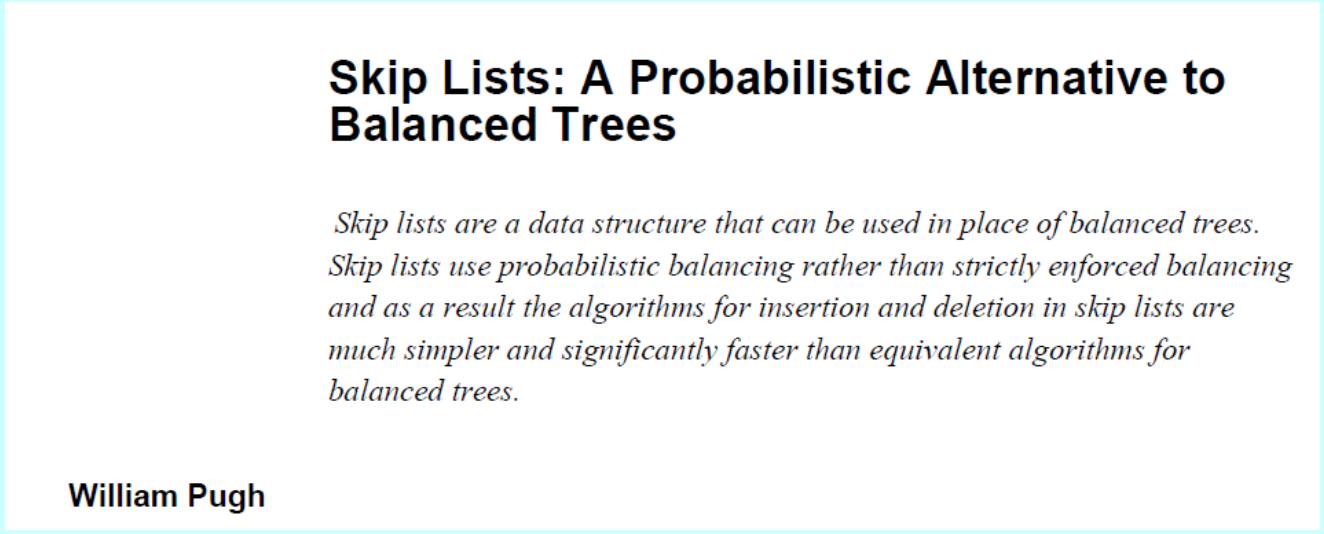
高级数据结构-review

Scope

- 上课ppt/清华教材和讲义（对上课涉及的论文不做要求）
- Project /Lab/Homework 也需要复习
- 考试不会直接涉及源代码填空，但需要理解原理
- 复杂度分析了解即可，数据结构部分不用证明或推导
- 注意，如果编程需要写主要思路，要把主要想法写清楚，如果不记得某个库函数的方法/参数，可用伪代码加注释说明。

1 skiplist

- (a) 散列：循值访问　//基本原理掌握
- (b) 跳转表：结构　//基本数据结构
- (c) 跳转表：算法 // 理解代码，不需要复杂度分析
- (d) 应用：LSM算法

