UNIVERSITÄT DES SAARLANDES

ZBI ZENTRUM FÜR BIOINFORMATIK

Possible Exam Questions

Algorithms for Sequence Analysis

Sven Rahmann

Summer 2021

Recurring Topics

- Asymptotic analysis (big-O notation)
- Amortized analysis (KMP; Ukkonen's suffix tree)
- Poisson distribution ("law of rare events")

Exact Pattern Matching First Ideas

- State the pattern matching problem and known variants of it.
- What is the worst-case and average-case running time of the naïve algorithm?
- The naïve algorithm is fast on average; why bother with more complex algorithms?
- Explain Horspool's algorithm.
- Construct the Horspool shift table for a short pattern.
- For which pattern properties is Horspool's algorithm fast or slow?
- How may Horspool's algorithm be modified to be fast on long patterns with small alphabet?

Exact Pattern Matching with Automata

- How can finite automata be used to solve the pattern matching problem?
- What is the difference between NFAs and DFAs?
- What running times can be achieved in NFA/DFA based pattern matching?
- How is the set of active NFA states related to the read text so far?
- Give the formal definition of a pattern matching NFA and explain it.
- Explain the Shift-And algorithm.
- Explain the subset construction (from NFA to DFA).
- Why do the special NFAs studied here have the same number of states as the corresponding DFAs?
- Explain the Knuth-Morris-Pratt (KMP) algorithm and its relation to DFAs.
- How can one construct the *lps* function and in what time?
- What is the running time of KMP (worst case / best case)?

Exact Pattern Matching with Bit-Parallel Algorithms

- Explain the idea of bit-parallel simulation of NFAs.
- Explain the suffix automaton and the BNDM algorithm.
- What are the advantages of BNDM over Horspool's algorithm?
- What are the advantages of BNDM over the Shift-And algorithm?
- What is a generalized string?
- How does the Shift-And algorithm change when you allow generalized strings?
- Why would you want to use the Shift-And algorithm for runs with bounded length, when the algorithms for optional characters is more general ($\#(3,5) = \texttt{\#?\#?\#\#\#}$)?
- How do you implement bit-parallel propagation of an active state?

Suffix Trees

- Define a suffix tree. What is a suffix trie?
- Construct the suffix tree with suffix links of an example string.
- What is the running time of pattern search with a suffix tree?
- How can the longest repeated substring problem and the shortest unique substring problem be solved in optimal time with suffix trees?
- Explain Ukkonen's algorithm.
- What is the important trick to achieve linear space consumption in Ukkonen's algorithm?
- What is a suffix link? What are suffix links used for in Ukkonen's algorithm?
- Apply Ukkonnen's algorithm to an example string.
- Why does Ukkonen's algorithm run in $O(n)$ time?
- Explain the skip & count trick.
- Explain how one could implement the elements of a suffix tree.
  What are alternative ways of storing the children of a suffix tree node?

Suffix Arrays

- Define a suffix array.
- Construct a suffix array for an example string.
- Explain pattern search in suffix arrays.
- Give the definition of the LCP array and explain it.
- Construct the LCP array for a given string.
- What is the advantage of an enhanced suffix array over a suffix tree?
- Explain the following problems and how they can be solved using an enhanced suffix array: longest repeated substring, shortest unique substring, longest common substring, maximal unique matches.
- Why and how can a suffix array be inverted?
- Explain Kasai's algorithm. What is its running time?
- Apply Kasai's algorithm to a given example.

Linear Time Suffix Array Construction

- Explain the principle of induced sorting.
- Why are L-positions on the left and S-positions on the right of each bucket?
- What is the goal of the text reduction step?
- Conduct the first iteration of induced sorting for a small example string.
- Explain why the induced sorting algorithm has linear running time.

Connections between Suffix Trees and Arrays

- How are suffix tree leafs related to suffix arrays?
- Which suffix tree operations can be simulated using suffix array plus LCP array?
- What is a range minimum query (RMQ)?
- How can RMQs be answered in constant time after at most $O(n \log n)$ preprocessing?
- Why are RMQs on the LCP useful?
- What is needed to do $O(|P|)$ time pattern matching with a suffix array?
- Explain how to achieve linear time/space preprocessing for constant time RMQs.
- What is the lowest common ancestor (LCA) problem?
- How is LCA connected to RMQ?
- What is a ± 1RMQ?

