CSE 250 Recitation

March 4-5 - Recursion

Binary Search

The binary search algorithm let's us effective search a List

To work correctly and efficiently the List must:

- Be sorted
- Allow constant time random access (ie an Array)

It works by comparing our target to the midpoint, then searching **only** the left half or the right half

Binary Search Code

```
int binarySearch(ArrayList<T> list, T target) {
  return binarySearch(list, target, 0, list.size() - 1);
}

int binarySearch(ArrayList<T> list, T target, int start, int end) {
  if(start == end) { return start; }
  int mid = (start + end) / 2;
  T guess = list.get(mid);
  if(guess.equals(target)){ return mid; } // We found our target!
  else if(target.compareTo(guess) < 0) {  // Target is in the left half
    return binarySearch(list, target, start, mid);
  } else {                                // Target is in the right half
    return binarySearch(list, target, mid+1, end);
  }
}
```

Runtime Growth Function

$$T(N) = \begin{cases} T\left(\frac{N}{2}\right) + \theta(1) & if\ target\ is\ not\ found \\ \theta(1) & else \end{cases}$$

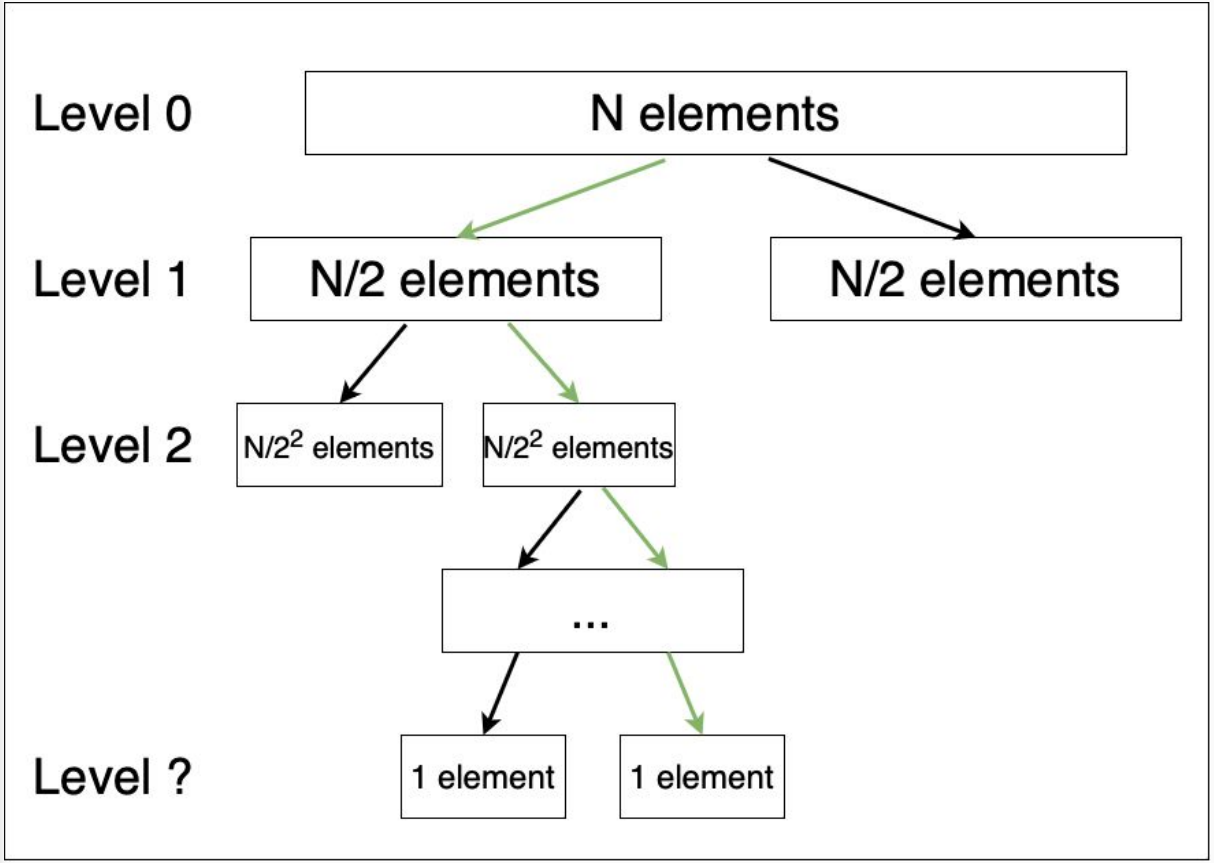
Runtime Growth Function

$$T(N) = \begin{cases} T\left(\frac{N}{2}\right) + \theta(1) & \mathit{if\ target\ is\ not\ found} \\ \theta(1) & \mathit{else} \end{cases}$$

