Changchun University of Science and Technology trainig. Topic survey.

*** 1. Please, rate your knowledge of the following algorithms and data structures.**

A - I know all about this algorithm and it's variations, I have used in while solving ACM problems, I can easily implement and use it again
B - I heard a lecture or two about this algorithm, I may have implemented it one or two times, but my knowledge can be improved
C - I heard something about this algorithm, but I never implemented it
D - I have never heard about it

	A (excellent knowledge)	B (well knowledge)	C (poor knowledge)	D (absence of knowledge)
General knowledge: computational complexity	○	○	○	○
General knowledge: big-O notation	○	○	○	○
Sorting and search: quick sort	○	○	○	○
Sorting and search: merge sort	○	○	○	○
Sorting and search: inversion count	○	○	○	○
Sorting and search: binary search	○	○	○	○
Sorting and search: ternary search	○	○	○	○
Dynamic programming (in general)	○	○	○	○
Dynamic				

Dynamic programming: subtrees	○	○	○	[✓ ○]
Dynamic programming: subsegments	○	○	○	[✓ ○]
Dynamic programming: DP by profile	○	○	○	[✓ ○]
Dynamic programming: subsets	○	○	[✓ ○]	○
Graph theory: adjacency matrix, edge lists	○	○	○	[✓ ○]
Graph theory: connected components	[✓ ○]	○	○	○
Graph theory: BFS	○	○	[✓ ○]	○
Graph theory: DFS	○	[✓ ○]	○	○
Graph theory: DFS applications (bridges, cut vertices, strong connectivity)	○	○	[✓ ○]	○
Graph theory: MST (Prim's algorithm)	○	○	○	[✓ ○]
Graph theory: MST (Cruscal's algotirhm)	○	○	○	[✓ ○]
Graph theory: Dijkstra algorithm	○	○	○	[✓ ○]
Graph theory: Floyd-Warshall algorithm	○	○	○	[✓ ○]
Graph theory:	○	○	○	[✓ ○]

maximum flow, minimum cut	○	○	○	[✓ ○]
Graph theory: Ford-Fulkerson algorithm	○	○	○	[✓ ○]
Graph theory: Dinic's algorithm	○	○	○	[✓ ○]
Graph theory: maximum flow of minimum cost	○	○	○	[✓ ○]
Graph theory: maximum matching in bipartite graph	○	○	○	[✓ ○]
Data structures: stack	[✓ ○]	○	○	○
Data structures: queue	[✓ ○]	○	○	○
Data structures: deque	○	○	○	[✓ ○]
Data structures: segment tree	○	○	○	[✓ ○]
Data structures: binary search tree	○	○	[✓ ○]	○
Data structures: cartesian tree (treap)	○	○	○	[✓ ○]
Data structures: implied cartesian tree (rope)	○	○	○	[✓ ○]
Data structures: group operations on segment/cartesian tree	○	○	○	[✓ ○]
Data structures: DSU (disjoint set union)	○	○	○	[✓ ○]

Computational

Computational geometry: line equation, vectors, points	○	○	[✓ ○]	○
Computational geometry: scalar product, vector product	[✓ ○]	○	○	○
Computational geometry: line intersection	○	[✓ ○]	○	○
Computational geometry: segment intersection	○	[✓ ○]	○	○
Computational geometry: convex hull	○	○	○	[✓ ○]
Computational geometry: tangent line to circle, circle intersection	○	○	[✓ ○]	○
Computational geometry: tangent line to convex polygon	○	○	○	[✓ ○]
Combinatorics: permutations, binary vectors, combinations	[✓ ○]	○	○	○
Combinatorics: generating i-th lexicographical object	○	○	○	[✓ ○]
Combinatorics: generating lexicographical number of object	○	○	○	[✓ ○]

Done