The Burrows-Wheeler Transform (BWT)

- Define the BWT.
- What is the relation of the BWT to the suffix array of the same string?
- Compute the BWT for a given string.
- Compute the original string from a given BWT.
- Define the Last-to-First (LF) mapping.
- Why is it useful?
- How can the LF-mapping be substituted by `C` and `Occ`?
- What is an FM-index?
- Explain backward pattern search with the FM-index.

The FM Index

- Why and how is the FM index compressed?
- How can rank (`Occ`) queries be implemented in constant time with succinct space?
- What is a wavelet tree? How does it support character and rank queries?
- What are the successor / predecessor arrays? Construct an example.
- Explain sparse suffix arrays.
- How long does a query on a sparse suffix array take in the worst case?
- How can one determine the position of pattern matches for a BWT interval?

Text Compression

- Why can the BWT be easier to compress than the input string?
- What is run length encoding?
- Explain Move-to-Front encoding. Apply it to an example.
- Explain Huffman coding.
- Define the LZ77 factorization.
- How do you efficiently compute the LZ77 factorization?
- Define the LZ78 factorization.
- How do you efficiently compute the LZ78 factorization?
- What if the alphabet is not of constant size, but grows as e.g., $\sqrt{n}$:
  How does the time of the LZ77 and LZ78 factorization algorithms change?

Distance and Similarity Measures between Strings

- How can the distance between strings be measured?
- How long does it take to compute the Hamming distance between two strings?
- And for the edit distance?
- What is an alignment of two strings?
- How are alignment and edit distance related?
- Compute an optimal global alignment for two given strings.
- Give the recursive formulation of edit distance computation.
- How can edit distance computation be formulated as a graph problem?

Error Tolerant Pattern Matching I

- Define variations of the error tolerant pattern matching problem.
- How does error-tolerant pattern search relate to semi-global alignment?
- How does the edit / alignment graph differ from that for global alignment?
- Explain Ukkonen's speed-up (in theory / on a small example)
- Explain the horizontal, vertical, and diagonal properties.
- What are the ideas to prove these properties?
- Why are these properties helpful?
- What is the idea behind Myers' bit vector algorithm?
- How does the original DP matrix relate to the used bit vectors?

Error Tolerant Pattern Matching II

- Explain how the Shift-And algorithm can be adjusted to solve the approximate pattern matching problem.
- Explain the semantics of the states in the corresponding NFA.
- Explain the meaning of the different types of edges.
- How many states are always active for a NFA that allows k mismatches?
- How exactly does the bit-parallel update of the active state matrix A work?
- How can backward search be applied to error tolerant search?
- Explain the idea of the Four Russians Technique.
- Why is the block size chosen as $t := 1 + (\log_{3\sigma} n)/2$ in the Four Russians Method?

Pairwise Sequence Alignments

- Define alignment (in general).
- Define global / semiglobal / etc. alignment of two strings s, t.
- Explain four variants of alignments and their applications / use cases.
- What is the difference between score and cost function and why is it important?
- Why can't we use costs for free end gap and local alignment?
- How can sequence alignment be formulated as a graph problem?
- Show the alignment graph topology for each variant.
- Explain the universal alignment algorithm on the alignment graph.
- Give the DP formulation for computing an alignment score (any variant).
- Compute an optimal alignment (any variant) for two given strings.
- Explain traceback.

Score Matrices

- Define joint frequencies J, transition probabilities P, marginal probabilities π.
- Describe how to compute log-odds scores.
- Explain how the BLOSUM and PAM matrix families were constructed.
- What is a rate matrix?
- Define the evolutionary time units 1 PAM and 1 PEM.
- How can Q be expressed in terms of P or all $P^{(t)}$?
- How can you estimate the divergence time of an observed alignment?
- What are the advantages of the resolvent method for estimating Q?
- Are score matrices symmetric? Why? When not?