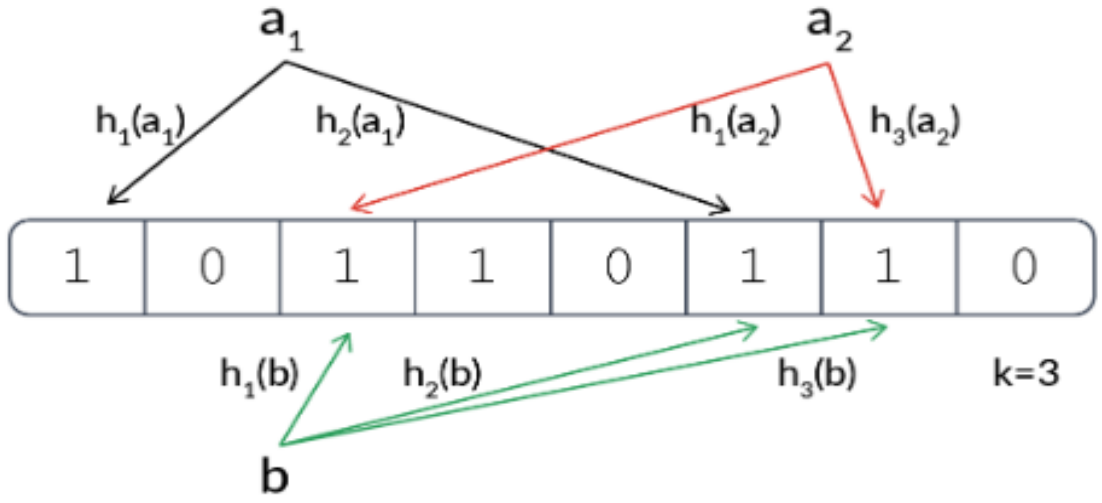


2 Sets & Bloom filters

- (a) Sets　//基本原理
- (b) Bloom filters　//基本原理

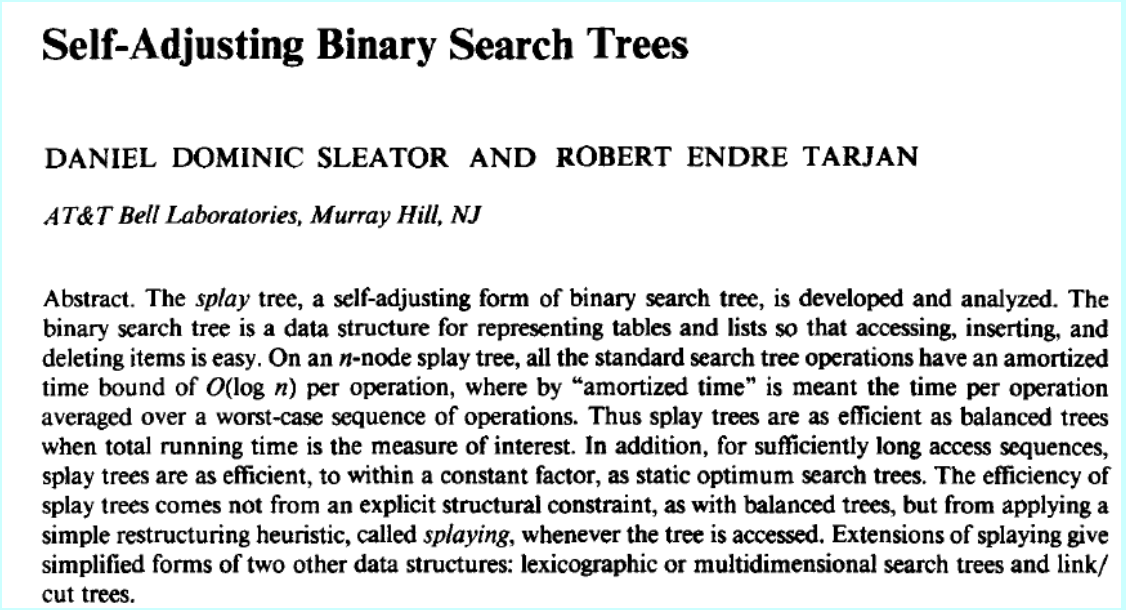
Space/Time Trade-offs in Hash Coding with Allowable Errors

BURTON H. BLOOM
Computer Usage Company, Newton Upper Falls, Mass.



