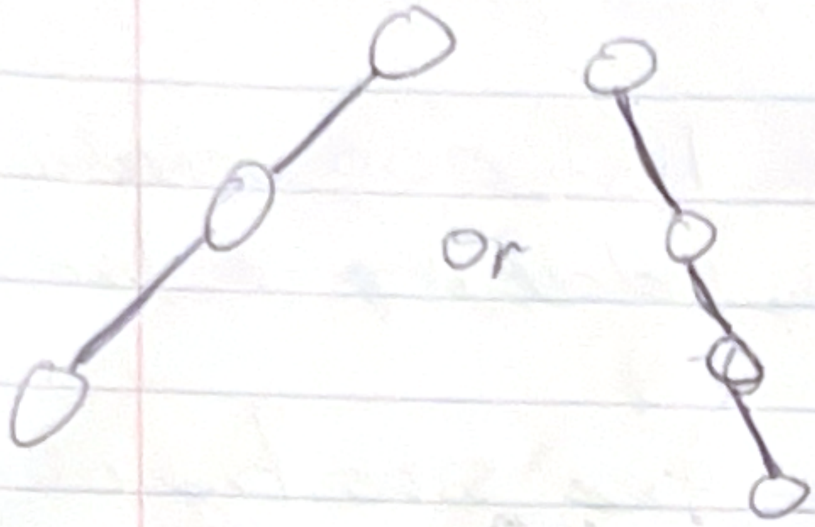
3) Perform 100,000 searches selecting random #'s

* add #'s to array and tree

4) Increase intervals by powers of 10's graph results

See fig. 1 for Balanced Generation

Linked List (Worst)

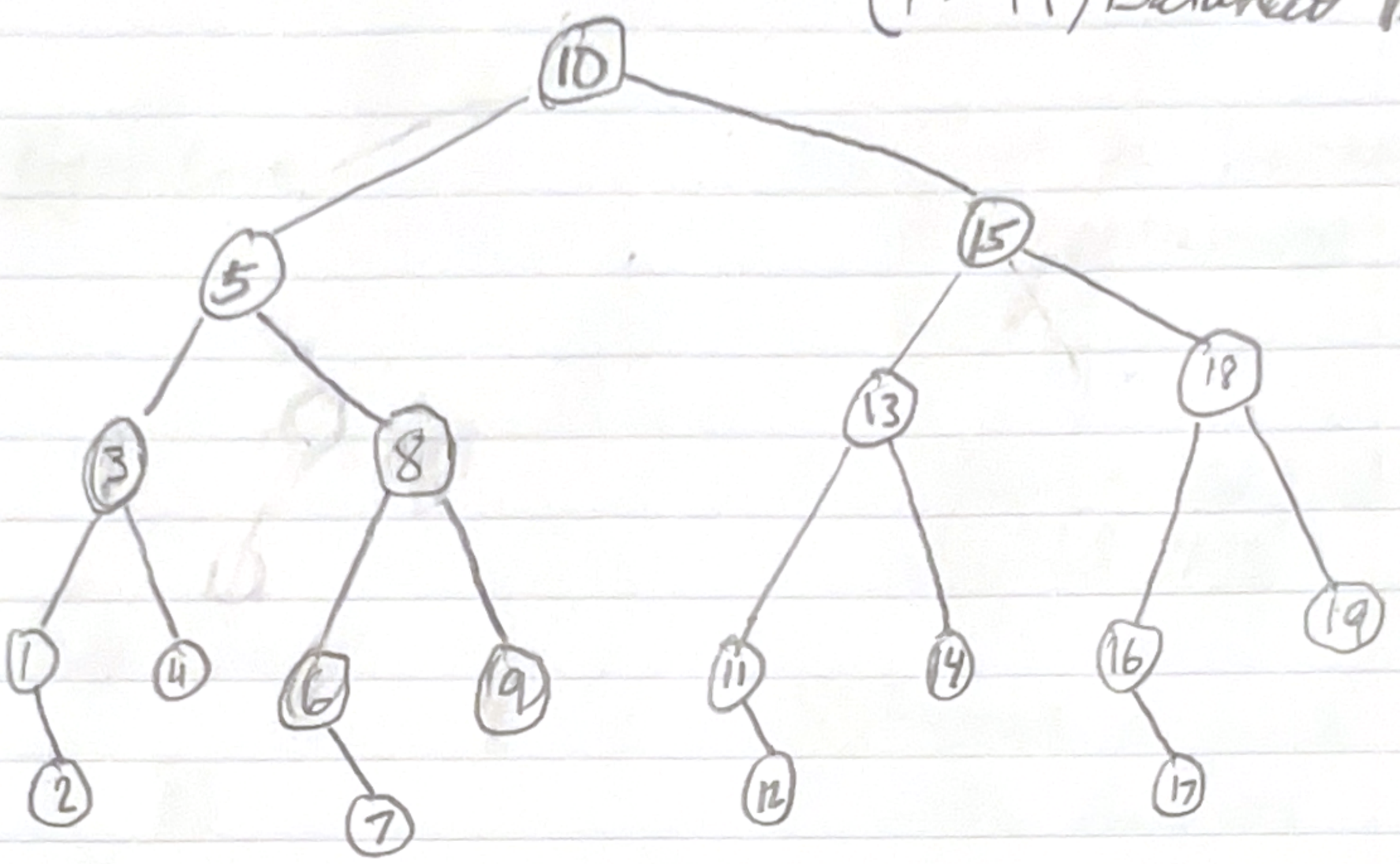


*In order

1) Take list of #'s in between 1–10
2) perform 100,000 searches selecting random #'s
3) Increase interval by powers of 10

Figure 1

(1-19) Balanced BST



1) Take the array of numbers. Find midpoint. This will be the parent

2) Split into 2 arrays. Exclude the midpoint. Call method for each side using either array

3) Repeat steps until size == 0

How to make a balanced BST