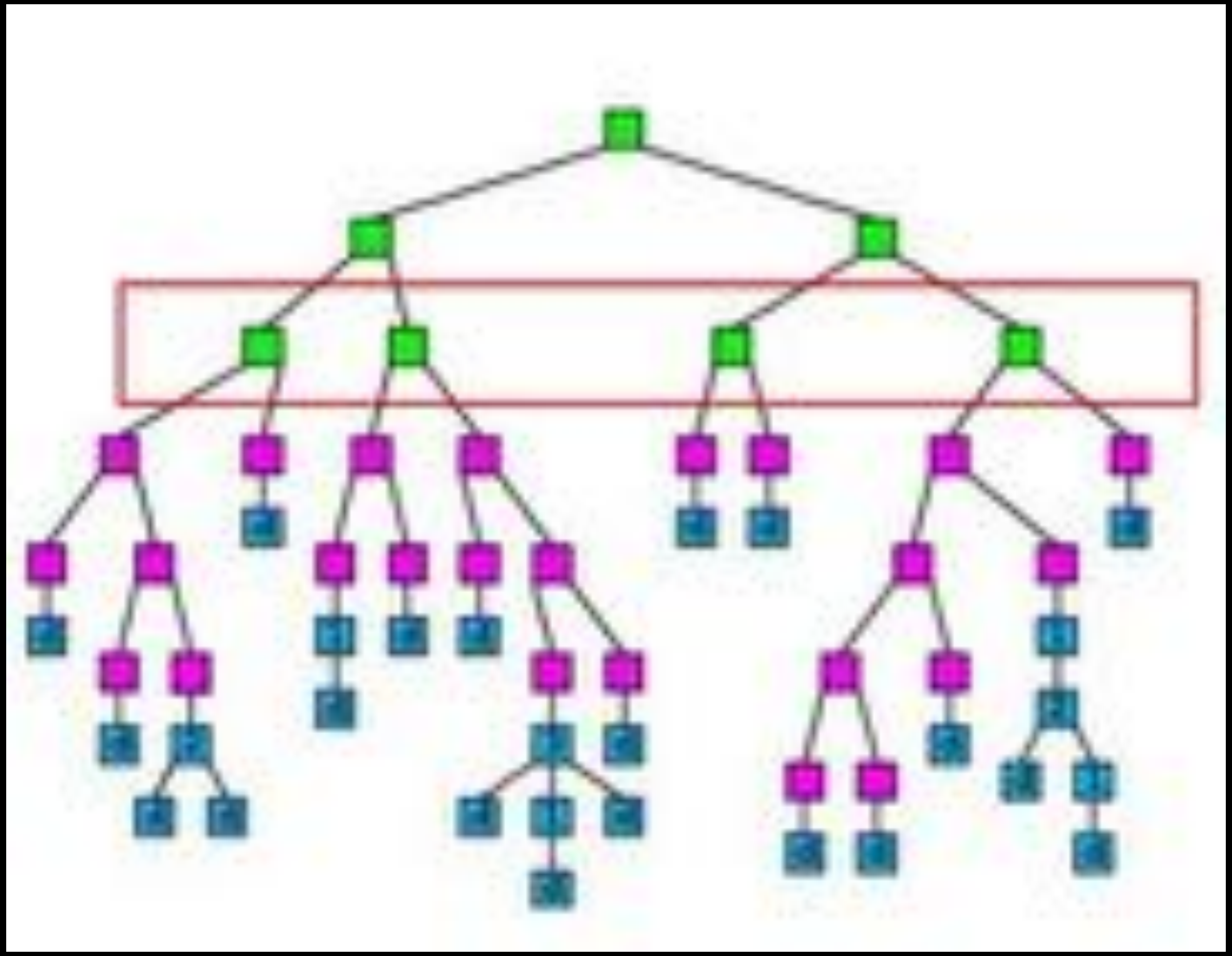
Parallelization and Algorithm Design



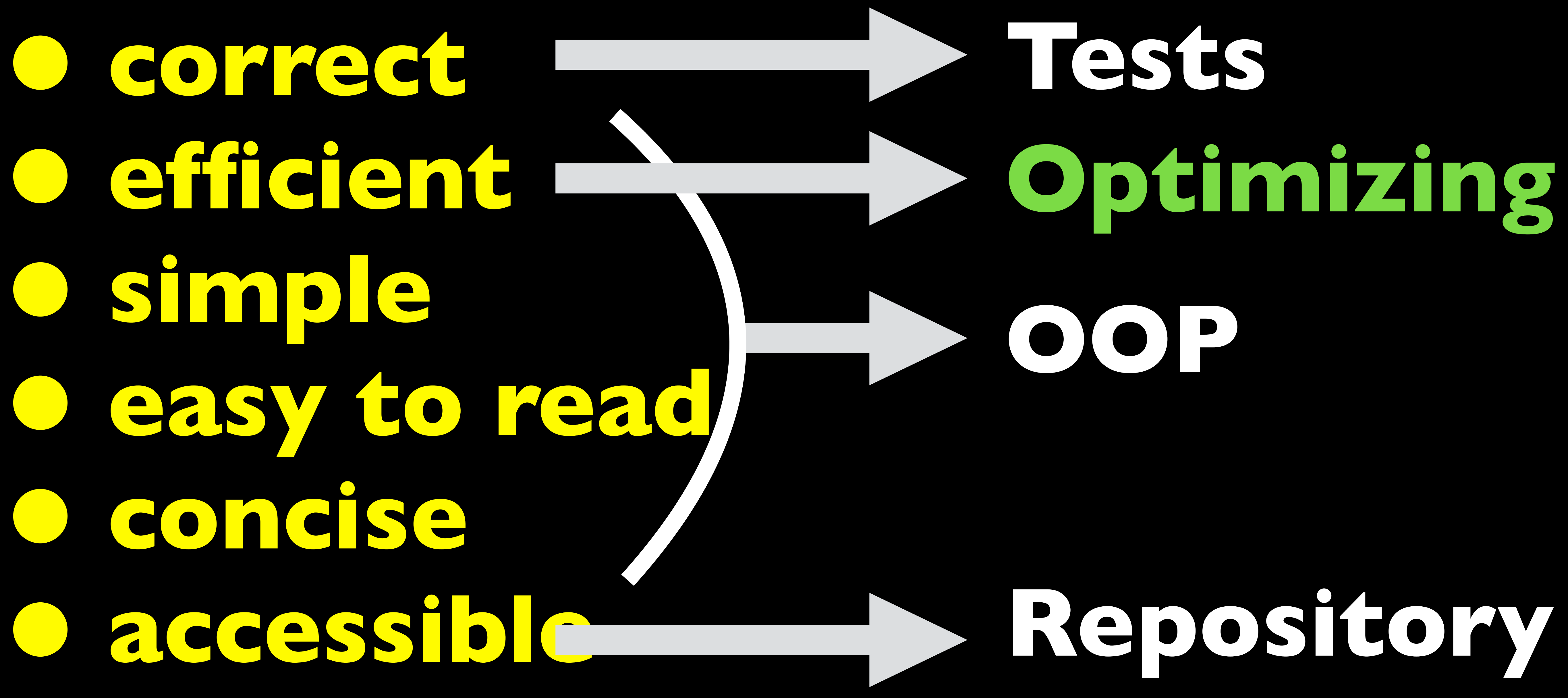
Cameron Hummels
NSF Fellow, Caltech

Code should*
be:

Code should* be:

- correct
- efficient
- simple
- easy to read
- concise
- accessible

Code should* be:



