CIT 596: ALGORITHMS & COMPUTATION

Counting Steps: Models of Computation

Measuring Algorithmic Speed

- Naïve idea: Implement the algorithm, generate some sample inputs, and measure time in seconds.
  - Depends on hardware and implementation details.
  - Depends on the sample inputs chosen.
- Instead, we analyze algorithms by counting the operations they perform
  - Robust to changes in hardware and implementation.
- We will almost always look at the **worst-case** number of operations for each input size.
  - Our algorithms should run fast on ***every*** input.

Counting Swaps in Insertion Sort

When there are loops, sum over all iterations.

```
INSERTIONSORT(A)
    for i = 1 to length(A)
        j = i
        while j > 1 and A[j − 1] > A[j]
            swap A[j] and A[j − 1]
            j = j − 1
```


- One swap per iteration of the **while** loop.
- At most $i - 1$ iterations of the **while** loop in the i^{th} iteration of the **for** loop.

Letting $n = length(A)$, the total number of swaps is at most

$$0 + 1 + \cdots + (n-1) = \frac{(n-1) \cdot n}{2}.$$

Counting Moves in HANOI

To analyze recursive algorithms, we use **recurrence relations**.

```
HANOI(A, B, C, n)
    if n == 1
        move the top ring on A to C
    else
        HANOI(A, C, B, n − 1)
        HANOI(A, B, C, 1)
        HANOI(B, A, C, n − 1)
```


Let $T(n)$ be the number of moves used in HANOI(A, B, C, n).

- $T(1) = 1$.
- For $n \geq 2$,
$$T(n) = 2T(n-1) + T(1).$$
- Closed form (see self-study notes):
$$T(n) = 2^n - 1.$$

Computational Steps

- Here, we counted just one type of operation for each algorithm.
- Usually, we will count *all* computational steps or primitive operations.
- E.g., arithmetic, comparison, or variable assignment with primitive data types in Java.
- But this is not the only possible model of computation!