Algorithm Design (XXIV)

Conclusion

Guoqiang Li
School of Computer Science

SHANGHAI JIAO TONG UNIVERSITY

What Is Algorithm

Algorithm Design

Basic algorithms:

Algorithm Design

Basic algorithms:

- Recursion

Algorithm Design

SHANGHAI JIAO TONG UNIVERSITY

Basic algorithms:

- RECURSION
- ALGORITHMS ON LISTS, TREES AND GRAPHS

Algorithm Design

Basic algorithms:

- Recursion
- Algorithms on Lists, Trees and Graphs

Advanced strategies:

Algorithm Design

Basic algorithms:

- Recursion
- Algorithms on Lists, Trees and Graphs

Advanced strategies:

- Divide and Conquer

Algorithm Design

Basic algorithms:

- Recursion
- Algorithms on Lists, Trees and Graphs

Advanced strategies:

- Divide and Conquer
  - Master Theorem
- Dynamic Programming
- Greedy
- Duality
- Reduction

Algorithm Design

Basic algorithms:

- Recursion
- Algorithms on Lists, Trees and Graphs

Advanced strategies:

- Divide and Conquer
  - Master Theorem
- Dynamic Programming
- Greedy
- Duality
- Reduction
- Approximation
- Randomization
- Computational Geometry
- Algorithms on Massive Data
- ...

Graphs

Algorithms on Special Structures

Graphs

- undirected graphs, directed graphs.

Graphs

- undirected graphs, directed graphs.
- DAG.

Algorithms on Special Structures

Graphs

- undirected graphs, directed graphs.
- DAG.
- bipartite.

Algorithms on Special Structures

Graphs

- undirected graphs, directed graphs.
- DAG.
- bipartite.
- graphs with weights.

Algorithms on Special Structures

Shanghai Jiao Tong University

Graphs

- undirected graphs, directed graphs.
- DAG.
- bipartite.
- graphs with weights.
- ...

Algorithms on Special Structures

Graphs

- undirected graphs, directed graphs.
- DAG.
- bipartite.
- graphs with weights.
- ...

Network flows

- Ford-Fulkerson algorithm, Edmonds-Karp algorithm
- ...

Computational Geometry

Algorithm Analysis

Big-O Notation (Ω, Θ)

Algorithm Analysis

Big-O Notation (Ω, Θ)

Advanced Methodology:

Algorithm Analysis

Big-O Notation (Ω, Θ)

Advanced Methodology:

- Probability Analysis

Algorithm Analysis

Big-O Notation (Ω, Θ)

Advanced Methodology:

- PROBABILITY ANALYSIS
- AMORTIZED ANALYSIS

Algorithm Analysis

Big-O Notation (Ω, Θ)

Advanced Methodology:

- PROBABILITY ANALYSIS
- AMORTIZED ANALYSIS
- COMPETITION ANALYSIS

Standard Algorithms

Standard Algorithms

SHANGHAI JIAO TONG UNIVERSITY

- SORTING

Standard Algorithms

- Sorting
- Searching & Hashing

Standard Algorithms

- Sorting
- Searching & Hashing
- Strongly connected components

Standard Algorithms

- Sorting
- Searching & Hashing
- Strongly connected components
- Finding shortest paths in graphs

Standard Algorithms

- Sorting
- Searching & Hashing
- Strongly connected components
- Finding shortest paths in graphs
- Edit distances

Standard Algorithms

- Sorting
- Searching & Hashing
- Strongly connected components
- Finding shortest paths in graphs
- Edit distances
- Minimum spanning trees in graphs

Standard Algorithms

- Sorting
- Searching & Hashing
- Strongly connected components
- Finding shortest paths in graphs
- Edit distances
- Minimum spanning trees in graphs
- Matchings in bipartite graphs

Standard Algorithms

- Sorting
- Searching & Hashing
- Strongly connected components
- Finding shortest paths in graphs
- Edit distances
- Minimum spanning trees in graphs
- Matchings in bipartite graphs
- Maximum flows in networks

Data Structure

Data Structure

- Balance trees, red-and-black trees
- Kripke structure, automata

Data Structure

- Balance trees, red-and-black trees
- Kripke structure, automata
- Priority queue
- Disjoint set
- Ordered binary decision diagrams (OBDD)

Data Structure

- Balance trees, red-and-black trees
- Kripke structure, automata
- Priority queue
- Disjoint set
- Ordered binary decision diagrams (OBDD)
- ...

