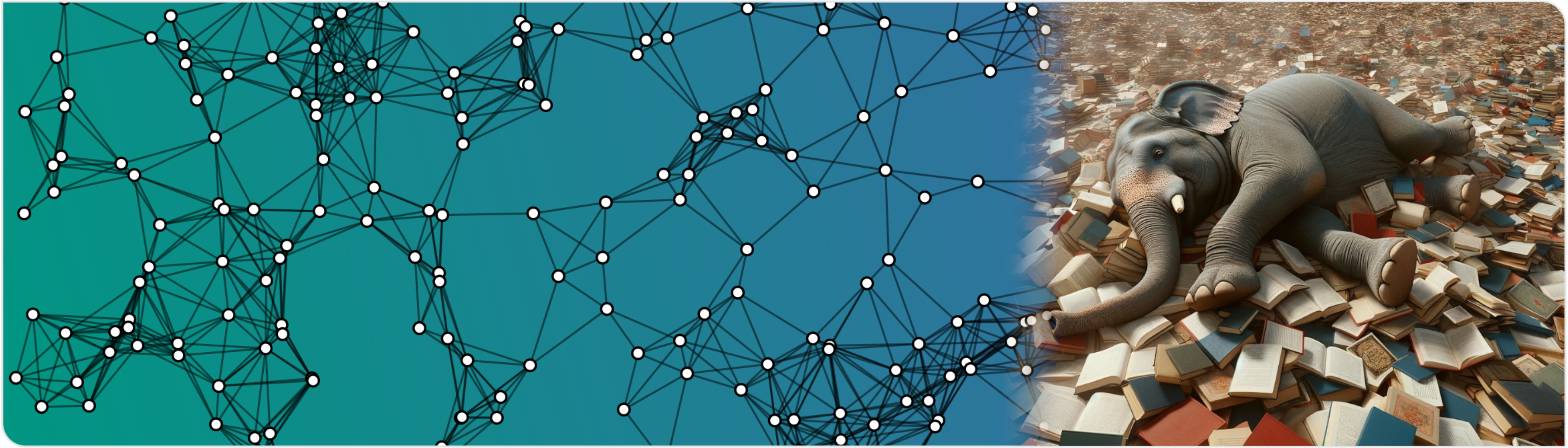
Karlsruhe Institute of Technology

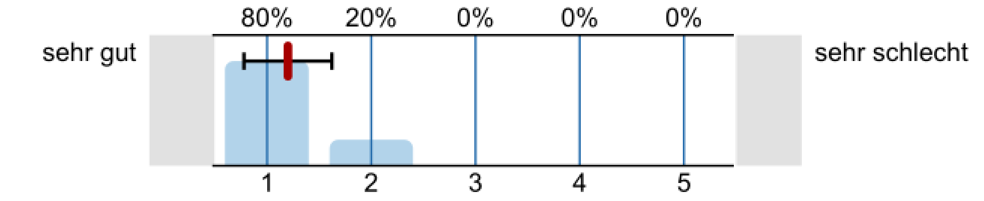
Probability & Computing

Conclusion

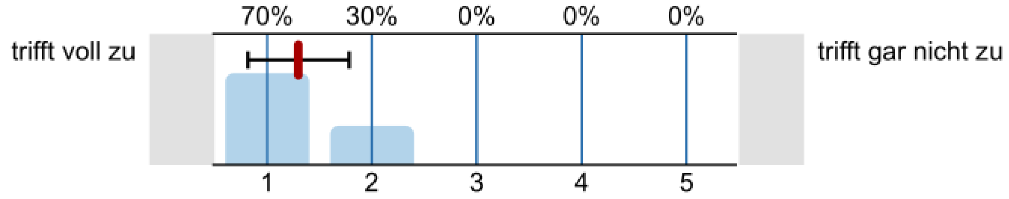


Eval

Bitte benoten Sie die Lehrveranstaltung insgesamt



In dieser Lehrveranstaltung lerne ich viel.



Things to keep

Die Folien sind extraklasse

Elephantenbilder

hinweisen auf aktuelle Forschung

spannende Themen

Things to improve

Ich finde, dass max tendentiell zu schnell die Folien bespricht.

