算法设计与分析课程教学大纲

<table>
<tr><td colspan="6">课程基本信息（Course Information）</td></tr>
<tr><td>*课程代码
（Course Code）</td><td>CS217</td><td>*学时
（Credit Hours）</td><td>48</td><td>*学分
（Credits）</td><td>3</td></tr>
<tr><td rowspan="2">*课程名称
（Course Name）</td><td colspan="5">（中文）算法设计与分析</td></tr>
<tr><td colspan="5">（英文）Algorithms Design and Analysis</td></tr>
<tr><td>课程性质
(Course Type)</td><td colspan="5">专业必修课（Compulsory course）</td></tr>
<tr><td>授课对象
（Audience）</td><td colspan="5">Undergraduate students (ACM class)</td></tr>
<tr><td>授课语言
(Language of Instruction)</td><td colspan="5">English</td></tr>
<tr><td>*开课院系
（School）</td><td colspan="5">Zhiyuan College</td></tr>
<tr><td>先修课程
（Prerequisite）</td><td colspan="5"></td></tr>
<tr><td>授课教师
（Instructor）</td><td colspan="2">Dominik Scheder</td><td>课程网址
(Course Webpage)</td><td colspan="2">https://basics.sjtu.edu.cn/~dominik/teaching/index.html</td></tr>
<tr><td>*课程简介
（Description）</td><td colspan="5">（中文 300-500 字，含课程性质、主要教学内容、课程教学目标等）</td></tr>
<tr><td>*课程简介
（Description）</td><td colspan="5">Classical algorithms and algorithmic design paradigms (e.g. divide and conquer, greedy algorithms, dynamic programming); maximum flow and minimum cut; Linear programming (duality, simplex, applications). We will also discuss some important data structures and how to use them in algorithms. Depending on the time frame, we will teach more advanced topics like randomized, approximation, exponential, and streaming algorithms. We will devote ample time to: (i) the path from a basic idea to the final algorithm; (ii) proofs of correctness; (iii) running time analysis.</td></tr>
<tr><td colspan="6">课程教学大纲（course syllabus）</td></tr>
</table>

*学习目标(Learning Outcomes)	Algorithmic thinking. Mastery of several algorithm paradigms and ability to apply them to new problems. Deep understanding of the mathematical structure that underlies algorithms. Ability to apply the learned methods and paradigms to new problems.

*教学内容、进度安排及要求 (Class Schedule & Requirements)	1. Introdcution, administrative stuff. Euclid's algorithm. Running time, exact running time. Fibonacci number, linear recurrences; recursive and dynamic programming algorithm or Fibonacci numbers. 2. Computing Fibonacci numbers (maybe modulo k) via fast matrix multiplication. Algorithms for multiplying integers. 3. Mergesort, worst case and best case; 4. Quicksort, worst case and expected case. Indicator variables $A_{i,j}$ which are 1 if i is an ancestor of j. 5. Vector Domination Problem: a kind of multidimensional quicksort 6. Spanning trees in graphs. Kruskal's algorithm for finding the minimum spanning tree. Good edge sets, cut lemma. 7. Keeping track of connected components: Union-Find data structures: naive list approach; "rename the smaller one" improvement. Union find with path compression. 8. Shortest path algorithms: breadth-first search and Dijkstra's algorithm. Longest paths in graphs. Longest paths in acyclic graphs. The randomized color coding technique. 9. Flow networks. Network flows, cuts, residual networks. Maxflow-Mincut theorem. Ford-Fulkerson method and Edmonds-Karp algorithm. 10. Improving Edmonds-Karp: Dinic algorithm finding paths in "$O(n)$" steps in a layered digraph. Dinic algorithm on unit capacity networks. 11. Matchings in bipartite graphs. Finding a maximum matching using maximum flow algorithms (e.g. Dinic' algorithm). Hall's Theorem. König's Theorem (maximum matching equals minimum vertex cover in bipartite graphs) and Dilworth's theorem on chains and antichains in partial orders. 12. Introduction to linear programming. Feasible solutions, standard form, dualizations, weak LP duality. 13. How to dualize linear programs with equality constraints and unbounded variables. 14. Integrality of the Matching LP and the Vertex Cover LP for bipartite graphs. 15. Existence of optimal solutions for every bounded feasible linear program. 16. Farkas Lemma (geometric version) and Strong LP Duality. Different versions of Farkas Lemma. Proof of Farkas Lemma by Fourier-Motzkin elimination. 17. Basic probability theory. 18. Streaming algorithms; approximating the median. 19. Streaming algorithms; median and second moment. 20. Algorithms for NP-complete problems: colorability. 21. Algorithms for NP-complete problems: satisfiability.

*考核方式 (Grading)	50% homework, 50% final exam
*教材或参考资料 (Textbooks & Other Materials)	Algorithms by Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani
其它 （More）	
备注 （Notes）	

备注说明：

1．带*内容为必填项。

2．课程简介字数为 300-500 字；课程大纲以表述清楚教学安排为宜，字数不限。