Church-Turing Thesis

Computational Complexity

Church-Turing Thesis

Complexity class

Computational Complexity

SHANGHAI JIAO TONG UNIVERSITY

Church-Turing Thesis

Complexity class

- P, NP, Co-NP, NPI, NP-complete

Computational Complexity

SHANGHAI JIAO TONG UNIVERSITY

Church-Turing Thesis

Complexity class

- P, NP, Co-NP, NPI, NP-complete
- PSPACE

Computational Complexity

Church-Turing Thesis

Complexity class

- P, NP, Co-NP, NPI, NP-complete
- PSPACE
- RP, ZPP

Computational Complexity

Church-Turing Thesis

Complexity class

- P, NP, Co-NP, NPI, NP-complete
- PSPACE
- RP, ZPP

Handling hard problems

Computational Complexity

Church-Turing Thesis

Complexity class

- P, NP, Co-NP, NPI, NP-complete
- PSPACE
- RP, ZPP

Handling hard problems

- Simplex, DPLL(CDCL)(backtracking)
- Approximation,
- local search
- treewidth

The Door of Algorithms Will Open!

Roadmap

SHANGHAI JIAO TONG UNIVERSITY

	算法策略					算法结构		
	分治法	动态规划	贪婪	规约	对偶	图	流	数
基本问题与算法	排序问题、中位数等	最长公共子序列、编辑距离等	最小生成树、哈夫曼编码等	图、树上的常规算法	最大流最小割、最大匹配最小顶点覆盖等	深度搜索、广度搜索、DAG图、最短路径	福特弗格森算法	大数问题、模问题
理论分析方法	大师定理、Akra-Bazzi定理	树宽、时空转化，自顶向下，自底向上	—	难易问题划分	原问题-对偶问题	优先队列、并查集	良结构系统证明方法	概率分析、大数分析
高级问题与算法	快速傅里叶变换	马尔科夫链、序列比对、树宽等	近似算法	复杂性类问题Karp规约、图灵规约	拉格朗日对偶	强连通子图、Bellman-Ford算法、图同构	Dinitz算法	公钥加密、一次一密
工程算法与具体应用	工程快速傅里叶变换算法	动态规划中的空间压缩、马尔科夫链	Boruvka算法、簇聚类算法	DPLL/CDCL算法、不变量生成	神经网络的验证方法	Kosaraju算法、Tarjan算法、形式验证算法	前项流推动算法	Miller-Rabin算法

Guidelines of This Exam

Algorithms in This Lecture

Algorithm Strategies

Algorithms in This Lecture

Algorithm Strategies

- divide and conquer
- dynamic programming
- greedy algorithms
- duality
- reduction

Algorithms in This Lecture

SHANGHAI JIAO TONG UNIVERSITY

Algorithm Strategies

- divide and conquer
- dynamic programming
- greedy algorithms
- duality
- reduction

Specific algorithms

Algorithms in This Lecture

Algorithm Strategies

- divide and conquer
- dynamic programming
- greedy algorithms
- duality
- reduction

Specific algorithms

- algorithm with numbers
- graph algorithms
- network flows

Algorithms in This Lecture

Algorithm Strategies

- divide and conquer
- dynamic programming
- greedy algorithms
- duality
- reduction

Specific algorithms

- algorithm with numbers
- graph algorithms
- network flows

NP problems

Algorithms in This Lecture

Algorithm Strategies

- divide and conquer
- dynamic programming
- greedy algorithms
- duality
- reduction

Specific algorithms

- algorithm with numbers
- graph algorithms
- network flows

NP problems

- NP, Co-NP, NPC
- reduction
- handling NPH problem

First of ALL

Hand in ALL homework!

Languages

The exam is given in Chinese,

Languages

The exam is given in Chinese,

with translation sheet for international students.

Types of the Exam Problems

M1. Show modelling ability, proof ability, and algorithm analysis ability (20')

Types of the Exam Problems

M1. Show modelling ability, proof ability, and algorithm analysis ability (20')

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

Types of the Exam Problems

M1. Show modelling ability, proof ability, and algorithm analysis ability (20')

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

M3. Design algorithms and analysis on numbers, graphs, and flows. (25')

Types of the Exam Problems

M1. Show modelling ability, proof ability, and algorithm analysis ability (20')

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

M3. Design algorithms and analysis on numbers, graphs, and flows. (25')

M4. Prove a NPC problem (15')

Types of the Exam Problems

M1. Show modelling ability, proof ability, and algorithm analysis ability (20’)

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30’)

M3. Design algorithms and analysis on numbers, graphs, and flows. (25’)

M4. Prove a NPC problem (15’)

M5. Cope with NPH problem (10’)

Types of the Exam Problems

M1. Show modelling ability, proof ability, algorithm analysis ability (20')

Types of the Exam Problems

M1. Show modelling ability, proof ability, algorithm analysis ability (20')

- given an problem, try to model it formally.