- correct
- efficient
- simple
- easy to read
- concise
- accessible

Tests

Optimizing

OOP

Repository

Algorithm Design

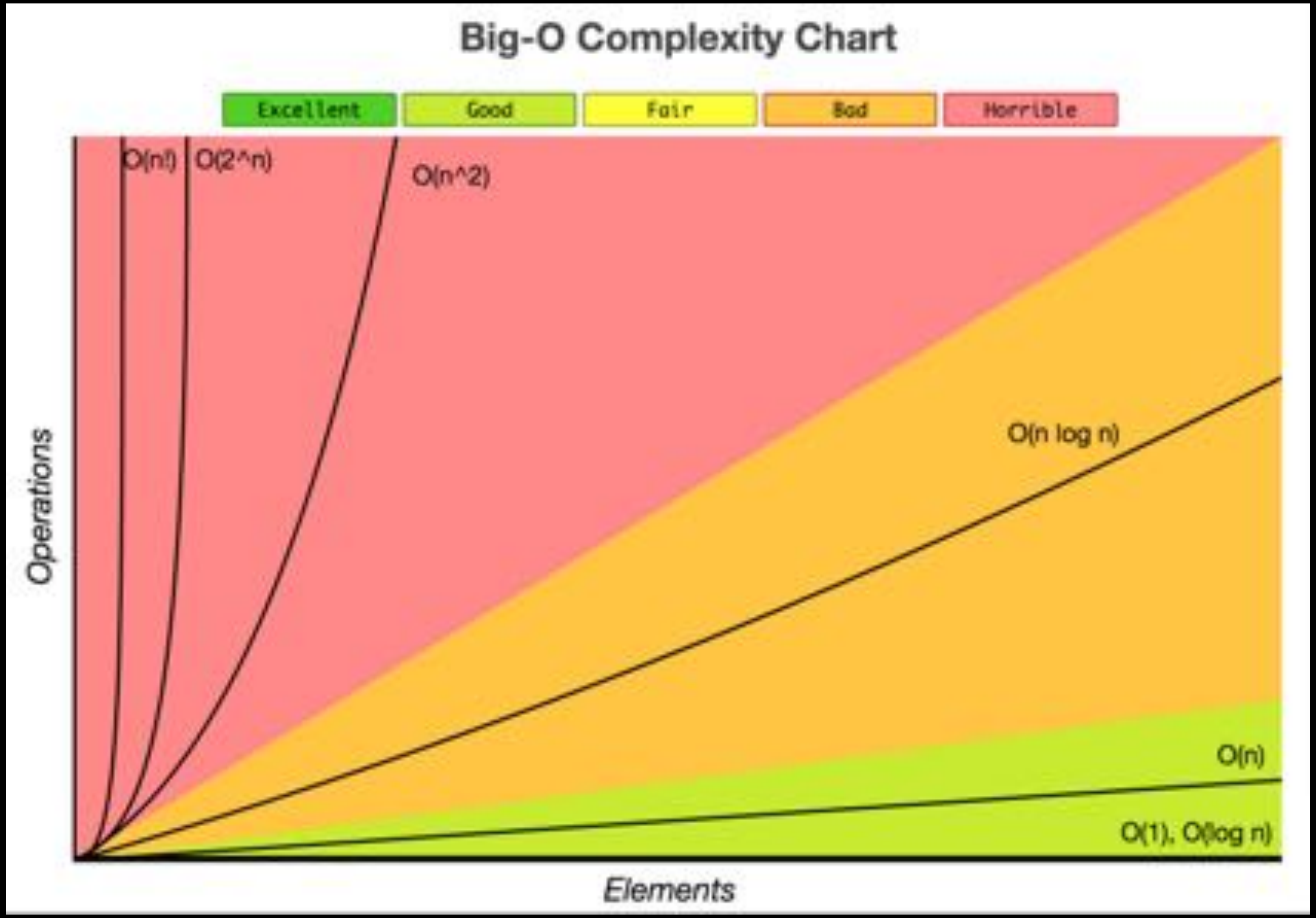
- Worst-case efficiency
- Big-O Notation

Concepts to remember

```
import numpy
a = 3
arr = np.arange(10)
for val in arr:
    val *= 47
    if val < a:
        val ** 2
```

