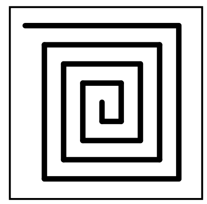
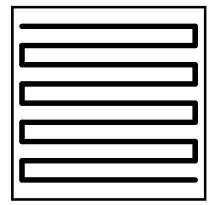
Image Processing
(Divide-and-Conquer)

<Students should read M&S, section 2.5.3, on Transitive Closure *before* starting this discussion on D&C: Image Processing since it is heavily used.>

1. Digitized Picture **Component Labeling**
   a) Explain input and problem for $n \times n$ image
   b) Give sequential $O(n^2)$ time algorithm.
   c) Give parallel propagation algorithm on an MCC of size n^2, which requires $O(n^2)$ time.
      i) Show samples of bad images:
         a) Spirals
         b) Snakes





   d) Divide-and-Conquer algorithms
      i) Recursively solve the problem in each quadrant. (This will reduce $O(n^2)$ black pixels to $O(n)$ pieces of information.)
      ii) Do one of the following:
         a) Compress this **edge** data and solve (*discuss* the ramifications of this compression and the flexibility it gives the algorithm designer), **OR**
         b) Create cross-product (adjacency matrix) and solve (using the transitive closure algorithm which we have **not** discussed)
      iii) Concurrent read to obtain the correct final results.
      iv) Running times:
         a) Compression:
            $T(n^2) = T(n^2 / 4) + Comp(n, n^2) + Edge(n^2) + CR(n^2)$
         b) Cross Product:
            $T(n^2) = T(n^2 / 4) + Create(n^2) + Adj(n^2) + CR(n^2)$

2. Digitized Picture **Internal Distances**

a) *Nearest neighbor for every labeled set* (**not** a D&C algorithm):

i) **<u>Defn.:</u>** Assume every PE has a (not necessarily unique) label. For each PE, find the nearest neighboring PE with a different label.

ii) Perform a **row rotation** so that every PE finds its nearest distinctly labeled PE in its row.

iii) Perform a **column rotation** where the information passed consists of 2 pieces per PE (initial value and value just obtained in previous step).

iv) **Running time:** $T(n^2) = \Theta(n)$

v) Extension: Solve the problem so that for every labeled *set* of processors, the label of a nearest distinctly labeled set is determined.

b) Minimum Internal Distance to Marked Pixel in Connected Component (D&C algorithm)

i) **<u>Defn.:</u>** Given a digitized picture $A_{n \times n} = \{a_{i,j}\}$ distributed one pixel per PE so that PE $P_{i,j}$ stores pixel $a_{i,j}$, and given a (single) special marked pixel x, determine for every pixel *a* the minimum **<u>internal</u>** distance from *a* to *x*, denoted $d(a, x)$. {Note: for every pixel *a* in a distinct connected component from *x*, $d(a, x) = \infty$.

ii) *Propagation* algorithm requires $\Theta(n^2)$ time in the worst case.

iii) Generalized *Transitive Closure* requires $\Theta(n^4)$ space in the worst case.

iv) Asymptotically optimal worst-case algorithm consists of Bottom-Up D&C / Data Reduction / Generalized T.C.

(*See Over*)

Phase 1: Objective is to obtain the distance from every border element to x. This is done by bottom-up merging, which uses the generalized T.C. and includes an extra row/column for the marked PE labeled x.

<*insert M&S Figure 3.5* - Win 3.1/Word 6.0 doesn't seem to want to let me include a .ps file directly and be able to process it properly so it can be previewed with a variety of PS previewers>

Possible border elements of a submesh of size k^2.

<*insert M&S Figure 3.6*- Win 3.1/Word 6.0 doesn't seem to want to let me include a .ps file directly and be able to process it properly so it can be previewed with a variety of PS previewers >

Rows labeled R and columns labeled C represent those rows and columns of the matrices associated with the special marked PE x.

Note that at step i, the algorithm works on $2^i \times 2^i$ images and requires $\Theta(2^i)$ time.

Therefore, the total time for phase 1 is $\Theta(n)$.

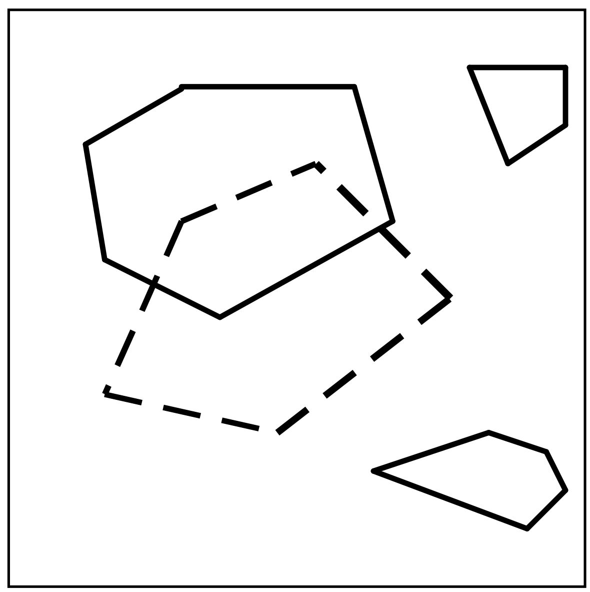
Phase 2: Recursively find distance to all other pixels. In some sense, this is the reverse of Phase 1, where at each stage in this top-down phase, the results that are known for the larger square at the previous stage are used.

At any stage *i*, phase 2 works on $2^{\log_2 n-1} \times 2^{\log_2 n-1}$ images, which requires $\Theta(2^{\log_2 n-1})$ time.

Therefore, the running time of phase 2 is $\Theta(n)$.

3. Digitized Picture **Convex Hull**

a) **Defn.:** Given a MCC of size n^2, in which every processor has a (not necessarily unique) label, mark the set of extreme points for every labeled set of processors. We assume that PE $P_{i,j}$ is associated with lattice point (i, j). (*Note*: sets of PEs/lattice points need not necessarily be connected.)



b) **Algorithm**:

   i) Row rotation to identify leftmost and rightmost PE for every label in every row.

   ii) Sort all such points by label (note that there are $\leq 2n$ points per label).

   iii) Rotate all points within its label. During this operation, suppose a PE is responsible for lattice point x. As a new point Y arrives, the PE

      a) stores Y if no other point is stored

      b) stores Y if only 1 other point U has been stored. If X, Y, and U are on the same line, then the processor eliminates the interior point of the 3.

      c) If U and Y are stored when Y arrives:

         (1) If X is in ΔUVY, then X is not an extreme point.

         (2) If Y is on a line determine by X and one other, then Y is discarded.

         (3) Eliminate whichever of U,V,Y is inside the angle formed by X and the other 2.

      d) If after the rotation, X has not been eliminated, the X is an extreme point.

   iv) Concurrent Read to send points back and mark extreme points.

c) Discuss Running Time: $\Theta(n)$