Literature Notes on Homeworks

Homework 1

1: **Orientability**

The notation of c-orientability is very closely related to the notion of bounded arboricity, and the definitions are basically identical. The papers [1] and [2] considers the implicit representation of graphs of bounded arboricity, where [1] considers the problem for static graphs and [2] considers the dynamic version of the problem where edges can be inserted and deleted.

2: **Array initialization**

Section III.8.1 of [15] contains a description of how a bitvector can be intitialized in worst case constant time.

3: **Sink**

Exercise 23.1-6 in [6].

4: **Scorpion**

Exercise 3 of homework 2 in [13], from which also the following paragraph is taken.

"This is a counterexample to an earlier version of the famous *Anderaa-Rosenberg conjecture*, which stated that any nontrivial graph property that is invariant under graph isomorphism requires $\Omega(n^2)$ probes of the adjacency matrix to test. Anderaa disproved this version in 1975 using a different counterexample (see [18]), but conjectured that it held for *monotone* properties (those that cannot change from false to true when edges are deleted). This was later verified by Rivest and Vuillemin [18]."

Homework 2

1: Extendible Arrays

The array doubling idea is folklore. A generalization of this idea has been described by Overmars in [16] for the design of dynamic data structures.

2: Binomial Heaps

A proof that binomial heaps support merging in amortized constant time was given in [10].

3: Dijkstra's Algorithm

Folklore.

4: Prim's Algorithm

Exercise 2 of homework 4 in [13]. Prim's algorithm for computing a minimum spanning tree is, *e.g.*, described in Section 24.2 of [6].

5: The Stable Marriage Problem

Exercise 1 of homework 6 in [13]. Gale and Shapley were the first to investigate the stable marriage problem and gave an $O(n^2)$ solution in [8].

Homework 3

1: Minimum Spanning Trees

A linear time algorithm for finding a minimum spanning tree for planar graph was first given in [5]. The $O(m \log^* n)$ time algorithm for finding a minimum spanning tree in a general graph was described in [7]—the paper introducing Fibonacci heaps. The current best dertministic minimum spanning tree algorithms use time $O(m\alpha(m,n))$, where α is an inverse of Ackerman's function [4, 17]. A randomized linear time algorithm has been given in [12].

2: Augmenting Paths of Maximum Bottleneck Capacity

Exercise 1(b) of homework 5 in [13].

3: Ghostbusters

Problem 35-4 in [6]. The ghostbusters problem can be solved in time $O(n \log n)$ by a divide-and-conquer algorithm, by applying the Ham-Sandwich Theorem and using a linear time algorithm for finding ham-sandwich cuts from [14].

4: Self Adjusting Trees

The class of self-adjusting trees considered in the exercise are often denoted *splay-trees*, and were described in [20].

Homework 4

1: Disk intersection

Exercise 35.2-6 in [6].

2: Edit distance

Computing the edit distance of two strings is one of the classical applications of dynamic programming. Hirshberg developed in [11] a general technique that, among others, allows the computation of the edit sequence in linear space.

3: Minimal perfect hash functions

An asymptotic space optimal perfect hash function implementation was first described in [19]. The exercise follows the basic approach of [19].

Homework 5

Homework 5 covers results from three papers. The application of level linked (a,b)–trees for finger searches was given in [9]. The use of efficient merging of search trees for finding regularities in strings was explored in [3], where the regularities found are *pairs with bounded gap* which are an generalization of tandem repeats. The approach of extracting all tandem repeats from the set of all branching tandem repeats was taken in [21]. The algorithm in [21] for computing all tandem repeats is simpler than the approach taken in the exercise, since [21] does not generate the leaf-lists explicitly but only applies two depth first traversals of the suffix tree.

References

[1] Srinivasa R. Arikati, Anil Maheshwari, and Christos D. Zaroliagis. Efficient computation of implicit representations of sparse graphs. *Discrete Applied Mathematics*, 78:1–16, 1997.

[2] Gerth Stølting Brodal and Rolf Fagerberg. Dynamic Representations of Sparse Graphs. In *Proc. 6th International Workshop on Algorithms and Data Structures*, volume 1663 of *Lecture Notes in Computer Science*, 342–351. Springer Verlag, Berlin, 1999.

[3] G. S. Brodal, R. B. Lyngsø, C. N. S. Pedersen, and J. Stoye. Finding maximal pairs with bounded gap. In *Proceedings of the 10th Annual Symposium on Combinatorial Pattern Matching (CPM)*, volume 1645 of *Lecture Notes in Computer Science*, pages 134–149, 1999.

[4] B. Chazelle. A Minimum Spanning Tree Algorithm with Inverse-Ackermann Type Complexity. NECI Technical Report 99-099, July 1999.

[5] D. Cheriton and R. E. Tarjan. Finding minimum spanning trees. *SIAM Journal of Computing*, 5:724-742, 1976.

[6] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms.* MIT Press, 1990.

[7] Michael L. Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, 1987.

[8] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9-14, 1962.

[9] Scott Huddleston and Kurt Mehlhorn. A new data structure for representing sorted lists. *Acta Informatica*, 17:157–184, 1982.

[10] C. M. Khoong and H. W. Leong. Double-ended binomial queues. In *Algorithms and Computation*, volume 762 of *Lecture Notes in Computer Science*, 128–137. Springer Verlag, Berlin, 1993.

[11] D. S. Hirschberg. Algorithms for the longest common subsequence problem. *Journal of the ACM*, 24:664-675, 1977.

[12] D. R. Karger, P. N. Klein, and R. E. Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM*, 42:321-328, 1995.

[13] Dexter C. Kozen. *The Design and Analysis of Algorithms.* Springer-Verlag, 1992.

[14] C.-Y. Lo, J. Matoušek, and W. L. Steiger. Algorithms for ham-sandwich cuts. *Discrete Computational Geometry*, 11:433–452, 1994.

[15] Kurt Mehlhorn. *Data Structures and Algorithms 1: Sorting and Searching.* Springer Verlag, Berlin, 1984.

[16] Mark H. Overmars. *The Design of Dynamic Data Structures*, volume 156 of *Lecture Notes in Computer Science.* Springer Verlag, Berlin, 1983.

[17] S. Pettie. Finding Minimum Spanning Trees in $O(m\alpha(m,n))$ Time. UTCS Technical Report TR99-23.

[18] R. L. Rivest and J. Vuillemin. On recognizing graph properties from adjacency matrices. *Theoretical Computer Science*, 3:371-384, 1976/77.

[19] J. P. Schmidt and A. Siegel. The spatial complexity of oblivious k-probe hash functions. *SIAM Journal on Computing*, 19(5):775–786, 1990.

[20] Daniel Dominic Sleator and Robert Endre Tarjan. Self-adjusting binary search trees. *Journal of the ACM*, 32(3):652–686, 1985.

[21] J. Stoye and D. Gusfield. Simple and flexible detection of contiguous repeats using a suffix tree. In *Proceedings of the 9th Annual Symposium on Combinatorial Pattern Matching (CPM)*, volume 1448 of *Lecture Notes in Computer Science*, pages 140–152, 1998.