Exercise: Draw the recursion tree for this growth function

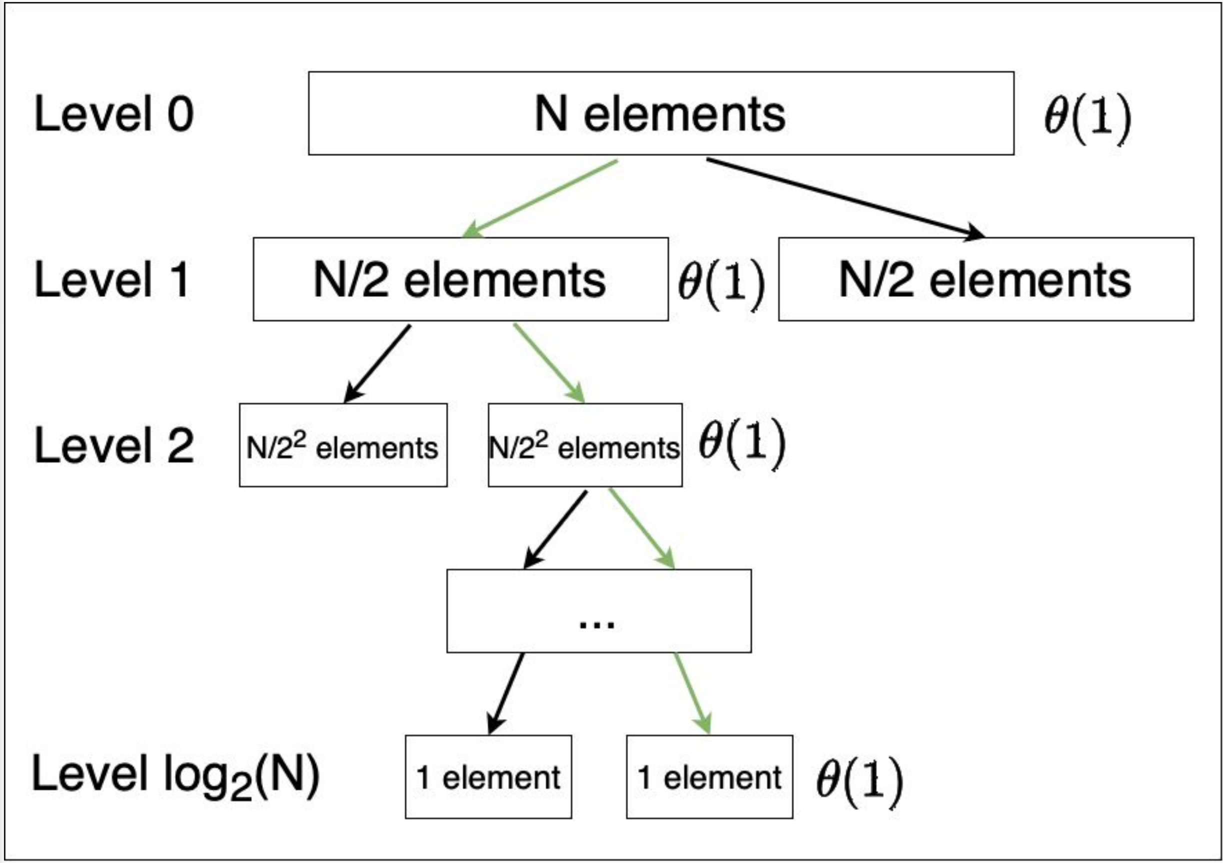
Hypothesis

- To form your hypothesis for the runtime of the algorithm, you should ask yourself two questions:
  - How much work are you doing on each level?
  - How many levels are there?



Hypothesis

- We are doing a constant amount of work on each level
- We are guaranteed to have a maximum of $\log_2(N)$ levels



Hypothesis Summation

$$\sum_{i=1}^{O(\log_2(N))} \theta(1)$$

Inductive Hypothesis

$$T(N) \in O(\log_2(N))$$

Base Case

$$T(1) \leq c \cdot \log_2(1)$$

Base Case

$$T(\overset{2}{\cancel{1}}) \leq c \cdot \log_2(\overset{2}{\cancel{1}})$$

Inductive Case

Assume: $T\left(\frac{N}{2}\right) \leq c \cdot \log_2\left(\frac{N}{2}\right)$

Show: $T(N) \leq c \cdot \log_2(N)$