Algorithm Analysis

- A <u>problem</u> for computer must be defined precisely by its input and desired output.
- Ex: sort an array
- Input: array and a way to compare the elements
- Output: Sorted array
- Ex: Compute the factorial of a positive integer
- Input: a positive integer n
- Output: n!

Note that we can use the tools of discrete math to define these inputs and outputs precisely

- A solution is some method of taking an arbitrary input and computing the output with the desired properties defined by the problem
- An <u>Algorithm</u>, is a sequence of steps you can perform to get the desired output from the input.

Sort(A)

This is an example of a pseudo code. This

is a way of communicating an algorithm to another human. It is written in a way that is easier to understand by a human. It has a mix of precise and unambiguous

notations and words.

Let S = the set of all permutations of A For x in S:

if x is sorted:

return x

1. Is this algorithm correct?

2. Does this algorithm work efficiently? (runtime)

How to measure the runtime?

1. Idea 1: Implement the algorithm, run it and then time it.

Depends on the software, hardware, operating system, programming language, etc.

Implementation takes time and money.

How to determine which input to run?

How do we know the inputs that we selected to run reflect the runtime?

2. Idea 2: Find a function that reflects the runtime in terms of the input size

Runtime: number of primitive operations (arithmetic operations, logical operations, variable retrieval, variable assignment, etc.)

For i = 1 to n: Sum = 0 ← Algorithm 1 sum = sum + 1 variable (1) raticable (1)

assignment (1)

logical operation (1)

(3 operations)

(3 operations)

For i = 1 to n: For i = 1 to n: sum = sum + 1

 $f_1(n) = 1 + (1 + s.n) \times 3 = 1 + 3 + 15n = 15n + 4$

sum = sum + 1

For
$$i = 1$$
 to n :
For $j = 1$ to n :

103 i aco l'operation (2) pological operation (2)

variable assignment(1)

sum = 3n Else

(+2+(1+(2+(1+(2+5)n)))

$$= q_{N} + N + 3$$

f(N) =

Algorithm 1

$$Sum = 0$$

For
$$i = 1$$
 to n:

$$sum = sum + 1$$

For
$$i = 1$$
 to n:

$$sum = sum + 1$$

$$sum = sum + 1$$

For i = 1 to n:

Variable assignment

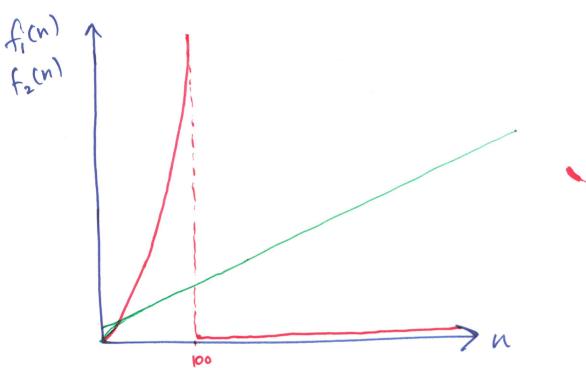
Each loop: Assigns i

1 + (1 + 4*n)*3 = 4 + 12n

$$f_4(n) = 12n + 4$$

Given $f_i(n) = 15 n + 4$, $f_2(n) = \begin{cases} 9n^2 + n + 3 & n < n \\ 6 & n > 1 \end{cases}$

What is best algorithm?



red: fz(n)
green: f,(n)