Vor allem bei Max war die Zeit manchmal zu knapp für die Inhalte

sehr viele Mathematische Umformungen

Wall-of-Formeln

Manchmal fehlen den Folien ohne clicks viel Stoff ✓ ?

Übungsblätter jede Woche sind einfach anstrengend ✓ ?

Inverted Classroom Thoughts? (active sessions, videos, etc.)

Exercises Thoughts?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
  - What are Monte Carlo algorithms?
  - What kinds of biases can they have?
  - What is probability amplification?
  - How does probability amplification affect the running time and success probability of an algorithm?
  - How do we deal with the biases during probability amplification?
  - Can you derive the trade-off for different running times and success probabilities?
  - We have seen examples where probability amplification was not really useful. Why?
  - For the problem of minimum cuts we saw Karger’s algorithm
    - How does the algorithm work? How was this approach motivated?
    - How did the Karger-Stein approach improve over it? How was this adjustment motivated?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms

- What is a randomized approximation algorithm (for a counting problem)?
- We considered the counting problem $\#B$ for Boolean formulas. Did we generally succeed? Why not?
- Which special case did we consider then? Why did we not run into the same issue?
- We saw an algorithm that, given sets $S \subseteq D$ approximates the size $|S|$
  - Under which assumptions is it applicable?
  - How does the algorithm work?
  - How does the number of required samples depend on $|S|$ and $|D|$?
- To approximate $\#B$ for a DNF B we considered a more sophisticated approach
  - How does it work?
  - How does it avoid the problem of the naive approach?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms

- Definition of streaming algorithms
  - What is the task of a streaming algorithm (with respect to a value $F = F(a_1, ..., a_m)$)?
  - What is the specific challenge that streaming algoithms face?

- Streaming algorithms for $F_1 = m$
  - For what application may we need an estimate of F_1?
  - How much memory is needed when simply counting? Can a deterministic method do something better?
  - How does the LossyCounting algorithm work? Why is that not useful?
  - How does Morris' algorithm work?
    - Can you prove that it is an unbiased estimator?*
    - Can you prove that the required space is doubly-logarithmic in m?
    - What is its weakness and how did we fix it?

- Streaming algorithms for $F_0 = \{a_1, ..., a_m\}$
  - For what application may we need an estimate of F_0?
  - How much memory does the naive deterministic algorithm need? What can we reach with CVM?
  - In an intermediate step we considered the LossyStore algorithm/ How does it work?
  - How does the CVM algorithm work? What the connection to the LossyStore algorithm?
  - During the analysis of the failure probability of CVM we distinguished between two kinds of problems. Which ones?*

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables

- What would an ideal hash function be like? How would that be useful? What is the problem with this ideal version?
- What is the Simple Uniform Hashing Assumption (SUHA)? Why should we use it? What are alternatives?
- How is a cryptographic pseudorandom function with certain guarantees with respect to being indistinguishable from actuall random functions useful for us? How is that connected to SUHA?*

- Universal hashing
  - What is a c-universal hash family?
  - Which classes of c-universal hash functions did we encounter? How did we prove them to be c-universal?
  - How is d-independence defined for classes o hash functions?
  - Which classes of d-independent hash functions did we encounter?
  - What is the connection between c-universality and d-independence? (exercise)
  - Chernoff bounds are well suited for sums of independent random variables. What can we do if the random variables are only d-independent?*
- Hash tables with chaining
  - What bound on expected insertion time did we prove? How?
  - Where does the distribution of the hash function come into play?
  - Can you name a property of a class of universal hash functions that is sufficient for the proof to work?
- Hash tables with linear probing
  - What bound on expected insertion time did we prove? How?
  - Where does the distribution of the hash function come into play?
  - Can you name a property of a class of universal hash functions that is sufficient for the proof to work?
  - How did we use that property?*

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter

- Approximate membership query data structures
  - What is their task?
  - What is the advantage over exact data structures?
  - For what applications may we need one?
- Bloom filter
  - What does it consist of and which operations does it support?
  - How is it parameterized and how are the parameters related?
  - What did our analysis reveal about a good parameter choice? How do we choose the remaining parameters? What is the resulting memory requirement?
- About that analsysis
  - What are the expected numbers of zeroes and ones?
  - How is the false-positive probability related to the number of zeroes and ones?
  - How can we argue that the numbers of zeroes and ones in the bloom filter are close to their expectation?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing

- What is cuckoo hashing and what can it do?
  - What is the basic idea? How do operations work?
  - What do we need to consider about table size and load factor?
  - What do we know about the running time of the operations?
  - What are advantages and disadvantages with respect to other techniques like linear probing?
- Analysis
  - A failed insertion corresponds to certain structures in the cuckoo graph. Which ones?
  - How did we show that such structures are unlikely to exist?
  - How did we bound the expected insertion time?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing
- Peeling

- Cuckoo hashing and the peeling algorithm
  - (How) can you extend cuckoo hashing to use more than 2 hash functions?
  - What is the advantage over using 2 functions?
  - How does the peeling algorithm for placing keys in a cuckoo hash table work?
  - Peeling can be seen as a simple process on graphs. How?
  - What does the main result, that we proved about the peeling process, state?
- Proof of the peeling theorem (yes, that's a tough one)
  - In the proof we defined two graphs: a finite one and a (potentially) infinite one. How were they defined?
  - How are the degrees of the nodes in T_α and $G_{m,\alpha m}$ related?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing
- Peeling
- Retrieval and perfect hashing

- Retrieval data structures
  - What operations does a retrieval data structure support?
  - What are advantages and disadvantages compared to a normal hash table?
  - What can retrieval data structures be used for?
  - How can we construct a retrieval data structure using a peeling algorithm? What are construction and retrieval times? What are the memory requirements?
- Perfect hash functions
  - What properties does a good perfect hash function have?
  - We learned about hash tables without keys. What is that about?
  - How can we construct perfect hash functions using trial and error?
  - How can we construct a perfect hash functions using cuckoo hashing and retrieval? What are the memory requirements?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing
- Peeling
- Retrieval and perfect hashing

Things to Analyze

- Complexity classes

- Define: What is a PTM? What is the difference to an NTM?
- Define the complexity classes **RP**, co-**RP**, **BPP**, **PP**, **ZPP**
- What is the relevance of the constants $\frac{1}{2}, \frac{1}{4}, \frac{3}{4}$ in the definitions? With respect to what are they irrelevant?
- What is the relation between **ZPP** and Las Vegas algorithms? How do the two implications work?
- Which containment relationships are known for the complexity classes?
- For each contianment, can you explain why it is true?
- Are there relationships that we know to be strict? Are there classes that experts believe to be identical?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing
- Peeling
- Retrieval and perfect hashing

Things to Analyze

- Complexity classes
- Random graphs

- What is a random graph model?
  - What do we use them for?
  - What are desirable properties?
- How are Erdős-Rényi random graphs and Gilbert's model defined?
  - How are they related?
  - How do they differ?
- What properties do sparse $G(n, p)$ graphs have? (Degree distribution? Locality?)
- How did we show that the degree of a single vertex in a $G(n, p)$ is approximately Poisson-distributed?
- What are random geometric graphs? What are the degrees of freedom we have when defining them?
- What choices did we make for these degrees of freedom when defining *simple* random geometric graphs?
- How can we compute the expected degree of a node in a simple random geometric graph?
- What are geometric inhomogeneous random graphs? Why are they interesting? How do they compare to (simple) random geometric graphs?
- What do we have to do in order to compute the expected degree of a vertex with a given weight in a GIRG?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing
- Peeling
- Retrieval and perfect hashing

Things to Analyze

- Complexity classes
- Random graphs

Tools

- Coupling

- What is a coupling? What is it used for?
- Can you develop the simpler couplings we considered in the lecture?
- How is the total variation distance defined for the distributions of two random variables?
- What is the coupling inequality?
- What does the Binomial-Poisson approximation state?
  - How was a coupling used in the proof?
- How did we use the Binomial-Poisson approximation to approximate the distribution of a vertex degree in a $G(n, p)$?
  - What random variables did we consider?
  - How were they coupled?
  - What property of the total variation distance did we use there?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing
- Peeling
- Retrieval and perfect hashing

