MA/CSSE 473
Day 02

Algorithms Intro
Numeric Algorithms

Brainstorm

- What is an algorithm?
- In groups of three, try to come up with a good definition.
- Goal: Short but complete
- In a few minutes I'll have some of you write them on the whiteboard.

Levitin picture

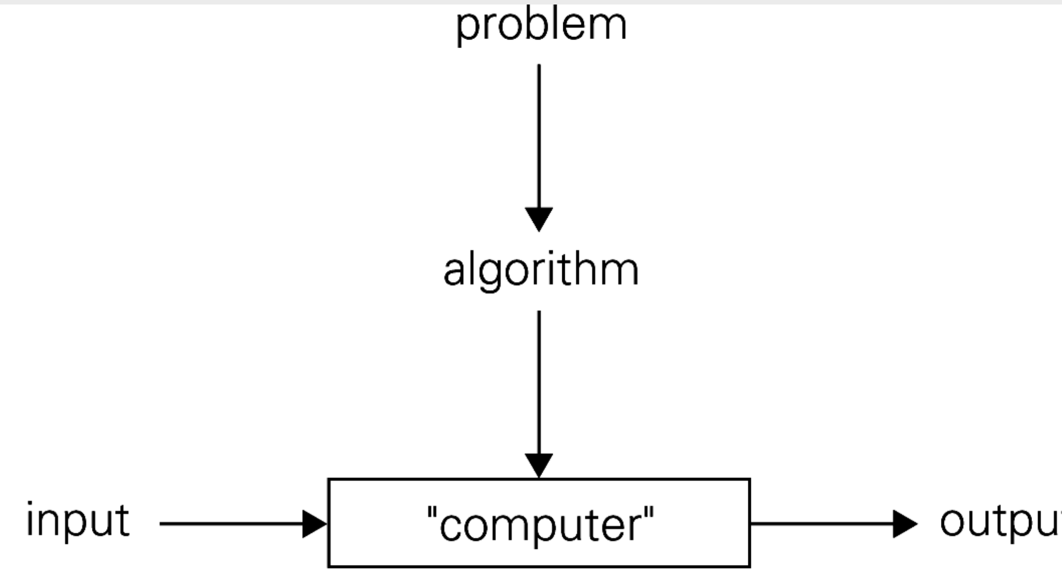

FIGURE 1.1 Notion of algorithm

Write an algorithm …

- … based on the schedule page for this course
- Input: A session number (1 .. 40)
- Output: A number representing the day of the week.   0 represents M, 1 T, 2 R, 3 F.
- Write it (or perhaps more than one) with your group.

