MA/CSSE 473 Day 23

Transform and Conquer

MA/CSSE 473 Day 23

- Scores on HW 7 were very high (except for 5 students who didn't do it at all). More perfect scores on this one than any previous assignment. Good job!
- HW 9 is due tomorrow
  - with a "late day" that extends until Friday at noon.
- HW 10 due next Tuesday, Oct 19
- ConvexHull implementation due Thursday, Oct 21
- HW 11 due Friday, Oct 22. It is smaller than most assignments.
- HW 12 due Tuesday, Oct 26.
- Take-home exam available Oct 29 (Friday) at 9:55 AM, due Nov 1 (Monday) at 8 AM.
- **Student Questions**
- Transform and conquer
  - Many of the examples should be review.

Transform and Conquer Algorithms

- Transform a problem to a simpler instance of the same problem – **instance simplification**
- Transformation to a different representation of the same instance – **representation change**
- Transformation to an instance of a different problem that we know how to solve – **problem reduction**

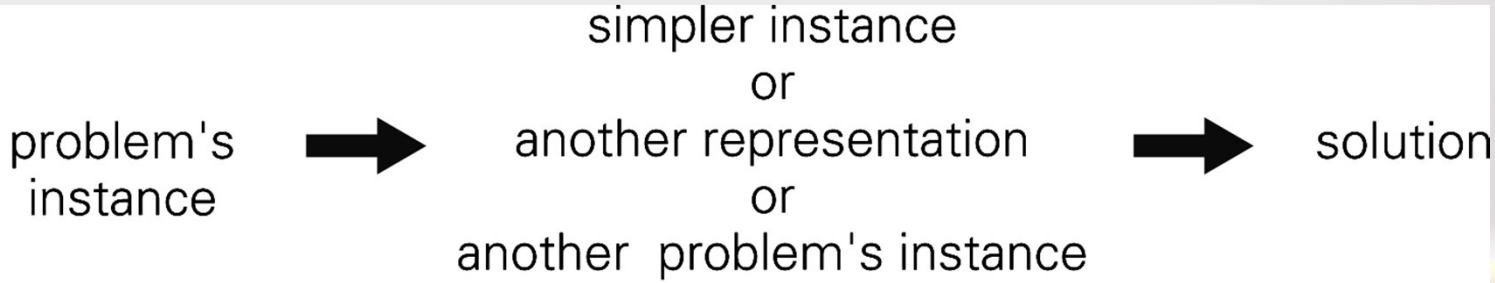


FIGURE 6.1 Transform-and-conquer strategy

Q1

Presorting an Array

- The following problems are simplified by pre-sorting the array:
  - Search (can do Binary or Interpolation search)
  - Determine whether the array contains duplicates
  - Find the **mode** of the elements of the array
    - The most frequently-occurring element
  - A related problem: Anagrams
    - In a large collection of words, find words that are anagrams of each other
    - How can pre-sorting help?
    - Sort the letters of each word
  - Interval union problem from early part of PLC

Gaussian Elimination (review of MA221)

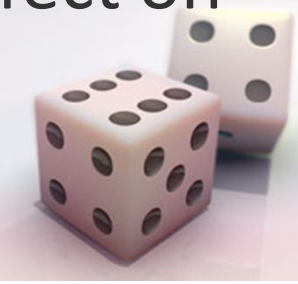
- Solve a system of n linear equations in n unknowns
  - Represent the system by an augmented coefficient matrix
  - Transform the matrix to triangular matrix by a combination of the following solution-preserving elementary operations:
    - exchange two rows
    - multiply a row by a nonzero constant
    - replace a row by that row plus or minus a constant multiple of a different row
  - Look at the algorithm and analysis on pp 207-208; if you can't understand them, ask at some point.
    - Hopefully you saw this in your D.E. class.
  - $\Theta(n^3)$

Other Applications of G.E.

- Matrix inverse
  - Augment a square matrix by the identity matrix
  - Perform elementary operations until the original matrix is the identity.
  - The "augmented part" will be the inverse
  - More details and an example at http://en.wikipedia.org/wiki/Gauss-Jordan_elimination

Other Applications of G.E.

- Determinant calculation
  - Calculation of the determinant of a triangular matrix is easy
- What effect does each of the elementary operations have on the determinant?
  - exchange two rows
  - multiply a row by a nonzero constant
  - replace a row by that row plus or minus a constant multiple of a different row
- Do these operations until you get a triangular matrix
- Keep track of the operations' cumulative effect on the determinant



LU Decomposition

- This can speed up all three applications of Gaussian Elimination
- Write the matrix A as a product of a Lower Triangular matrix L and an upper Triangular matrix U.
- Example:

$$[A]=\begin{bmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{bmatrix}$$

$$[L]=\begin{bmatrix} 1 & 0 & 0 \\ 2.56 & 1 & 0 \\ 5.76 & 3.5 & 1 \end{bmatrix} \quad [U]=\begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix}$$

Q6

Balanced Search Trees

- Why do we introduce the idea of Binary Search Trees?
  - i.e., why are array lists or linked lists not good enough?
- What is the problem if we don't make sure Binary Search Trees are kept balanced after insert or delete?
- AVL Trees (height balanced)
- Review of why we know that the max height of an AVL tree is $\Theta(\log N)$.
- Review the balancing algorithms after insertion, deletion.

- Named for authors of original paper, **A**delson-**V**elskii and **L**andis (1962).
- An AVL tree is a height-balanced Binary Search Tree.
- A BST T is **height balanced** if T is empty, or if
  - $| \text{height}(T_L) - \text{height}(T_R) | \leq 1$, and
  - T_L and T_R are both height-balanced.
- Show: Maximum height of an AVL tree with N nodes is $\Theta(\log N)$.
- How do we maintain balance after insertion?
- **Exercise:** Given a pointer to the root of an AVL tree with N nodes, find the height of the tree in log N time.
- Details on balance codes and various rotations are in the CSSE 230 slides that are linked from the schedule page.