Analysis of algorithms and heuristic problem solving: sample questions for oral exam 2020/21

Analysis of computational complexity

1. Suppose the numbers are not presented with a fixed size representation in the RAM model of computational complexity, but the representation depends on the size of numbers. How this affects the analysis of computational complexity?
2. What are suitable input sizes used in the analysis of computational complexity? Give a few examples.
3. For very large inputs, what are the limits of computational complexity to get practically useful algorithms?
4. Explain the differences between different asymptotical bounds of computational complexity: tight Θ, upper O, and lower bound Ω. What are the relations between them?
5. What is the practical use of asymptotical imprecise bounds o and ω?
6. Explain different approaches to the analysis of divide and conquer algorithms? Present strengths and weaknesses of each.
7. Explain the difference between Akra-Bazzi and the master method for analysis of divide and conquer algorithms.
8. Which recurrences cannot be analysed with the master method even if the splits are equal? Explain the holes between the three cases of the master approach?
9. Which types of divide and conquer algorithms cannot be solved with the Akra-Bazzi method? Give an example.
10. What are the advantages of using the strong form of the Akra-Bazzi theorem compared to the base form?
11. Which type of linear recurrences cannot be solved with annihilators? Give an example? What happens if the obtained polynomial cannot be factored with real number coefficients?
12. What is the difference between probabilistic analysis of algorithms and randomization of algorithms?
13. What is the purpose of randomization? How do we achieve it?
14. Why do we use pseudo-random numbers instead of better hardware generators?
15. Present a few ideas to get pseudo-random generators and a few ideas to test their randomness?
16. Present the differences, strengths, and weaknesses of three methods for the amortized analysis of algorithms?
17. Why shall amortized costs always be higher than real costs in the accounting/potential method?
18. How to analyse a multithreaded algorithm?
19. Explain races in parallel algorithms?
20. Describe parallel speedup?
21. Describe the limitations of parallelization expressed through Amdahl and Gustafson's laws.

NP-completeness and approximation algorithms

22. What does the NP-completeness of an algorithm mean?
23. What is the relation between classes of problems P, NP, NP-hard, and NP-complete?
24. What is the verification algorithm, and what role it plays in determining the NP-completeness?
25. What is the polynomial-time reduction between two algorithms?
26. How to prove the NP-completeness of an algorithm?
27. What is the approximation ratio?
28. What strategies for proving approximation ratios of algorithms exist?
29. What is the expected approximation ratio of an algorithm?
30. What is an approximation scheme?
31. Which approximation algorithms do you know? What are the ideas of proofs for their approximation ratios?

Linear programming (LP)

32. Present the (standard) linear programming problem? What are the feasible solutions?
33. What types of constraints are allowed in LP?
34. How to convert different types of inconsistencies into the standard LP?
35. How do we deal with strict inequality constraints in LP?
36. How to formulate different problems into LPs (shortest path, maximum flow, minimum-cost flow, multicommodity flow)?
37. What is LP relaxation?
38. What are 0-1 programming and integer programming?

Local search (LS) and metaheuristics

39. Define the search space, local, and global optimum?
40. What are plato and ridge? How they affect the local search?
41. Explain the idea of the Metropolis algorithm and simulated annealing?
42. What is the difference between the stochastic and deterministic LS?
43. Discuss different neighborhoods and their complexity in LS?
44. Explain the multicast routing problem?
45. Explain the Nash equilibrium, social choice, and the price of stability?
46. Explain the relation between the local extreme and Nash equilibrium?
47. What is a metaheuristic? Give examples.
48. Explain ideas of tabu search, guided local search, and variable neighbourhood search?
49. Give examples of different metaheuristics classes, e.g., nature-inspired vs. non-nature inspired, population-based vs. single point search, dynamic vs. static objective function, one vs. various neighbourhood structures, memory usage vs. memory-less methods?
50. Explain a few extensions of the vehicle-routing problem and how to integrate them into local search and metaheuristics?
51. Describe different types of tabus and different implementations of tabu lists in tabu search.

52. Describe the intention and implementations of intensification and diversification in different types of metaheuristics.
53. What is the purpose of surrogate and auxiliary objectives in metaheuristics?
54. Describe the purpose of different components in the guided local search?
55. Describe the main components of the workforce scheduling problem.
56. What is the main idea of variable neighbourhood search, and how to implement it?

Nature-inspired computation

57. Describe the main ideas of swarm intelligence, their advantages, and shortcomings.
58. Describe the main ideas of ant colony optimization (ACO), their strengths and weaknesses.
59. How can ACO be used for dynamic optimization?
60. What is the Max-Min ant system?
61. How to use ACO to solve the travelling salesman problem/quadratic assignment problem?
62. How to use ACO for rule learning?
63. Describe the main ideas and components of particle swarm optimization (PSO), its advantages and disadvantages.
64. Describe crossover, mutation, and selection in genetic algorithms and differential evolution (DE)? How they differ?
65. Describe the main parameters of DE?
66. Describe the notation DE/x/y/z.
67. Give examples for the hybridization of DE.
68. Describe approaches for adaptation of differential evolution algorithm to discrete problems. What is angle modulation?
69. Explain L-SHADE (success-history based adaptive differential evolution with linear reduction of population size) and its difference to classical DE/rand/1/bin variant.
70. Describe the main categories of nature-inspired methods, e.g., evolution-based, swarm-based, physics-based, and human-based.
71. Describe the similarities and differences between evolution-based methods: evolution strategies, genetic algorithms, genetic programming, and differential evolution?
72. Describe the main representatives of swarm-based methods: particle swarm optimization, ant colony optimization, artificial bee colony, firefly algorithm, and social spider optimization, grey wolf optimizer, cuckoo search.
73. Describe the main representatives of physics-based methods: simulated annealing, gravitational search, electromagnetism-like mechanism, and states of matter search.
74. Describe the main representatives of human-based search: harmony search, firework algorithm, artificial immune systems.
75. Describe the similarities and differences between the differential evolution (DE) algorithm and artificial bee colony (ABC). Which variant of DE is very similar to ABC? Justify the answer.
76. Describe the similarities and differences between the differential evolution (DE) algorithm and grey wolf optimizer (GWO). Which variant of DE is very similar to GWO? Justify the answer.

77. Suppose you want to solve the classical vehicle routing problem with the ACO. Describe the necessary setting: the search graph, the costs, constraints, deposition, and evaporation of pheromones, probability of choosing a certain edge, etc.
78. Explain the following three mutation strategies of differential evolution algorithm and their differences: DE/rand/1/exp, DE/best/2/bin, DE/current-to-best/1/bin.