3 Splay Trees

- (a)伸展//基本原理
- (b)插入算法 // 基本概念
- (c) top-down splaying //不作要求



Self-Adjusting Binary Search Trees

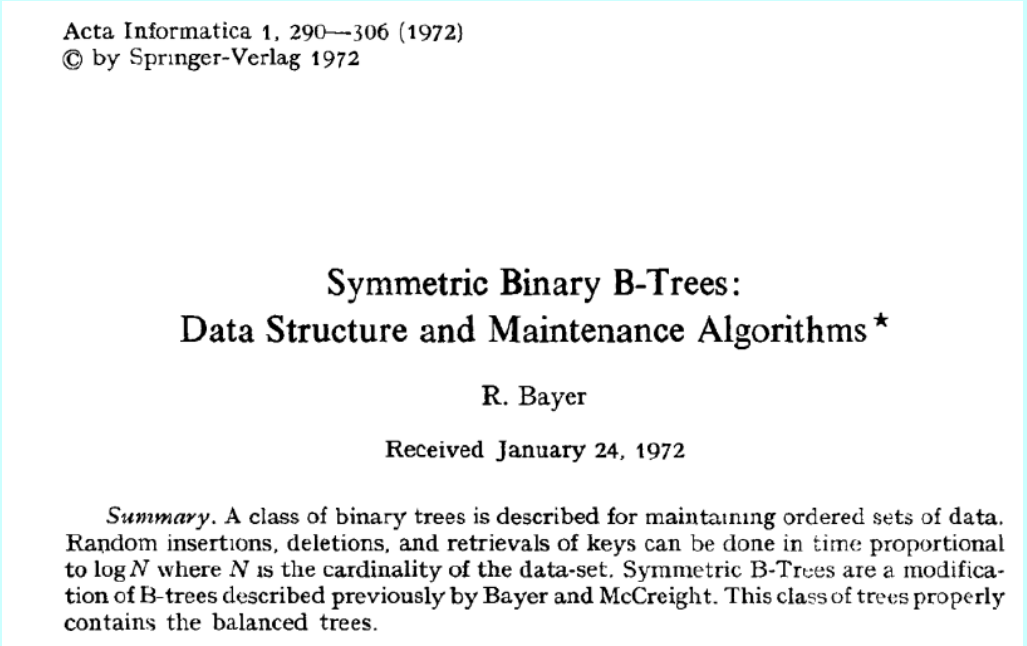
DANIEL DOMINIC SLEATOR AND ROBERT ENDRE TARJAN

AT&T Bell Laboratories, Murray Hill, NJ

Abstract. The *splay* tree, a self-adjusting form of binary search tree, is developed and analyzed. The binary search tree is a data structure for representing tables and lists so that accessing, inserting, and deleting items is easy. On an n-node splay tree, all the standard search tree operations have an amortized time bound of $O(\log n)$ per operation, where by "amortized time" is meant the time per operation averaged over a worst-case sequence of operations. Thus splay trees are as efficient as balanced trees when total running time is the measure of interest. In addition, for sufficiently long access sequences, splay trees are as efficient, to within a constant factor, as static optimum search trees. The efficiency of splay trees comes not from an explicit structural constraint, as with balanced trees, but from applying a simple restructuring heuristic, called *splaying*, whenever the tree is accessed. Extensions of splaying give simplified forms of two other data structures: lexicographic or multidimensional search trees and link/cut trees.

4 Red-Black Tree

- (a)基本原理 //掌握
- (b)插入算法 // 基本概念
- (c) 删除算法 //不作要求



Acta Informatica 1, 290—306 (1972)

Symmetric Binary B-Trees:
Data Structure and Maintenance Algorithms*

R. Bayer

Received January 24, 1972

Summary. A class of binary trees is described for maintaining ordered sets of data. Random insertions, deletions, and retrievals of keys can be done in time proportional to $\log N$ where N is the cardinality of the data-set. Symmetric B-Trees are a modification of B-trees described previously by Bayer and McCreight. This class of trees properly contains the balanced trees.

