Introduction: Computing Runtimes

Daniel Kane

Department of Computer Science and Engineering
University of California, San Diego

Algorithmic Toolbox
Data Structures and Algorithms

Learning Objectives

- Describe some of the issues involved with computing the runtime of an actual program.
- Understand why finding exact runtimes is a problem.

Outline

1. Revisit Fibonacci
2. Other Things to Consider

Runtime Analysis

Function FibList(n)

create an array $F[0 \ldots n]$
$F[0] \leftarrow 0$
$F[1] \leftarrow 1$
for i from 2 to n:
  $F[i] \leftarrow F[i-1] + F[i-2]$
return $F[n]$

Runtime Analysis

Function FibList(n)

```
create an array F[0...n]
F[0] ← 0
F[1] ← 1
for i from 2 to n:
    F[i] ← F[i − 1] + F[i − 2]
return F[n]
```

$2n + 2$ lines of code. Does this really describe the runtime of the algorithm?

Individual Lines

```
Function FibList(n)
create an array F[0...n]
F[0] ← 0
F[1] ← 1
for i from 2 to n:
   F[i] ← F[i − 1] + F[i − 2]
return F[n]
```

Depends on memory management system.

Individual Lines

Function FibList(n)

create an array $F[0 \ldots n]$
$F[0] \leftarrow 0$
$F[1] \leftarrow 1$
for i from 2 to n:
  $F[i] \leftarrow F[i-1] + F[i-2]$
return $F[n]$

Assignment.

Individual Lines

Function FibList(n)

create an array $F[0 \ldots n]$
$F[0] \leftarrow 0$
$F[1] \leftarrow 1$
for i from 2 to n:
  $F[i] \leftarrow F[i-1] + F[i-2]$
return $F[n]$

Assignment.

Individual Lines

Function FibList(n)

create an array $F[0 \ldots n]$
$F[0] \leftarrow 0$
$F[1] \leftarrow 1$
for i from 2 to n:
    $F[i] \leftarrow F[i-1] + F[i-2]$
return $F[n]$

Increment, comparison, branch.

Individual Lines

Function FibList(n)

create an array $F[0 \ldots n]$
$F[0] \leftarrow 0$
$F[1] \leftarrow 1$
for i from 2 to n:
  $F[i] \leftarrow F[i-1] + F[i-2]$
return $F[n]$

Lookup, assignment, addition of big integers.

Individual Lines

Function FibList(n)

create an array $F[0 \ldots n]$
$F[0] \leftarrow 0$
$F[1] \leftarrow 1$
for i from 2 to n:
  $F[i] \leftarrow F[i-1] + F[i-2]$
return $F[n]$

Lookup, return.

Outline

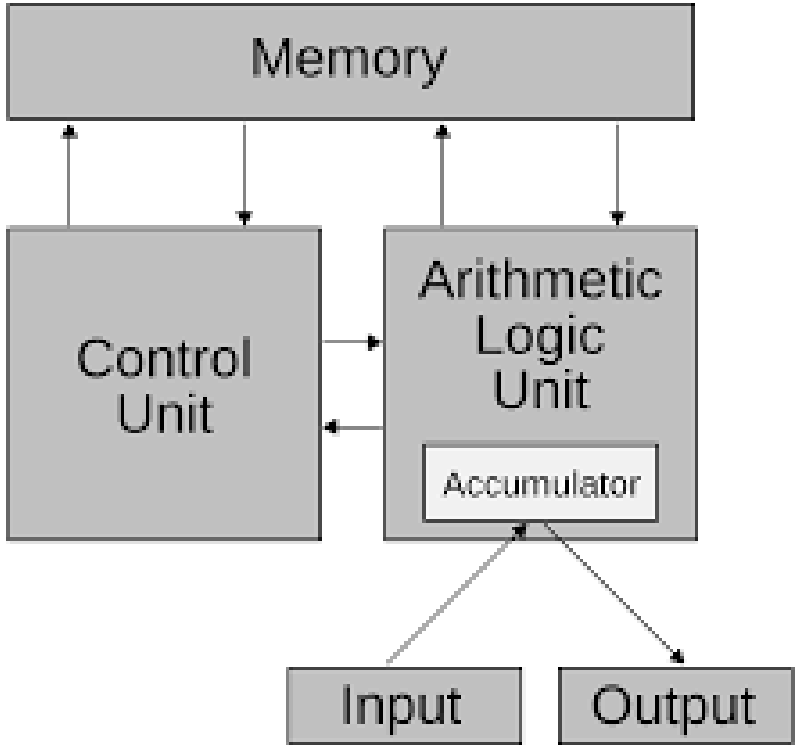
1. Revisit Fibonacci
2. Other Things to Consider

