- Algorithms Research Topics
  - 6.851 Data Structures
  - 6.889 Graph Algorithms
  - SP.268 Recreational Algorithms
  - Parallel Processor Architectures
    - PC
      - Intel
        - 8086
        - 80486
          - 25 MHz
          - i486
        - Pentium
          - 66 MHz
        - Pentium 4
          - 1.5 GHz
          - Deep pipeline
            - 30 Stage
          - Bugs
            - Division in corner cases was failing
            - F00F bug
              - Malicious program crashes system
        - Pentium D
          - 3.2 GHz
          - Clock frequency stopped here
        - Quadcore Xeon
          - 3 GHz
          - 4 Processors
      - IBM
        - IBM PC
          - 5 MHz
    - QuadCore Processor
      - 4 Processors
      - SRAM
        - On Chip
        - Fast communication between 4 processors and SRAM
        - Mega Bytes access
        - Problem
          - Bottleneck
            - Read Ports ~ 4
              - Supports upto 4, 8 or 16 cores
          - Solution
            - Each Core has it's own SRAM
              - tiles architecture
                - Interconnection between processors
                - Routing problem
                  - Shortest path
                    - Weighted
                - Thread sharing SRAM with other Thread
                  - Thread 1 makes request and gets data back
                    - Round-trip time
                      - 10s of cycles
                  - Research: Execution Migration
                    - Migrate program "context" to another core
                      - context
                        - Program Counter
                        - Register File
                        - ...
                      - Advantages: One-way-trip
                      - Disadvantages: Kbits
                        - program > data
                        - Solution: Optimization algorithm to find when data migration is preferred over context migratio

                          Statement: Assume we know or can predic the access pattern of a program (pre-fetch engines, branch-predictors) and m1, m2, ... mN are memory addresses (accesses) and p(m1), p(m2), ... p(mN) are processor caches - p1, p2, p2, ...p3 (accesses of the program). Let cost_mig(S,d) = distance(S,d) + L (contant proportional to context size- assumed fixed) , cost_access(S, d) = 2 * distance(S, d) - no additional overhead on data access.

                          Problem: Decide when to migrate to minimize total memory access cost.

                          Solution: Use dynamic programming
                          Program at Pi initially, number of processors = Q
                          Subproblems?

                          DP(k, pi) = cost of the optimal solution for the prefix m1, .., mk of memory accesses and ends at pi

                          Subproblems: NQ

                          DP(k + 1, pj) = { DP(k, pj) + cost_access(pj, p(m_k+1)) if pj != p(m_k + 1), MIN_i=1 to Q (DP(k, pi) + cost_migration(pi, pj)) if pi = p(m_k + 1)}

                          Cost: O(NQ^2)
          - From DRAM get Cache Line (block of memory)
            - Word Ram
              - Search
                - O(log_B n)
              - Sorting
                - O(n/B log_c n/B)
                - c = number of blocks that fit in cache
              - Insert, Delete, Predecessor, Successor
                - n = integers in {0, u-1}
                - O(lglg u)
                - Van Em De Boas
                - O(lg n / lg lg u)
                - Fusion trees
                - Advaced
                - O(sqrt(log n / lg lg n))
      - DRAM
        - Multiple modules
        - GB
    - Parallelel Programs
  - Computational Geometry
    - Folding
      - 6.849
      - Geometric Folding Algorithms
        - Paper into 3D shape
          - Put creases in paper
            - If 3D shape is given then what is the crease pattern
              - Solvable
                - Design is easier
                - In P
                - Optimal can be NP-Complete
              - Organizer
                - Software
              - Fold and cut stright lines we get different shapes
          - Fold it into shape
            - What is the foldability
              - Folding a crease pattern is NP-complete
    - Self-Assembly
      - 6.850