1. Maximum Speedup—Parallel Program Example

Given:

- Problem size: n = 10,000
- Sequential I/O time: 18000 + n = 28,000 µsec
- Parallelizable computation: n^2 / 100 = 10,000^2 / 100 = 1,000,000 µsec

- Step 1: Maximum speedup without communication overhead

  Maximum speedup formula: Speedup = Total sequential + parallel time / Sequential part + Parallel part / p

  - Sequential part (T_s) = 28,000 µsec
  - Parallel part (T_p) = 1,000,000 µsec
  - For ideal infinite processors, parallel part → 0
  - Maximum speedup = (T_s + T_p) / T_s = (28,000 + 1,000,000) / 28,000 ~=~ 36.7

- Step 2: Include parallel communication overhead

  - Number of communication points: ⌈log n⌉ = ⌈log 10,000⌉ ~=~ 14

  - Communication time at each point: n⌈log p⌉ + n/10 µsec

  - Assume very large p, ⌈log p⌉ grows slowly; just include n/10 as main term

  - Total overhead ~=~ 14 × (10,000 / 10) = 14 × 1,000 = 14,000 µsec

  - New sequential equivalent = 28,000 + 14,000 = 42,000 µsec

Maximum speedup with overhead: Speedup = (28,000 + 1,000,000) / 42,000 ~=~ 24.8

2. Parallel Code Segment Outcomes

All possible final values (x, y):

x	y
3	5
4	2
4	3
4	5
4	6
6	8

3. Maximum Achievable Speedup for 8% Sequential Computation

- Fraction sequential, f = 0.08
- Fraction parallel, 1 – f = 0.92

(a) Amdahl's Law (for very large $p \to \infty$):

Maximum speedup = $1 / f = 1 / 0.08 = 12.5$

(b) Gustafson-Barsis Law (scales with problem size):

Speedup = $f + (1 - f) \times p$ = For very large p, speedup scales almost linearly with p

- Maximum speedup grows, essentially unbounded in theory

(c) When to prefer Amdahl vs Gustafson-Barsis:

- Use Amdahl: When problem size is fixed, and you want to know the maximum speedup possible by parallelization.
- Use Gustafson-Barsis: When problem size can increase with more processors, and you want to predict realistic scaling with larger workloads.