Agenda

RANDOMIZED ALGORITHMS – INTRODUCTION

- MODEL
- LAS VEGAS AND MONTE CARLO

CS&IS, BITS, Pilani @Pilani

Sundar B.

Randomized Algorithms

- A randomized algorithm is an algorithm that
  - is allowed access to a source of independent unbiased random bits, and
  - is allowed to use these random bits to influence its computation.
- The performance of a randomized algorithm can be proved by relying solely on the random choices
  - i.e. without any assumptions about inputs
- Contrast this with probabilistic analysis where
  - one assumes a distribution on the inputs

Randomized Algorithms

- Cost of randomization:
  - Cost model:
    - Sampling of a random element from a set S is done by
      - choosing O(log|S|) random bits and then
      - using these bits to index an element in S
  - Thus cost of choosing a random number in a set of N numbers is O(logN)
    - assuming a single random bit can be obtained in unit time from an unbiased source or random bits (e.g. a single coin flip)

Randomized Algorithms

- Advantages over deterministic algorithms:
  - They provide better *expected* performance
    - Worst case performance may not be better!
  - Their performance is not dependent on the input
  - They are often easier to design and implement
    - assuming library support for random number generation

Las Vegas and Monte Carlo

- A randomized algorithm that always gives the correct solution is said to be a ***Las Vegas*** algorithm
  - E.g. Randomized Quicksort
    - *where the only variation from one run to another is the time complexity*
- A randomized algorithm that may produce an incorrect solution is said to be a ***Monte Carlo*** algorithm.
  - Q: Have you seen an example?
    - **Bloom Filter** vs. Hash Table

Hash Tables vs. Bloom Filters

- What is the difference between bucket / bin sorting and hashing?
  - Key Ranges?
  - Locations?
- Hashtables provide a ***Las Vegas*** algorithm for storing / retrieving a record in a dictionary data structure.
  - What about **Bloom Filters**?
    - When a Bloom Filter returns:
      - a ***yes*** answer to a ***find*** query it may be erring with a small probability
      - a ***no*** answer to a ***find*** query it is always correct

Monte Carlo algorithms

- Such techniques are referred to as *1-way error Monte Carlo algorithms* and
  - usually the error probability is bounded by a small fraction.
- REVIEW EXERCISE:
  - Calculate the false positive rate for a Bloom Filter given:
    - M the size of the table,
    - b the number of bits per record (which is 1 by default),
    - d the number of hash functions used, and
    - N the number of entries.