5 KD Tree

- (a)基本原理 //掌握
- (b)构造算法 // 基本概念
- (c) 查找算法 //基本概念

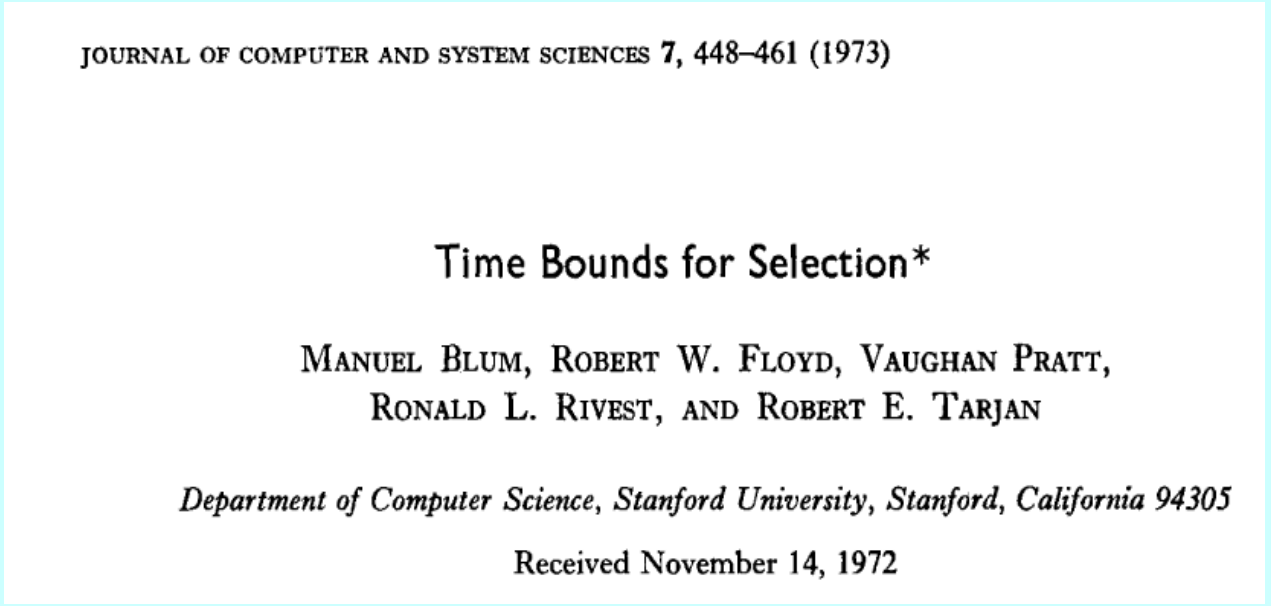

1975 ACM Student Award
Paper: Second Place

Multidimensional Binary Search Trees Used for Associative Searching

Jon Louis Bentley
Stanford University

6 众数和中位数

- (a)减治法基本原理 //掌握
- (b)中位数算法 // 基本概念
- (c) BFPRT算法 //基本概念



JOURNAL OF COMPUTER AND SYSTEM SCIENCES 7, 448–461 (1973)

