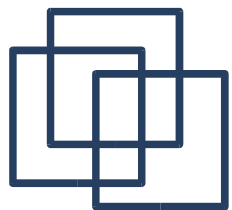


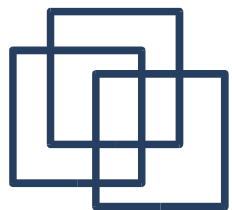
Algorithms and Architecture 1

Probabilistic Analysis and Randomized Algorithms



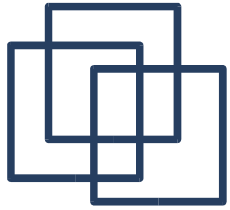
Probability Basics (Ap. C)

- Counting
 - Permutations, k-permutations
 - k-combination, binomial coefficients
- Probability
 - Axioms, discrete distribution
 - Conditional probability
 - Independent events
 - Bayes theorem $\Pr\{A \mid B\} = \Pr\{A\}\Pr\{B \mid A\} / \Pr\{B\}$



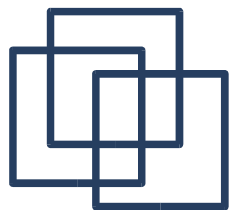
Probability Basics (Ap. C)

- Discrete random variables
 - function from sample space S to real numbers
 - probability density function $Pr\{X=x\} = \sum_{s \in S: X(s)=x} Pr\{s\}$
 - expected value $E[X] = \sum_x Pr\{X=x\}$
- Bernoulli trial
- Geometric distribution
- Binomial distribution



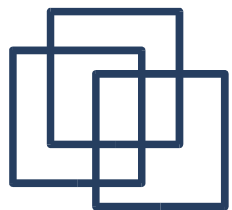
What is it about?

- Worst case analysis: worst cost or running time of an algorithm.
- Probabilistic analysis: compute the average case in terms of cost or running time.
- Randomized algorithms: algorithms that have a randomized decision.



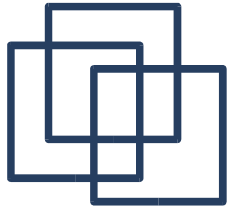
Hiring Problem Example

- The problem: hire new office assistant and fire the old (worse) one.
- Cost of interviewing and hiring $O(nc_i + mc_h)$
- Worst case $O(nc_h)$ (focus on hiring)
- Probabilistic analysis:
 - $E[X_i] = \Pr\{\text{candidate } i \text{ hired}\} = 1/i$
 - $E[X] = \ln(n) + O(1)$
 - expected cost is $O(c_h \ln(n))$



Randomized Algorithms

- Distinction between probabilistic analysis of an algorithm, knowing its *input distribution*, and analysis of randomized algorithms for *any input*.
- Randomized hire algorithm: permute list of candidates:
 - no particular input will show the worst-case
 - every run is different, although the final result is the same



Permuting Arrays

- A way to randomize inputs
- Permute-by-sorting
 - assign random priorities and sort
 - uniform random permutation
- Randomize-in-place
 - swap elements randomly
 - uniform random permutation

