Algorithms and Data Structures.

Examination questions.

Year: 2021

Organizational issues:

- *You need to receive at least 38 pts before the exam to be admitted.*
- *You need to receive at least 22 pts at the exam to pass it successfully.*
- *Each question is scored 10, 20 or 30 pts. The score is the maximum score you get for perfect answer.*
- *Each ticket contains 2 questions that have the total score of 40 pts.*

General requirements for the answers:

- *For questions that consider algorithm, time complexity estimation must be supplied.*
- *For questions that consider data structure, basic operations time complexity estimation must be supplied.*
- *For each question you should supply an example that demonstrates definitions and algorithm.*

Module 1. Introduction to algorithms and data structures.

1. Insertion sort: algorithm; using array and dynamic list to implement insertion sort. Time and space complexity estimation. {20 pts}
2. Graphs: definition; edge list representation. Time complexity of basic operations using edge list representation. {10 pts}
3. Graphs: definition; adjacency matrix representation. Time complexity of basic operations using adjacency matrix representation. {10 pts}
4. Graphs: definition; adjacency list representation. Time complexity of basic operations using adjacency list representation. {10 pts}
5. Depth-first search: recursive implementation; stack-based implementation. {20 pts}
6. Stack: abstract data structure; array-based implementation. {10 pts}
7. Stack: abstract data structure; linked list-based implementation. {10 pts}
8. Queue: abstract data structure; linear array-based implementation. {10 pts}
9. Queue: abstract data structure; circular array-based implementation. {10 pts}
10. Queue: abstract data structure; linked list-based implementation. {10 pts}
11. Breadth-first search: queue-based implementation. {10 pts}
12. Connected components detection with DFS or BFS. {10 pts}
13. Bipartiteness check algorithm. {10 pts}
14. DAGs: definition, topological sort (definition and algorithm). {20 pts}

Module 2. Greedy algorithms. Divide-and-Conquer strategy.

15. Minimum spanning tree: problem definition. {10 pts}
16. Kruskal's algorithm: general scheme; (improved) implementation with array-based safety check procedure. {20 pts}
17. Kruskal's algorithm: general scheme; implementation with safety check procedure based on reversed trees data structure. {20 pts}
18. Prim's algorithm: general scheme; naïve and priority queue-based implementation. {20 pts}
19. Priority queue: abstract data structure, unsorted array implementation. {10 pts}
20. Priority queue: abstract data structure, sorted array implementation. {10 pts}
21. Priority queue: abstract data structure, dynamic linked sorted list implementation. {10 pts}
22. Priority queue: abstract data structure, binary search tree implementation. {20 pts}
23. Priority queue: abstract data structure, 2-3 tree implementation. {20 pts}
24. Binary heap: abstract data structure, complete binary tree implementation. {30 pts}

25. Merge sort as a divide-and-conquer algorithm. {20 pts}
26. Master theorem for divide-and-conquer algorithms. Proof of the simplified version of Master theorem. {20 pts}
27. Integer multiplication: the standard and Karatsuba's algorithms. {20 pts}
28. Fast exponentiation algorithm. {10 pts}
29. Strassen's algorithm for matrix multiplication. {10 pts}

Module 3. Dynamic programming.

30. A general scheme of a simple version of a dynamic programming algorithm and meta-algorithm for transforming a recursive algorithm into a DP algorithm. Demonstration for Fibonacci numbers calculating. {20 pts}
31. Graph distances: definitions, three problem formulations. {10 pts}
32. The principle of optimality for graph distances. Troubles with negative weights. {20 pts}
33. Dijkstra's algorithm for a general graph without negative cycles. Building a shortest path.
34. Dijkstra's algorithm for a graph without negative edges. Building a shortest path.
35. Floyd-Warshall's algorithm for the all-t-all distances problem. Building a shortest path.
36. Edit distance: general definition, Levenshtein distance, algorithm for calculating. {30 pts}
37. Longest common subsequence problem: definition, Needleman-Wunsch algorithm. {20 pts}
38. Building an optimal binary search tree: problem definition, recurrence formula, algorithm. {30 pts}

Module 4. NP-hard problems.

39. Travelling salesman problem: definition. {10 pts}
40. Metric TSP: definition. {10 pts}
41. Brute-force algorithm for TSP. {20 pts}
42. Branch-and-bound method: the general scheme. {20 pts}
43. Branch-and-bound method for TSP: bound function variants. {10 pts}
44. Performance ratio of an approximate algorithm: definition. {10 pts}
45. 2-approximate algorithm for Metric TSP. {20 pts}
46. Three greedy heuristics for TSP. {20 pts}
47. Local search: the general scheme and its features. {20 pts}
48. Local search algorithms for TSP. {20 pts}
49. Tabu search. {20 pts}
50. Randomizations for local search. {10 pts}

Recommended sources

1. The course section at the Moodle: http://edu.mmcs.sfedu.ru/course/view.php?id=551
2. Open Data Structures. An open content textbook. URL:http://opendatastructures.org/
3. Jeff Erickson. Algorithms. A free electronic version. URL: http://jeffe.cs.illinois.edu/teaching/algorithms/
4. Clifford A. Shaffer. Data Structures & Algorithm Analysis. A free for educational use electronic book. URL: https://people.cs.vt.edu/~shaffer/Book/
5. The Computer Science Handbook. A Reference for Data Structures and Algorithms. URL: https://www.thecshandbook.com/
6. Lectures Notes on Algorithm Analysis and Computational Complexity (Fourth Edition) - Ian Parberry: URL: http://ianparberry.com/books/free/license.html
7. Kleinberg J., Tardos É. Algorithm Design. — Pearson Education Inc., 2006.
8. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd ed. — The MIT Press — 2009.