CS 271 Sept 7, 2017

Insertion Sort

We did a demonstration of how insertion sort works, using characters written on coffee cups. Insertion sort is very similar to the algorithm used in the moveLetters function.

Selection Sort

We did a demonstration of how selection sort works, using those same coffee cups.

Both sorts require two loops (nested loops). Both loops use an index to an array element.

Determining the Complexity of an Algorithm (Big O notation)

The complexity of an algorithm is an estimate of how long an algorithm will take to finish.

If we have nested loops, multiply the number of iterations of the outer loop times the number of iterations for the inner loop.

Let's say we have N elements in the array and we want to estimate the complexity of selection sort.

Outer loop: How many iterations? N - 1

Assume that we're on iteration x in the outer loop.

Inner loop: How many iterations? N - x + 1

Multiply (N - 1) * (N - x) = N^2 - xN - N + x

Rough Estimate: we only care about the largest power of N

Complexity for Selection Sort **O**(N^2) - read "big Oh of n squared"

Most common sorts are O(N^2)

Searching

Only two types of search algorithms:

1) linear, slow **O**(N)
2) binary, fast **O**($\log_2$ N) Array must be sorted first.

Practice with base 2 logarithms.

N = 1000, binary search ~ 10 comparisons (log base 2 of 1000 = ~10)

Note: A loop must iterate an integer number of times. We cannot repeat a loop 2.5 times.

1024 = 2^{10} This is the number of bytes in a kilobyte.

1024 * 1024 = The number of bytes in a megabyte.

2^{20} = ??

Complexity of the moveLetters function $O(N^2)$