As before, \( \text{WORK} = 26 \) and \( \text{CPL} = 11 \) for this graph

\( T_2 = 15 \), for the 2-processor schedule on the right

We can also see that

\[
\max(\text{CPL}, \text{WORK}/2) \leq T_2 < \text{CPL} + \text{WORK}/2
\]
Parallel Speedup

- Define Speedup(P) = $T_1 / T_P$
  - Factor by which the use of P processors speeds up execution time relative to 1 processor, for a fixed input size
  - For ideal executions without overhead, 1 <= Speedup(P) <= P
  - Linear speedup
    - When Speedup(P) = k*P, for some constant k, 0 < k < 1

- Ideal Parallelism

  = WORK / CPL

  = Parallel Speedup on an unbounded number of processors
Reduction Tree Schema for computing Array Sum in parallel

Assume input array size = $S$, and each add takes 1 unit of time:

- $\text{WORK}(G) = S - 1$
- $\text{CPL}(G) = \log_2(S)$
- Use upper bound to estimate $T_p = \frac{\text{WORK}(G)}{P} + \text{CPL}(G)$
  
  $= \frac{(S-1)}{P} + \log_2(S)$

- Within a factor of 2 of any greedy schedule's execution time
How many processors should we use?

- Define Efficiency(P) = Speedup(P)/ P = T\(_1\)/(P * T_P)
  - Processor efficiency --- figure of merit that indicates how well a parallel program uses available processors
  - For ideal executions without overhead, 1/P <= Efficiency(P) <= 1

- Half-performance metric
  - S\(_{1/2}\) = input size that achieves Efficiency(P) = 0.5 for a given P
  - Figure of merit that indicates how large an input size is needed to obtain efficient parallelism
  - A larger value of S\(_{1/2}\) indicates that the problem is harder to parallelize efficiently

- How many processors to use?
  - Common goal: choose number of processors, P for a given input size, S, so that efficiency is at least 0.5
ArraySum: Speedup as function of array size, $S$, and number of processors, $P$

- Speedup($S$, $P$) = $T(S,1)/T(S,P) = S/(S/P + \log_2(S))$

- Asymptotically, Speedup($S$, $P$) $\rightarrow S/\log_2S$, as $P \rightarrow \infty$

Efficiency($P$) $\leq 0.5$, for $P \geq 258$  
$\Rightarrow$ wasteful to use more than 256 processors for $S=2048$

Efficiency($P$) $\leq 0.5$, for $P \geq 128$  
$\Rightarrow$ wasteful to use more than 128 processors for $S=1024$
Amdahl’s Law [1967]

• If \( q \leq 1 \) is the fraction of WORK in a parallel program that must be executed sequentially for a given input size \( S \), then the best speedup that can be obtained for that program is \( \text{Speedup}(S,P) \leq 1/q \).

• Observation follows directly from critical path length lower bound on parallel execution time
  
  \[ \text{CPL} \geq q \times T(S,1) \]
  
  \[ T(S,P) \geq q \times T(S,1) \]
  
  \[ \text{Speedup}(S,P) = T(S,1)/T(S,P) \leq 1/q \]

• This upper bound on speedup simplistically assumes that work in program can be divided into sequential and parallel portions
  
  — Sequential portion of WORK = \( q \)
  
  — Parallel portion of WORK = \( 1-q \)

• Computation graph is more general and takes dependences into account
Illustration of Amdahl’s Law: Best Case Speedup as function of Parallel Portion

Figure source: http://en.wikipedia.org/wiki/Amdahl's law