Things to Analyze

- Complexity classes
- Random graphs

Tools

- Coupling
- Concentration

- What is concentration? Why are we interested in it?
- What is Markov's inequality? When can it be applied? Can you prove its correctness? In what sense is it tight?
- What is Chebychev's inequality? When can it be applied? Can you prove its correctness?
- What is a (raw/centered) moment?
- In what sense can moments be used to characterize the shape of a distribution?
- What is a moment generating function? What does it have to do with moments?
- What are Chernoff bounds? Can you prove their correctness? How can we use them for specific probability distributions?
- What is the method of bounded differences? When can it be applied / yield useful bounds?
- What is the method of typical bounded differences?*
- How do the different concentration inequalities compare? Are some stronger than othres?
- Given a random variable, can you decide which concentration inequalities can be applied and which cannot?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing
- Peeling
- Retrieval and perfect hashing

Things to Analyze

- Complexity classes
- Random graphs

Tools

- Coupling
- Concentration
- Probabilistic method

- What is the probabilistic method? What is the basic idea?
  - What is the basic idea?
  - What are the two steps that we typically followed when applying the probabilistic method?
- What is the expectation argument?
  - Can you prove its correctness?
- Can you develop the simpler applications of the probabilistic method from the lecture?
- What is the Lovász Local Lemma?
  - About what kind of dependencies does it make a statement?
  - How does it relate to the probabilistic method?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing
- Peeling
- Retrieval and perfect hashing

Things to Analyze

- Complexity classes
- Random graphs

Tools

- Coupling
- Concentration
- Probabilistic method
- Continuous probability spaces

- How are probabilties measured in continuous spaces?
- What is a probability density function?
- How does working in continuous probability spaces differ from the discrete case?
- What is the memorylessness of the exponential distribution?
  - Can you derive it formally?
- What is a Poisson process?
  - What is its connection to the uniform/exponential distribution?
- How is independence defined for continuous random variables?
  - In that regard, what are joint / marginal / cumulative density functions?
- What is the Pareto distribution?
  - How can we determine for which parameter choices it has (in)finite expectation and variance?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing
- Peeling
- Retrieval and perfect hashing

Things to Analyze

- Complexity classes
- Random graphs

Tools

- Coupling
- Concentration
- Probabilistic method
- Continuous probability spaces
- Yao's principle

- Application to $\overline{\wedge}$-trees?
  - What was our goal when evaluating $\overline{\wedge}$-trees? (minimize query complexity)
  - What worst-case costs can we achieve with a deterministic approach?
  - Can a randomized algorithm do better? How?
  - One can relatively easily see that the randomized complexity is $\Omega(\sqrt{n})$. How?
  - We have also seen a tighter analysis. What components was it made of? In particular: How is Yao's principle applied there?
  - What does Tarsi's theorem state?
- Ski-rental problem
  - Define the problem
  - How do you call this kind of problem? (*online* problem)
  - Is this only relevant to winter sports? (Only key points)
  - What is the competitive ratio?
  - What is the best deterministic algorithm? Why?
  - Is there a randomized algorithm that can beat Break Even? (idea only)
  - Define Yao's principle for online algorithms
  - Which input distribution did we use for the lower bound in Ski-rental? What is the intuition?
  - What are the costs for the online and offline algorithms when using this input distribution? What can we say about the corresponding competitive ratio?

What have we learned?

Randomized Algorithms & Data Structures

- Probability amplification
- Approximation algorithms
- Streaming algorithms
- Classic hash tables
- Bloom filter
- Cuckoo hashing
- Peeling
- Retrieval and perfect hashing

Things to Analyze

- Complexity classes
- Random graphs

Tools

- Coupling
- Concentration
- Probabilistic method
- Continuous probability spaces
- Yao’s principle

Learnings from exercises relevant as well!

Where to go from here?

Exam

- Get your appointment by mailing Isabelle
- Prepare for the exam, reach out via Discord or mail if you have questions
- Max, Stefan, and Thomas will be in the room with you
- A typical exam may touch two topics from each of the two blocks

Other courses

- Algorithm Engineering
- Algorithmen zur Visualisierung von Graphen
- Algorithmische Graphentheorie
- Fortgeschrittene Datenstrukturen
- Modelle der Parallelverarbeitung
- Fortgeschrittenes algorithmisches Programmieren
- Algorithmische Geometrie
- Algorithmen für Routenplanung
- Parallele Algorithmen
- Parametrisierte Algorithmen
- Proofs from the Book
- Text-Indexierung

Master thesis

- Reach out!