Time Bounds for Selection*

MANUEL BLUM, ROBERT W. FLOYD, VAUGHAN PRATT,
RONALD L. RIVEST, AND ROBERT E. TARJAN

Department of Computer Science, Stanford University, Stanford, California 94305

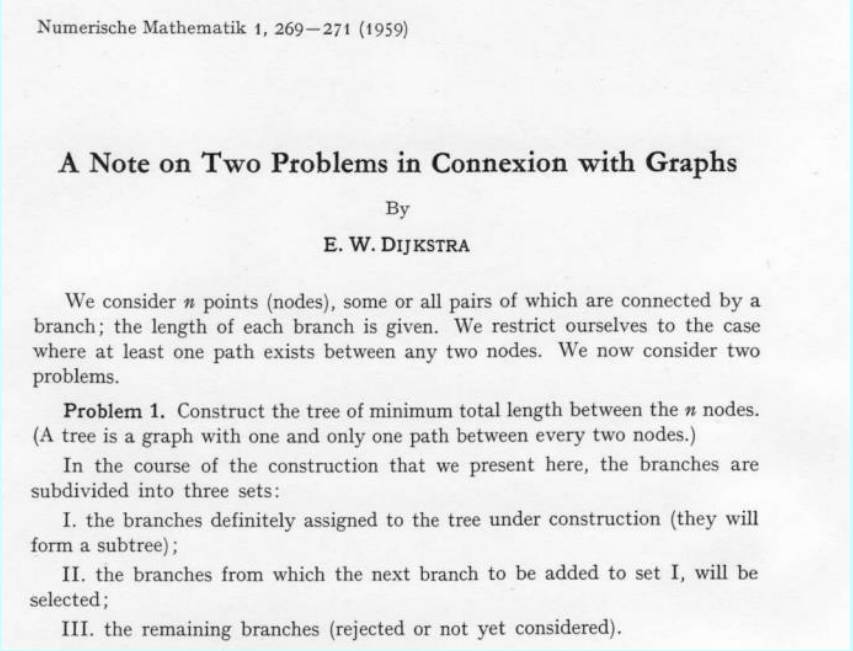
Received November 14, 1972

7 拓扑排序

- (a)基本原理 //掌握
- (b) 算法A：顺序输出零入度顶点// 基本概念
- (c) 算法B：逆序输出零出度顶点// 基本概念

8 最短路径算法

- (a)优先级搜索 //掌握
- (b)最小支撑树 // 基本概念
- (c) Dijkstra算法 算法 // 熟练掌握
- （d）路径规划讲义（了解）



Numerische Mathematik 1, 269—271 (1959)

A Note on Two Problems in Connexion with Graphs

By

E. W. DIJKSTRA

We consider n points (nodes), some or all pairs of which are connected by a branch; the length of each branch is given. We restrict ourselves to the case where at least one path exists between any two nodes. We now consider two problems.

Problem 1. Construct the tree of minimum total length between the n nodes. (A tree is a graph with one and only one path between every two nodes.)

In the course of the construction that we present here, the branches are subdivided into three sets:

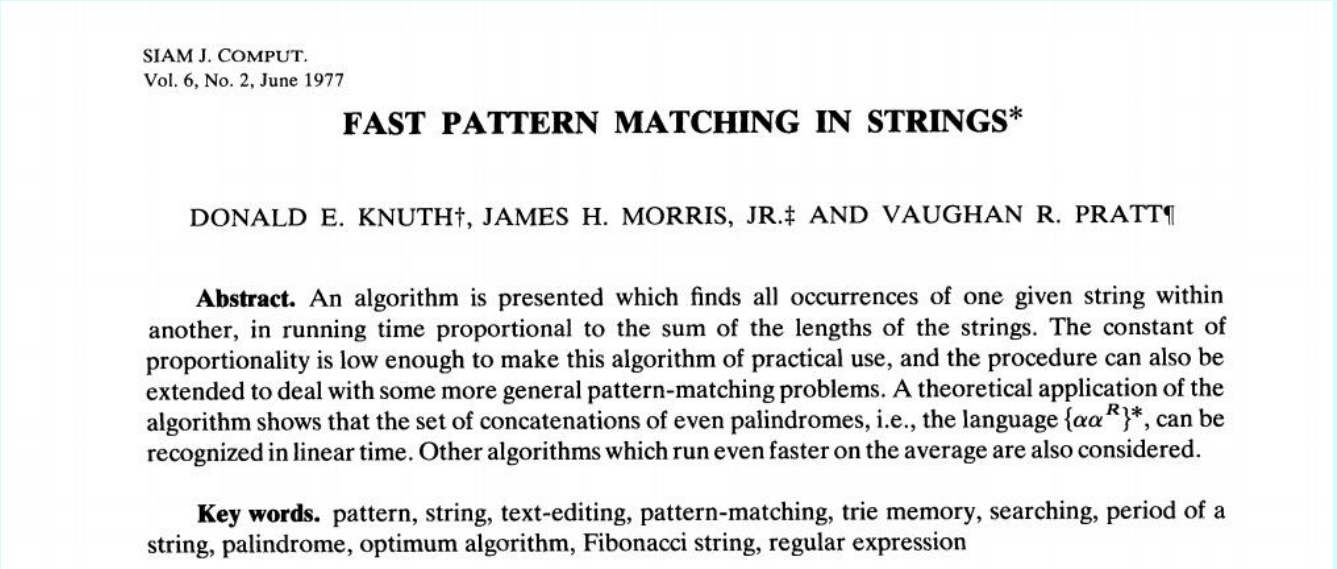
I. the branches definitely assigned to the tree under construction (they will form a subtree);

