V-6
Multi threaded Algo & Distributed Algo

- serial algo → uniprocessors → parallel algo → multiprocessors
- seprately runnable processes → threads

- Multi threaded algo → parallel algo → parallel computers
  ↓ multiple instructions
     to execute concurrently

- static threading → processors → thread is executed managed explicitly.
  Dynamic → programmer specify parallelism.

- concurrency constructs → acheive dynamic multithreading.

  3 keywords ⟶ • parallel — loop to indicate → each iter" parallel
                • spawn — create new subprocess & keep executing current process
                • sync — forces to wait until all active
                  parallel threads created by
                  instance of program finish.

performance measure → 1) work - total time to execute entire computation
                         on one processor.

                      2) span - longest time to execute strands
                         along any path in directed
                         acylic graph.

⊅ Dynamic Multithreading —

programmer to specify parallelism for appl^n without
worrying about

                                        ↗ • subroutine - spawn
                                          • called → parallel
— scheduler takes care of loadbalancing
— Dynamic threading → features ⟶ Nested Parallelism
                                  ↓
                          parallel loops ← simple loops
                                           executed parallel

Modes of Communication in Parallel Computing.

Shared Memory   Distributed Memory

all processors access common   all process run on
memory   their local processor

- serial algo → parallel → by adding keywords parallel, spawn & sync.
  Multi Threaded compu" → DAG → computation graph.

✳ Race cond^n ──→ multithreaded algo → deterministic → only if
                                                        same thing
                                                        on same input
                                                        & no matter how
                                                        instructions are
                                                        scheduled.

  ──→ multithreaded algo → need to be deterministic → but are NON DETEMTI
                          ↳ as they contain → determinacy rate

- determinancy rate → when two logically parallel instruction access
                      same memory location & at least
                      one of instructions perform a write

- Analysis of Multithreaded algo → 1) Analysis of Work
                                    2) Analysis of spawn.

Multi thread Merge Sort → merging two sorted sequence in
                          combined step.

Work $PMS_1(n) = \Theta(n \log n)$ $\dfrac{PMS_1(n)}{PMS_\infty(n)} = O\left[n\middle/\log^2 n\right]$
Span $PMS_\infty(n) = \Theta(\log^3 n)$

Distributed Algo-

* * supposed to work in distributed network or on mutipleprocess
  * created by interconnected processors.
  * subtype of parallel algo. typically executed concurrently

  * work on multiprocessor
  * don't share memory in multithreading

Appln ⟶
  * Distributed Computing
  * Telecommunication
  * Scientific computing
  * Distributed Info. processing

* ★ D BFS (Distribute Breadth First Search).

  * minimizes the max. commun^n time from process at distinguished node to all other proces in network.
  * output structure of BFS directed tree of network graph rooted at i0.

* # Distributed Minimun Spaning Tree.

  * min. spanning tree. of weighted connected graph G is a
    * spanning tree with min or smallest weight
    * algo works in phases.
    * each component tree → has leader.
    * each phase → spaning forest → component → Min Weight
      o Outgoing Edge (MWOE).
      o all are merged for next component

Message Complexity → $O(n \log n + |E|)$
Time Complexity → $O(n \log n)$.

⚡ String Matching

↳ used in text preprocessing
→ done in compilation of program.
→ finding one or more of occurrence in string → pattern.
→ string matching → pattern algo ✓

Naive Algo.

- Uses Brute Force Approach. → straight forward approach
- Just do it Approach.
- checking all position in texts b/w 0 to m-m → whether occurence
  state true or no.
- each attempt → shifts pattern by exactly one position to right
- match found → returns.
  - If not → matching process continues.
  - If no match → all text for given pattern → n compares

Rabin Karp Algorithm —

- Uses sliding window to match pattern.
- hashing technique.
- only if hash value of pattern matches with hash value of current
  substring

needs two things →   1) Hash value of pattern
                       2) Hash value of all substrings of
                          length m.