Alignment Statistics

- Define the p-value and E-value of an observed local alignment score.
- Why is p-value $\approx$ E-value when both are very small?
- Explain why the parametric form $p(s) = C \cdot \exp(-\lambda s)$ holds.
- How do the sequence lengths m, n enter the parametric form?
- How can the parameters C, λ be estimated?

Extensions and Improvements of Pairwise Sequence Alignment

- What are linear vs. affine gap costs?
- Explain how to implement alignment with general gap costs (time?)
- Explain how to implement alignment with affine gap costs (time?)
- How can alignments (tracebacks) be obtained in $O(m+n)$ space instead of $O(mn)$ space? Illustrate why this is crucial in practice.
- How is the running time affected by linear-space traceback?
- Illustrate two conceptual problems with local alignment.
- Explain the idea of length-normalized local alignment.
- What is the role of the parameter $L > 0$?
- Why can't we use the standard DP algorithms for length-normalized alignment?
- How can we efficiently find the optimal length-normalized alignment?

Genome-Wide DNA Read Mapping (or DNA Database Search)

- How can a suffix tree be used to search for approximate pattern occurrences?
- What is the resulting running time?
- What is the benefit of using an FM index instead of a suffix tree?
- Explain approximate search on an FM index by means of an example.
- What's the main idea behind the "seed and extend" paradigm?
- What is the purpose of a filter?
- What is a lossy vs. a lossless filter?
- Explain the q-gram lemma.
- What is a q-gram index? How is it related to the suffix array?
- What is the idea of gapped q-grams?

Locality Sensitive Hashing

- How can a k-mer index be implemented?
- What is the disadvantage of hash-based vs. encoding-based implementations?
- How can k-mers be mapped bijectively to the integers $0, \ldots, 4^k - 1$?
- What are some common collision resolution strategies when hashing?
- Explain (h, b) Cuckoo hashing
- Why are the advantages and disadvantages of (h, b) Cuckoo hashing?
- When is a set of hash functions "locality sensitive"?
- Why is standard hashing usually not locality sensitive?
- Explain min-hashing.
- Why is min-hashing locality sensitive for the Jaccard similarity?

Min-Hashing and Applications

- Prove that min-hashing is LS for Jaccard similarity
- What is a sketch?
- Why are sketches useful for similarity search in high-dimensional spaces?
- What are minimizers (precisely, (w, k)-minimizers) of a sequence?
- What property does the set of minimizers of a sequence have?
- What is the effect of changing the window size w?
- What is the effect of changing the k-mer size k?
- Name some application areas of (w, k)-minimizers in bioinformatics.

Genome Assembly

- What are the main approaches to genome assembly?
- Define a de Bruijn graph for genome assembly.
- Construct a de Bruijn graph for a given example.
- What is the effect of sequencing errors on a DBG?
- Explain strategies to remove such errors.
- Mention different representations for de Bruijn graphs.
- Which representation of a DBG is the most space-efficient?
- Why can Bloom filters have false positives?
- What is a critical false positive?

Multiple Sequence Alignment I

- Define a global multiple sequence alignment (MSA).
- What is the advantage of an MSA compared to a pairwise alignment?
- Define the sum-of-pairs objective for MSA.
- Do you know an algorithm to find the optimal MSA (wrt this objective)?
- What is its time and space complexity?
- Is there an exact algorithm with running time polynomial in the number of sequences k?
- What is an approximation algorithm?
- Explain the Center Star algorithm and its assumptions.
- What is its running time?
- Can you sketch the ideas for proving it to be a 2-approximation?

Variation and Conservation in MSAs

- How are haplotype panels related to multiple sequence alignments?
- What is a positional prefix array?
- What are commonalities/differences to a suffix array?
- Explain how to build the positional prefix arrays for all columns in $O(MN)$ time.
- How does this algorithm relate to induced sorting?
- Define the divergence array.
- What are commonalities and differences between the divergence array and the LCP array?
- How can the divergence array be constructed efficiently?
- Explain how to find all positional length-L matches in a binary matrix X.