1. Explain the difference between a deterministic, Monte Carlo, and Las Vegas algorithm using exact pattern matching as an example. You don't have to explain how the algorithms work inside, just what they achieve.

2. What is the Karp-Rabin-style fingerprint ϕ_r? State the lemma describing the behaviour of such fingerprint if we choose the parameter r at random. Why we should compute the the fingerprint modulo a prime instead of, say, a power of 2?

3. Define borders, periods, and state the relation between them. Explain how to compute the border of each prefix of a given word efficiently.

4. Explain the idea behind the Knuth-Morris-Pratt algorithm. State what is the information maintained during the computation and what is the invariant of the procedure.

5. State (informally) what is the streaming model using exact pattern matching as an example. What is the space complexity of the Knuth-Morris-Pratt in such model? Can we hope to get a significantly better complexity with a deterministic algorithm?

6. What is the periodicity lemma? What was (at a very high level) the idea behind its proof?

7. Define the Levenshtein distance. Show how to apply dynamic programming to compute such distance between two strings.

8. How the Hirschberg algorithm for reconstructing the longest common subsequence works?

9. How the Myers algorithm for computing the edit distance in $\mathcal{O}(nD)$ time works?

10. What is the Four Russians technique?

11. Define the suffix array. What additional data is usually stored together with the array? Why is it useful for finding an occurrence of a given pattern?

12. Explain (at a high level) the idea behind the linear time suffix array construction algorithm. State the recurrence describing its running time.

13. What is LCP and RMQ? How the `lcp` array can be used to compute the longest common prefix between any two substrings of our word?

14. How computing longest common prefixes is used in speeding up binary searching in the suffix array? State the invariant maintained during the improved procedure.

15. How to solve RMQ in $\mathcal{O}(\log n)$ time after a $\mathcal{O}(n)$ space preprocessing?

16. Define pattern matching with mismatches and errors. Show how to solve pattern matching with k mismatches in $\mathcal{O}(nk)$ time.

17. Define pattern matching with don't cares. What was the convolution of two vectors? How were the vectors we computed the convolution of defined in the simple case of $\Sigma = \{a, b\}$?

18. What is a word equation? Give an example where there are is a (positive) finite number of solutions, and an example where there is an infinite number of solutions.

19. Explain the difference between the Lempel-Ziv (LZ77) and Lempel-Ziv-Welch (LZ78/LZW) compression. Show an example where the size of a compressed representation of a text is substantially larger in one of them.

20. What is the Burrows-Wheeler transform? How to compute the transform efficiently? How to use the LF to reverse the transform? State formula that allows us to compute the LF array efficiently.

21. What is the zeroth entropy of a text? Explain how the Huffman encoding (which almost achieves the zeroth entropy bound) works.

22. What is the shortest superstring problem? Explain the definition of a c-approximate algorithm using this problem as an example. Define the prefix graph and state the relation between the shortest superstring and the cheapest cycle cover there.