II. the branches from which the next branch to be added to set I, will be selected;

III. the remaining branches (rejected or not yet considered).

9 KMP算法

- (a)串匹配//掌握
- (b)KMP算法// 熟练掌握
- (c) KMP算法：分摊分析// 不做要求



SIAM J. COMPUT.
Vol. 6, No. 2, June 1977

FAST PATTERN MATCHING IN STRINGS*

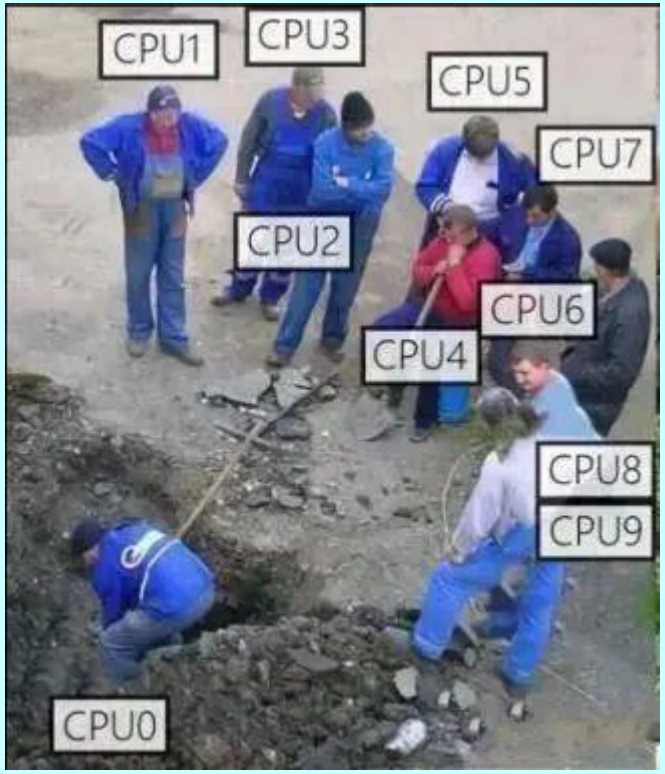
DONALD E. KNUTH†, JAMES H. MORRIS, JR.‡ AND VAUGHAN R. PRATT¶

Abstract. An algorithm is presented which finds all occurrences of one given string within another, in running time proportional to the sum of the lengths of the strings. The constant of proportionality is low enough to make this algorithm of practical use, and the procedure can also be extended to deal with some more general pattern-matching problems. A theoretical application of the algorithm shows that the set of concatenations of even palindromes, i.e., the language $\{\alpha\alpha^R\}^*$, can be recognized in linear time. Other algorithms which run even faster on the average are also considered.

Key words. pattern, string, text-editing, pattern-matching, trie memory, searching, period of a string, palindrome, optimum algorithm, Fibonacci string, regular expression

