Design and Analysis of Algorithms

Lower-Bound Arguments

Reetinder Sidhu

Department of Computer Science and Engineering

Design and Analysis of Algorithms

Lower-Bound Arguments

Reetinder Sidhu

Department of Computer Science and Engineering

- Dynamic Programming
  - Computing a Binomial Coefficient
  - The Knapsack Problem
  - Memory Functions
  - Warshall's and Floyd's Algorithms
  - Optimal Binary Search Trees
- Limitations of Algorithmic Power
  - **Lower-Bound Arguments**
  - Decision Trees
  - P, NP, and NP-Complete, NP-Hard Problems
- Coping with the Limitations
  - Backtracking
  - Branch-and-Bound

Concepts covered

- Lower-Bound Arguments
  - Trivial lower bounds
  - Adversary arguments
  - Problem reduction

Limitations of Algorithmic Power

- There are no algorithms to solve some problems
  - Ex: halting problem
- Other problems can be solved algorithmically, but not in polynomial time
  - Ex: traveling salesman problem
- For polynomial time algorithms also, there are lower bounds

Definition

Lower Bound

An estimate on a minimum amount of work needed to solve a given problem (estimate can be less than the minimum amount of work but not greater)

- Examples:
  - number of comparisons needed to find the largest element in a set of n numbers
  - number of comparisons needed to sort an array of size n
  - number of comparisons necessary for searching in a sorted array
  - number of multiplications needed to multiply two $n \times n$ matrices

Lower-Bound Arguments
Bound Tightness

- A lower bound can be:
  - an exact count
  - an efficiency class (Ω)

Tight Lower Bound

There exists an algorithm with the same efficiency as the lower bound

Problem	Lower Bound	Tightness
Sorting	$\Omega(n \log n)$	yes
Searching a sorted array	$\Omega(\log n)$	yes
Element uniqueness	$\Omega(n \log n)$	yes
Integer multiplication ($n \times n$)	$\Omega(n)$	unknown
Matrix multiplication ($n \times n$)	$\Omega(n^2)$	unknown

Methods for Establishing Lower Bounds

- Trivial lower bounds
- Information-theoretic arguments (decision trees)
- Adversary arguments
- Problem reduction

Trivial Lower Bounds

Trivial Lower Bounds

Based on counting the number of items that must be processed in input and generated as output

- Examples
  - finding max element
  - polynomial evaluation
  - sorting
  - element uniqueness
  - Hamiltonian circuit existence
- Conclusions
  - may and may not be useful
  - be careful in deciding how many elements must be processed

Adversary Arguments

Adversary Argument

A method of proving a lower bound by playing role of adversary that makes algorithm work the hardest by adjusting input

- Example 1: "Guessing" a number between 1 and n with yes/no questions
  - Adversary: Puts the number in a larger of the two subsets generated by last question
- Example 2: Merging two sorted lists of size n $a_1 < a_2 < \ldots < a_n$ and $b_1 < b_2 < \ldots < b_n$
  - Adversary: $a_i < b_j$ iff $i < j$
    Output $b_1 < a_1 < b_2 < a_2 < \ldots < b_n < a_n$ requires $2n - 1$ comparisons of adjacent elements

Problem Reduction

- Basic idea: If problem P is at least as hard as problem Q, then a lower bound for Q is also a lower bound for P
- Hence, find problem Q with a known lower bound that can be reduced to problem P in question
- Example: P is finding MST for n points in Cartesian plane Q is element uniqueness problem (known to be in $\Omega(n \log n)$)

Lower-Bound Arguments

Think About It

- Prove that the classic recursive algorithm for the Tower of Hanoi puzzle makes the minimum number of disk moves
- Find a trivial lower-bound class and indicate if the bound is tight:
  - finding the largest element in an array
  - generating all the subsets of an n-element set
  - determining whether *n* given real numbers are all distinct