Types of the Exam Problems

M1. Show modelling ability, proof ability, algorithm analysis ability (20')

- given an problem, try to model it formally.
- proof the correctness of a simple algorithm.

Types of the Exam Problems

M1. Show modelling ability, proof ability, algorithm analysis ability (20')

- given an problem, try to model it formally.
- proof the correctness of a simple algorithm.
- give an analysis to a piece of Pseudo codes.

Types of the Exam Problems

M1. Show modelling ability, proof ability, algorithm analysis ability (20')

- given an problem, try to model it formally.
- proof the correctness of a simple algorithm.
- give an analysis to a piece of Pseudo codes.
- given a linear programming, figure out its duality, and find out the optimization solution.

Types of the Exam Problems

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

Types of the Exam Problems

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

- Divide and conquer

Types of the Exam Problems

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

- Divide and conquer (master theorem)

Types of the Exam Problems

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

- Divide and conquer (master theorem)
- Dynamic programming

Types of the Exam Problems

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

- Divide and conquer (master theorem)
- Dynamic programming (design, border conditions, complexity)

Types of the Exam Problems

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

- Divide and conquer (master theorem)
- Dynamic programming (design, border conditions, complexity)
- Greedy

Types of the Exam Problems

SHANGHAI JIAO TONG UNIVERSITY

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

- Divide and conquer (master theorem)
- Dynamic programming (design, border conditions, complexity)
- Greedy
- Reduction

Types of the Exam Problems

M2. Adopt algorithmic strategies to solve and analyze problems (greedy, D&C, DP, etc.) (30')

- Divide and conquer (master theorem)
- Dynamic programming (design, border conditions, complexity)
- Greedy
- Reduction
- Duality

Types of the Exam Problems

SHANGHAI JIAO TONG UNIVERSITY

M3. Design algorithms and analysis on graphs, numbers and flow (25')

Types of the Exam Problems

M3. Design algorithms and analysis on graphs, numbers and flow (25’)

- DFS, BFS

Types of the Exam Problems

SHANGHAI JIAO TONG UNIVERSITY

M3. Design algorithms and analysis on graphs, numbers and flow (25')

- DFS, BFS
- Shortest path, MST

Types of the Exam Problems

M3. Design algorithms and analysis on graphs, numbers and flow (25')

- DFS, BFS
- Shortest path, MST
- Algorithms on DAG

Types of the Exam Problems

M3. Design algorithms and analysis on graphs, numbers and flow (25')

- DFS, BFS
- Shortest path, MST
- Algorithms on DAG
- Algorithms on numbers (modular)

Types of the Exam Problems

M3. Design algorithms and analysis on graphs, numbers and flow (25’)

- DFS, BFS
- Shortest path, MST
- Algorithms on DAG
- Algorithms on numbers (modular)
- Applications of network flows

Types of the Exam Problems

M4. Prove an NPC problem (15’)

Types of the Exam Problems

M4. Prove an NPC problem (15')

- Prove an NP problem

Types of the Exam Problems

SHANGHAI JIAO TONG UNIVERSITY

M4. Prove an NPC problem (15’)

- Prove an NP problem
- Prove an NPC problem

Types of the Exam Problems

M5. Cope with NPC problem (10’)

Types of the Exam Problems

M5. Cope with NPC problem (10')

- Approximation algorithm

Types of the Exam Problems

M5. Cope with NPC problem (10')

- Approximation algorithm
- Backtracking

Types of the Exam Problems

Shanghai Jiao Tong University

M5. Cope with NPC problem (10')

- Approximation algorithm
- Backtracking
- Local search

Exam This Year

	1	2	3	4	5	6	7	Total
M1: Modeling, proof and analysis (20')	5	5				10		20
M2: Strategies (30')	10		15				5	30
M3: Graph, flow, and number (25')		10					15	25
M4: Prove NPC (15')					15			15
M5: Handle NPH (10')				10				10
Total	15	15	15	10	15	10	20	100