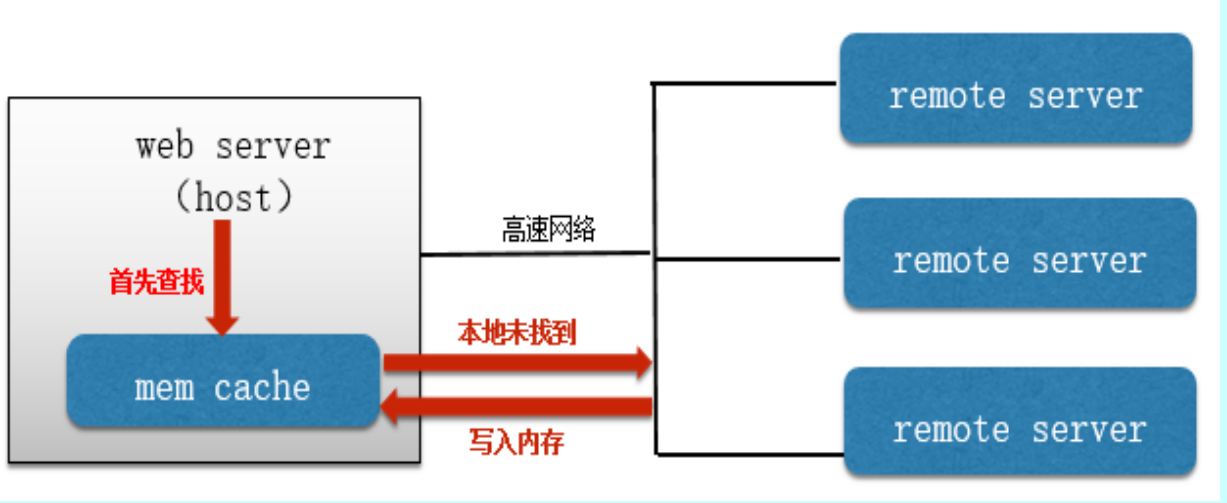
10 并行编程

- (a)串线程同步-互斥锁//掌握
- (b)并发List// 理解
- (c)并发散列表 //理解
- (d) 死锁//理解



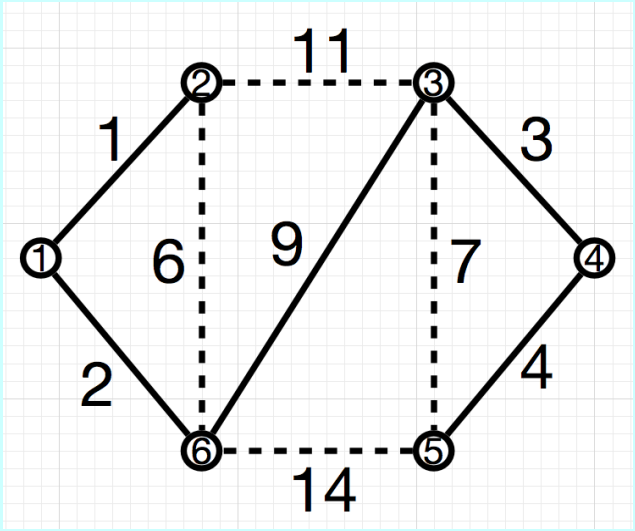
11 并行Cuckoo 哈希

- (a) Cuckoo hash基本操作//掌握
- (b)基于回溯的实现// 理解



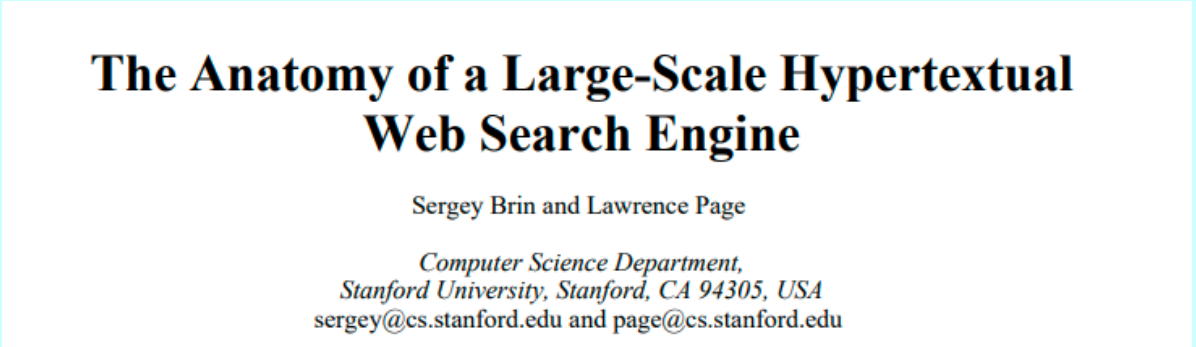
12 并行MST

- (a)并行Kruskal算法//理解
- (b)基于顶点的并行最小生成树算法// 了解



13 Pagerank

- (a) PageRank算法//理解
- (b)基于图结构的实现// 了解



The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

The PageRank Citation Ranking: Bringing Order to the Web

考试安排

- **考试时间**：6月9日 （周五）10:30-12:30
- **考试形式**：**闭卷.**
- **特殊处理**： 大四毕业生、申请缓考、重考（补、缓考成绩是P/F；重考是具体分数）
- **成绩**： 期末考试60%+Labs/Project 40%

祝大家考出好成绩！