Randomized Algorithms

Jonathan Fragoso

December 27, 2014

Contents

Chapter 1

Introduction

1.1 Randomized algorithms

Algorithms that make random choices during their execution. In practice, a randomized program would use values generated by a random number generator to device the next step at several branches of its execution.
In some applications, randomized algorithms are significantly more efficient that the best known deterministic solutions. In addition to this fact, in most cases the randomized algorithms are also simpler and easier to program.

Because of the randomized approach, the efficiency is guaranteed only with some probability, but if the probability of error is sufficiently small, then the improvement in speed or memory requirements may well be worthwhile.

1.2 Probabilistic analysis of algorithms

Complexity theory tries to classify computation problems according to their computational complexity. Probabilistic analysis gives a theorical explanation for this phenomenon. If we think of the input as being randomly selected according to some probability distribution on the collection of all possible inputs, we are very likely to obtain a problem instance that is easy to solve, and instances that are hard to solve appear with relatively small probability. Thus, we can conclude saying that probabilistic analysis of algorithms is the method of studying how algorithms perform when the input is taken from a well-defined probabilistic space.

Chapter 2

Events and probabilities

2.1 Axioms of probability

A probability space has three components:

1. Sample space (Ω) which is the set of all possible outcomes of the random process modeled by the probability space. An element of Ω is called a simple or elementary event.

2. Family of sets ($\mathcal{F}$) representing the allowable events, where each set in $\mathcal{F}$ is a subset of the sample space Ω

3. probability function (Pr) : $\mathcal{F} \to \mathbb{R}$ that satisfies the following conditions:

   (a) for any event E, $0 \leq \Pr(E) \leq 1$;

   (b) $\Pr(\Omega) = 1$; and

   (c) for any finite or countably infinite sequence of pairwise mutually disjoint events $E_1, E_2, E_3, ...,$

$$Pr\left(\bigcup_{i\geq 1} E_i\right) = \sum_{i\geq 1} Pr(E_i)$$

Suppose we roll two dice. If E_1 is the event that the first die is a 1 and E_2 is the event that the second die is a 1, then $E_1 \cap E_2$ denotes the event that both dice are 1 while $E_1 \cup E_2$ denotes the event that at least one of the two dices lands on 1. Similarly, we write $E_1 - E_2$ for the occurence of an event that is in $E_!$ but no in E_2.

Two events E and F are independent if and only if

$$Pr(E \cap F) = Pr(E) \cdot Pr(F).$$

The conditional probability that event E occurs given that event F occurs is

$$Pr(E|F) = \frac{Pr(E \cap F)}{Pr(F)}$$

The conditional probability is well-defined only if Pr(F) > 0.

2.2 Probability theory basics

2.3 Distributed algorithm to check equality of strings

2.4 Algorithm to verify polynomial equivalence

2.5 Algorithm to verify matrix products

Chapter 3

Random variables and expectations

3.1 Rnd. vars

3.2 Expectations

3.3 The Bernoulli rnd var

3.4 The Binomail rnd var

3.5 The Geometric rnd var

3.6 The coupons collector problem (I)

3.7 1/2-approximation algorithm for MaxCut

3.8 7/8-approximation algorithm for Max3Sat

3.9 Quicksort

Chapter 4

Moments and deviations

4.1 Markov's inequality

4.2 Variance and moments

4.3 Chebyshev's inequality

4.4 Coupons collector problem (II)

4.5 Randomized alg for the median

Chapter 5

Chernoff bounds

5.1 Chernoff bounds

5.2 Random geometric graphs

5.3 Concentration of Quicksort

Chapter 6

Balls and bins

6.1 Birthday paradox

6.2 Bucket sort

6.3 Poisson distribution

6.4 Poisson approximation

6.5 Hashing

Chapter 7

Markov chains

7.1 Definitions and basic properties

7.2 Stationary distributions

7.3 Random walks

7.4 Alg. for 2-sat

Chapter 8

Applications

8.1 Perfect dynamic hashing

8.2 Cuckoo hashing

8.3 Skip lists

8.4 Skip nets

8.5 Primality testing

8.6 Closest pair of points

8.7 Minimum enclsing disk

8.8 Pattern matching

8.9 Blum filters

8.10 Packet routing

8.11 Heuristics

8.12 Minimum cut

Bibliography

[1] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, New York, NY, USA, 2005.