Algorithm design Process

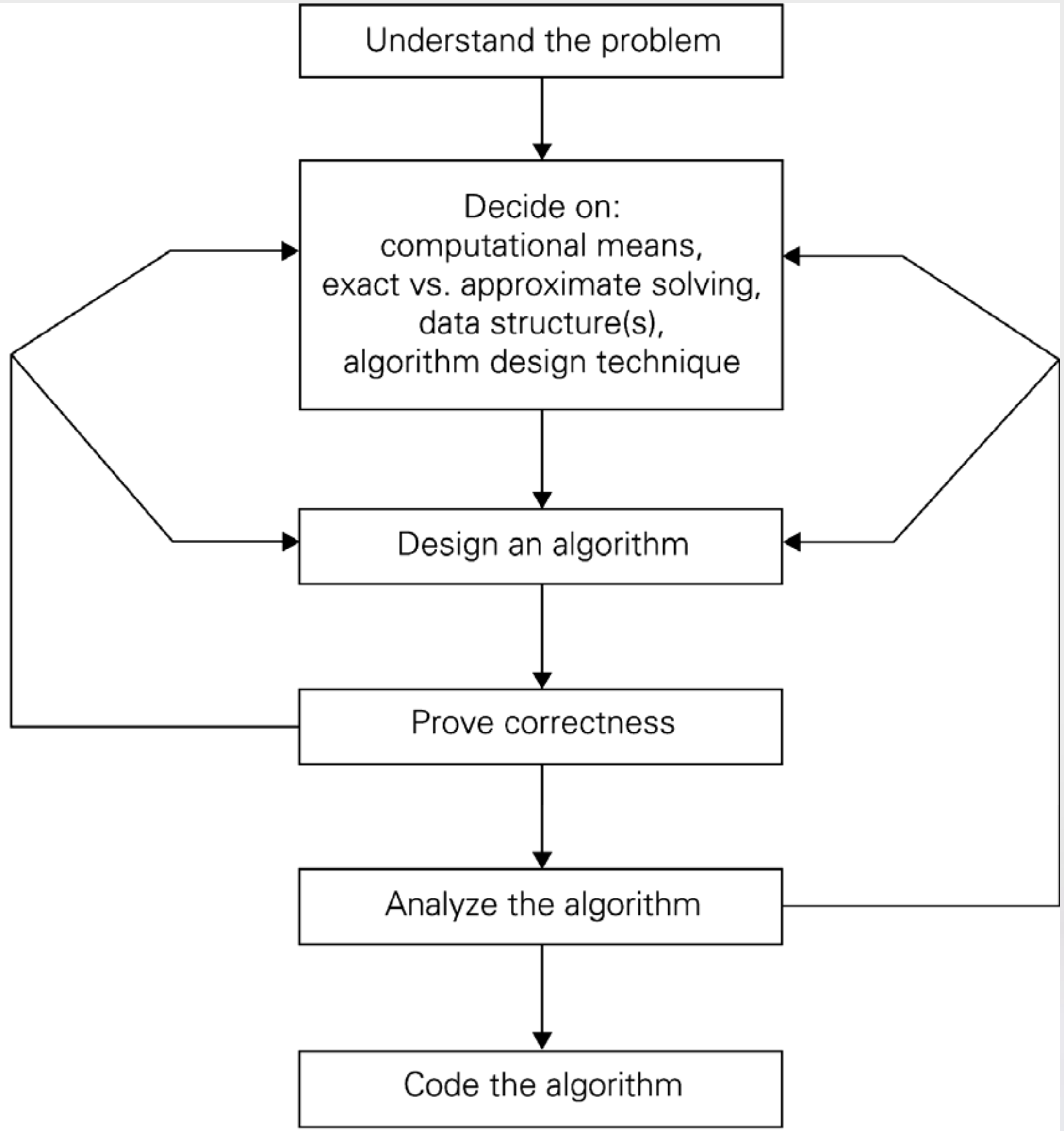

FIGURE 1.2 Algorithm design and analysis process

Fibonacci Numbers

- F(0) = 0, F(1) = 1, F(n) = F(n-1) + F(N-2)
- Sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …
- Straightforward recursive algorithm:

```
def fib1(n):
    if n==0:
        return 0
    if n==1:
        return 1
    return fib1(n-1) + fib1(n-2)

print fib1(6), fib1(7), fib1(8)
```

- Correctness is obvious. Why?

Analysis of the Recursive Algorithm

- What do we count?
  - For simplicity, we count basic computer operations
- Let T(n) be the number of operations required to compute F(N).
- $T(0) = 1$, $T(1) = 2$, $T(n) = T(n-1) + T(n-2) + 3$
- What can we conclude about the relationship between T(N) and F(N)?
- How bad is that?
- How long to compute F(200) on an exaflop machine (10^18 operations per second)?
  - http://slashdot.org/article.pl?sid=08/02/22/040239&from=rss

A Polynomial-time algorithm

- ```
  def fib2(n):
      nums = [0]*(n+1)
      nums[0] = 0
      nums[1] = 1
      for i in range(2, n+1):
          nums[i] = nums[i-1] + nums[i-2]
      return nums[n]
  ```
- Correctness is obvious because it again directly implements the Fibonacci definition.
- Analysis?
- Now (if we have enough space) we can quickly compute F(20000)


  ```

A Creative O(log N) Algorithm

- Let X be the matrix $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$
- Then $\begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = X \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$
- and $\begin{pmatrix} F_2 \\ F_3 \end{pmatrix} = X \cdot \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = X^2 \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}, and \begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = X^n \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$
- How many additions and multiplications of numbers are necessary to multiply two 2x2 matrices?
- If $n = 2^k$, how many matrix multiplications does it take to compute X^n?
- But there is a catch!