Analysis of Algorithms

CTEC 243

Analysis of Algorithms

- Criteria of Analysis
  - Space - memory required to solve the problem
  - Time - length of execution time
- Ways to Measure Efficiency
  - Timing execution - shows how long the program runs with that input, not the general efficiency
  - Count the steps - generalizes the speed of an algorithm based on what it does, not real world usage

Computational Problems and Basic Steps

- Computation Problem - problem to be solved with an Algorithm
  - Instances - real world examples of the problem
- Basic Step - a step executed by an algorithm
  - Run in constant time, doesn't take longer with bigger data sets
  - Example: Swap the elements in positions k and k+1

A little weirdness and generalization:

- Because Basic Steps take a specific amount of time but can vary based on hardware, we don't sweat the details
- Any series of Basic Steps is counted as taking the time of 1 Basic Step
  - So 500n is viewed the same as n

Complexity of Algorithms

```
sum = 0;
k = 0;
while K < n do
        sum = sum + a[k]
        k = k + 1
end while
```

In a worst-case scenario, this code runs n times


THE
BIG

Big O Notation

- Let’s not worry too much about the math…
- Assume that the data sets we’re working on are HUGE
- We think about the worst-case Basic Steps for the algorithm
- Common O()’s:
  - O(1)
  - O(log n)
  - O(n)
  - O(n log n)
  - O(n^2)

Common Errors

- Using an inefficient sort or search algorithm on a large array
- Forgetting to sort the data in an array before using binary search
- Using timing to determine the efficiency of an algorithm