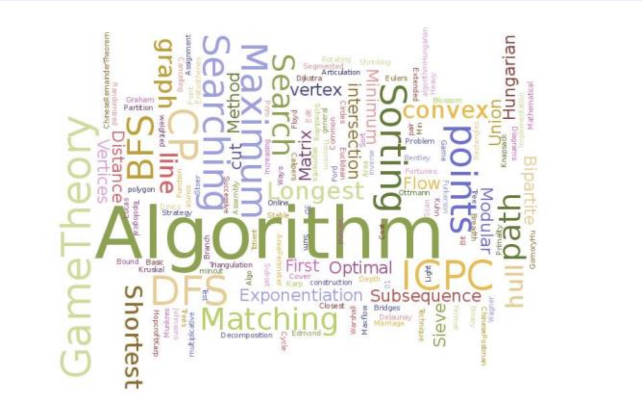
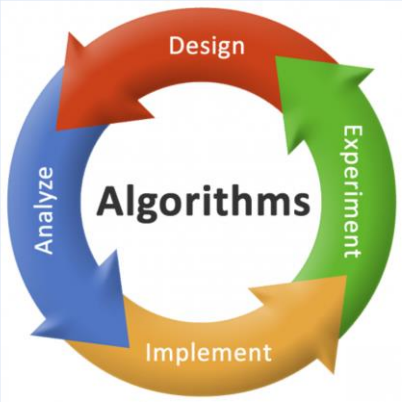
Brute Force

COP 3503
Fall 2021
Department of Computer Science
University of Central Florida
Dr. Steinberg





Introduction

- Now that you have the mathematical background, we can begin learning algorithm designs.
- The first design is most likely a strategy you have used in previous courses when solving problems on the spot.
- **Brute Force** – is a straightforward approach to solving a problem, usually directly based on the problem statement and definitions of the concepts involved.
- Also known as an exhaustive search.
- For example:
  - Computing an exponent $a^n = \overset{\text{n times}}{\overline{a * a * a \ldots a * a * a}}$

Introduction cont.

- While Brute Force may not be the best approach for algorithm designs it can be used as a yardstick to compare more efficient alternative for designing a different algorithm to solve the problem.

Exponent Algorithm Using Brute Force

```
ComputeExponent(b,n)
for i = 1 to n
    b = b * b
return b
```

Pros and Cons of Brute Force

- Pros
  - Easy to implement
- Cons
  - Doesn’t always produce the best running time

Remember The Sorting Problem Again?
Lets look at one of the solutions.
Bubble Sort

Bubble Sort

```
BubbleSort(A,n)
for i = 1 to n
        for j = 1 to n – i
                if A[j + 1] < A[j]
                        swap(A[j], A[j + 1])
```

$$\sum_{i=1}^{n}\sum_{j=1}^{n-i} 1 = \sum_{i=1}^{n}(n - i - 0 + 1) = \sum_{i=1}^{n} n - i - 1 = \frac{(n-1)n}{2} = \frac{n^2 - n}{2}$$

Running Time: $\Theta(n^2)$