Computing Runtime

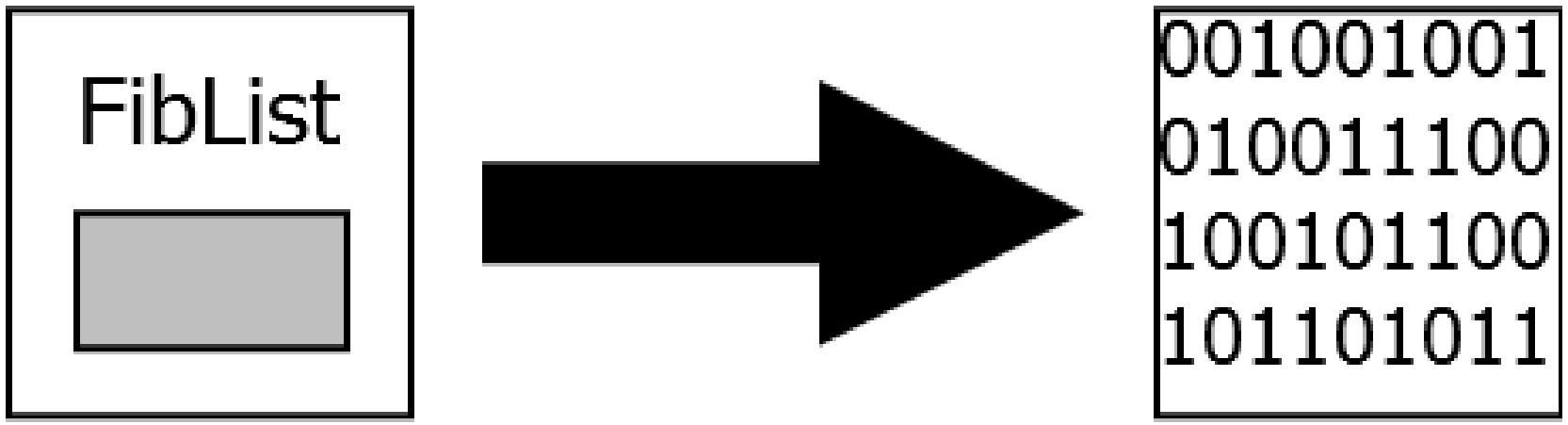
To figure out how long this simple program would actually take to run on a real computer, we would also need to know things like:

Speed of the Computer

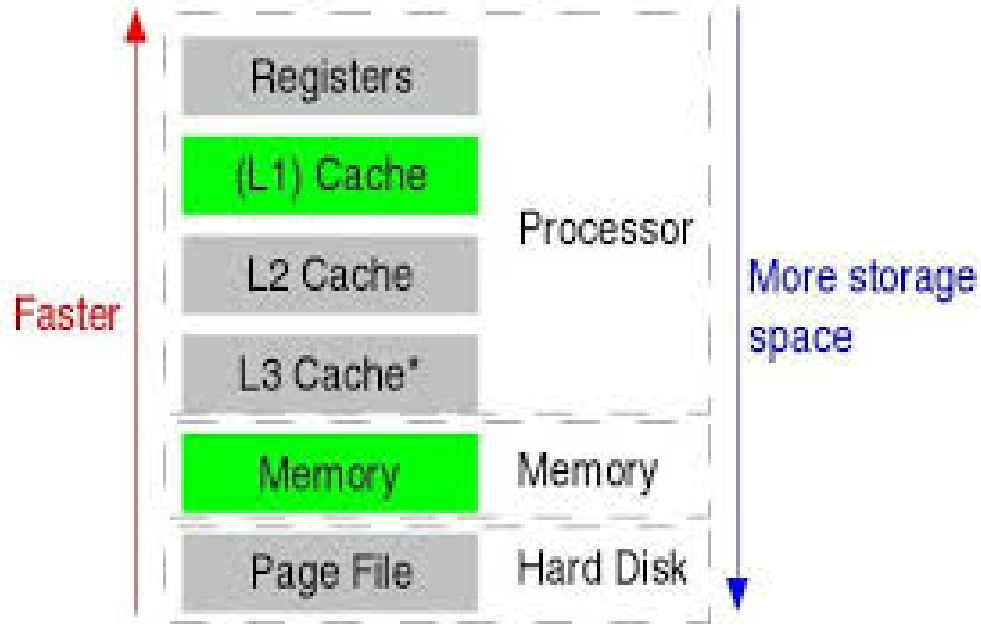
The System Architecture



The Compiler Being Used



Details of the Memory Hierarchy



Problem

- Figuring out accurate runtime is a huge mess

Problem

- Figuring out accurate runtime is a huge mess
- In practice, you might not even know some of these details

Goal

Want to:

- Measure runtime without knowing these details.
- Get results that work for large inputs.