Algorithms

why study algorithms?

- not be ~~confer~~ constrained by programming language.
- better hardware is not always the answer
- to learn what has already been done
- to be exposed to new and interesting problem solving techniques
- writing algorithms is a skill

COMMON ELEMENT(A)

```
1  while true
2        i = Random Integer in [1, n]
3        j = Random Integer in [1, n]
4        if i ≠ j and A[i] = A[j]
5              return A[i]
```

$\left(\frac{3}{4}\right)^{10} \approx .056$

Comparing Algorithms

What makes a good algorithm?

- Efficiency
  - Time efficiency ←
  - Space efficiency
- Simplicity
  - Fewer bugs
  - Maintainability
  - Ease of implementation
- Generality
  - variations on a given problem
  - variations on possible inputs

Asymptotic Analysis

- we focus on analyzing the behavior of algorithms as the input size approaches infinity.
- How does the time required change as the input size grows?
  - real time will vary between machines
  - count CPU cycles instead?
  - count the total number of operations for an input of some size?
- we count just the most frequent operation

Def: Basic operation

- the operation contributing the most to the running time.

```
FIND DUPLICATES(A)
1  for i = 1 to n
2      for j = 1 to n
3          if i ≠ j and A[i] = A[j]
4              return true
5  return false
```

Basic operation : A[i] = A[j]

How many comparisons occur on a list of size n?

Best case: first two elements match

1 comparison

Worst case: no duplicates

$\approx n^2$

The running time can be loosely approximated

$$Time(n) \approx c_{op} C(n)$$

c_{op} ← cost of the basic op

$C(n)$ ← # of time the basic op is performed

IMPROVED FIND DUPLICATES(A)

```
1  for i = 1 to n − 1
2      for j = i + 1 to n
3          if A[i] = A[j]
4              return true
5  return false
```

Worst Case:

$$C(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1$$

$$= \sum_{i=1}^{n-1} (n-i)$$

$$= \sum_{k=1}^{n-1} k$$

$$= \frac{n(n-1)}{2}$$

$$= \frac{1}{2}(n^2 - n) \approx \frac{1}{2} n^2$$

How will the running time change if we double the input size?

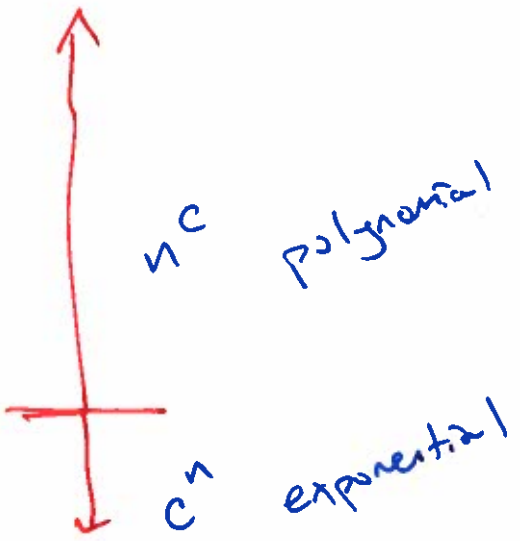
$$\frac{\text{Time}(2n)}{\text{Time}(n)} = \frac{\cancel{\text{Cop}}\, C(2n)}{\cancel{\text{Cop}}\, C(n)}$$

$$= \frac{\cancel{\frac{1}{2}}\,(2n)^2}{\cancel{\frac{1}{2}}\,(n)^2}$$

$$= \frac{4n^2}{n^2}$$

$$= 4$$

Growth Rate	**Name**
1	Constant
$\log n$	Logarithmic
n	Linear
$n \log n$	Superlinear
n^2	Quadratic
n^3	Cubic
2^n	Exponential
$n!$	Factorial



Algorithm Speed	Old Machine	New Machine
$\lg n$	1000000	≈ 1100000000000
$\sqrt{n}$	1000000	4000000
n	1000000	2000000
$n \lg n$	1000000	≈ 1920000
n^2	1000000	1414213
n^3	1000000	1259921
2^n	1000000	1000001
$n!$	1000000	1000000