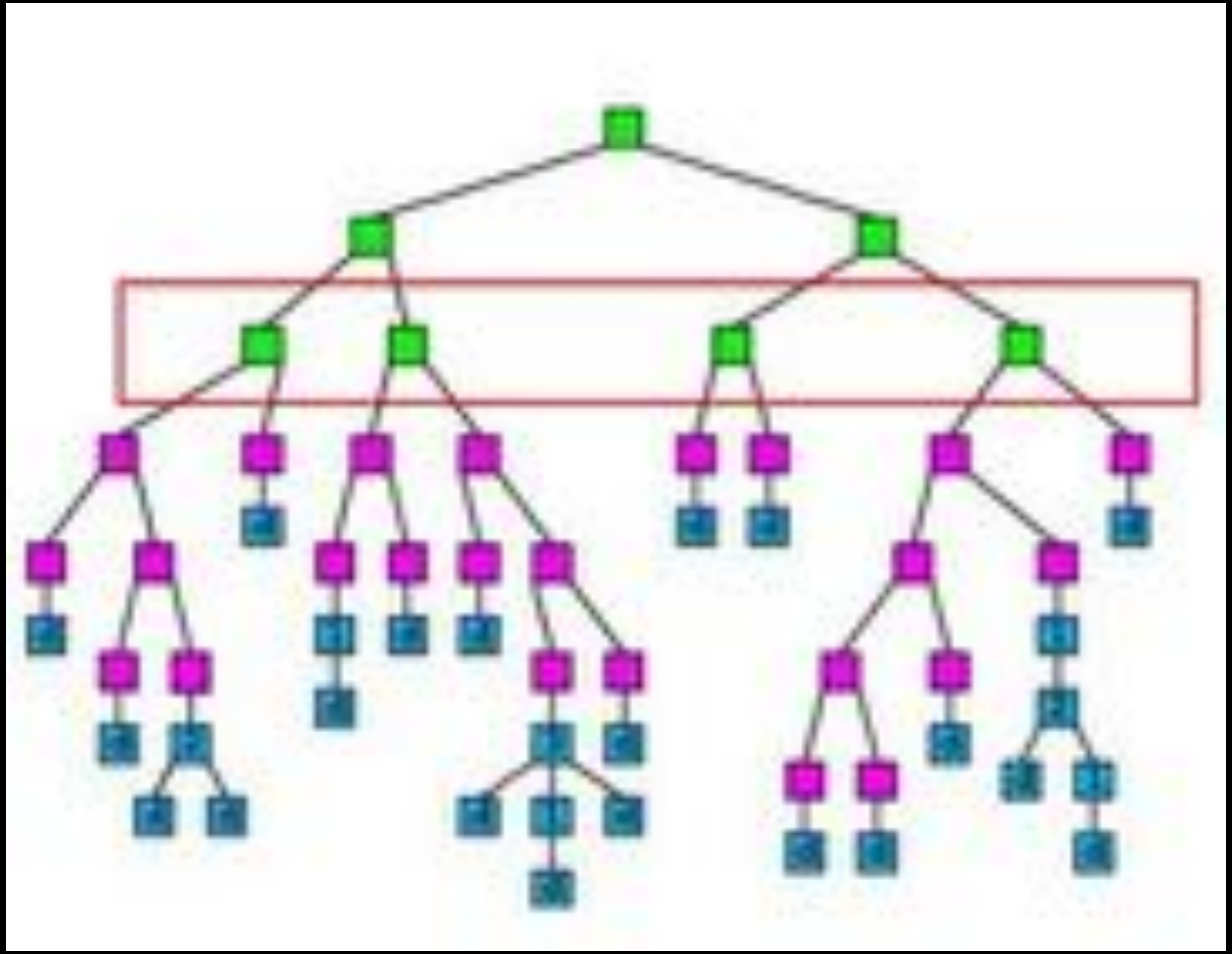
Algorithm Design

- Worst-case efficiency
- Big-O Notation



Algorithm Design

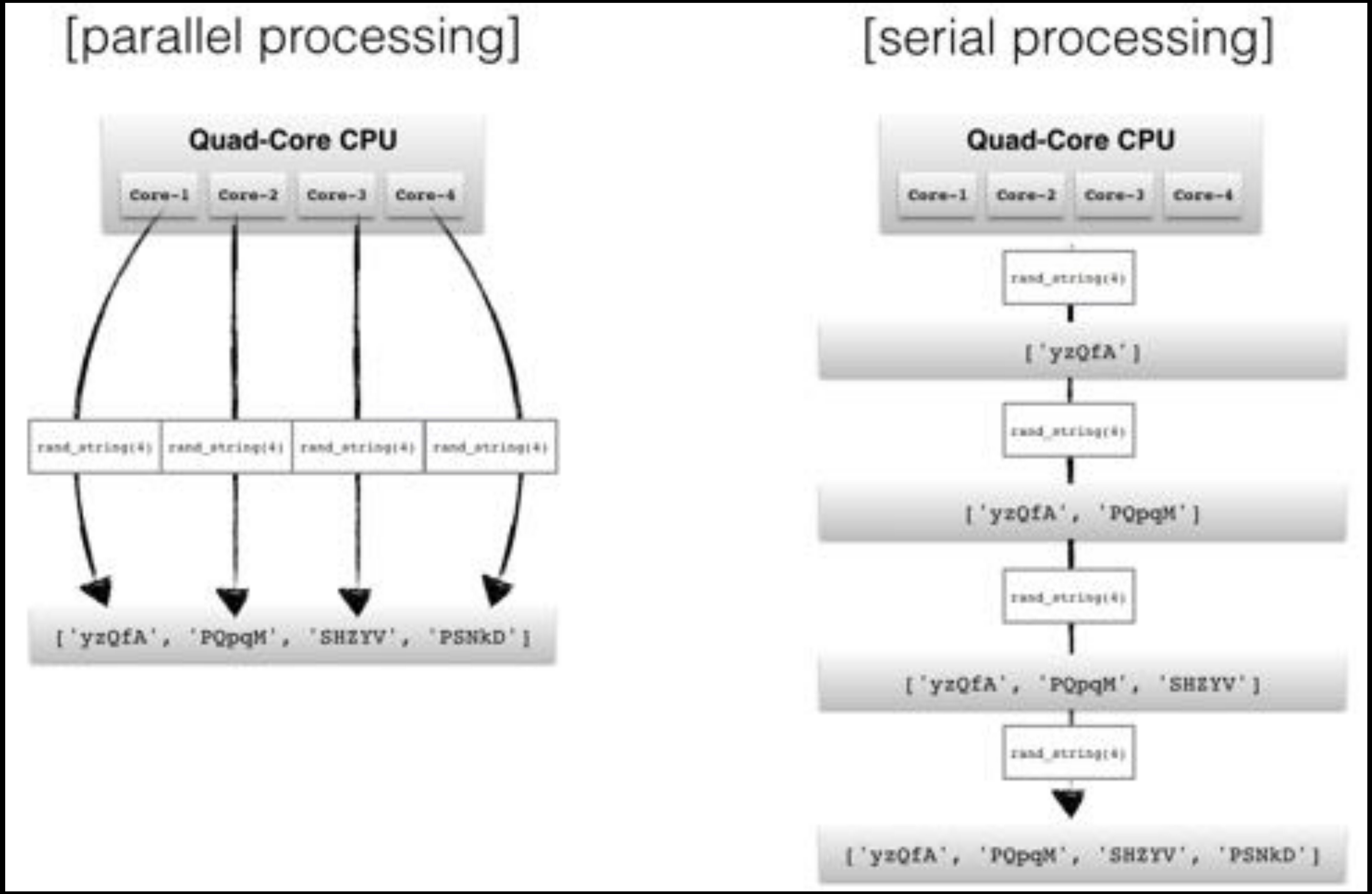
- Worst-case efficiency
- Big-O Notation



Random

Random

Parallel Computation



Parallel Computation

- multithreaded
- multiprocessing
- MPI and mpi4py

```
import multiprocessing
p = multiprocessing.Pool(2)
p.map(func, list_of_objects)
```