

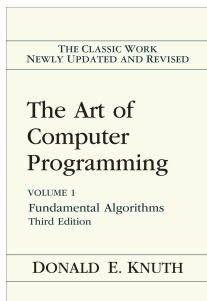
1-5 数据与数据结构 (II)

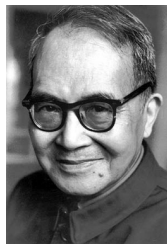
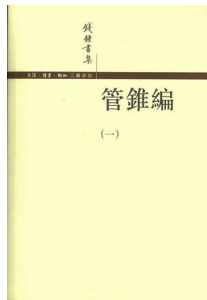
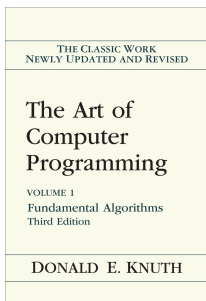
魏恒峰

hfwei@nju.edu.cn

2017 年 11 月 27 日

Chapter 2—Information Structures	232
2.1. Introduction	232
2.2. Linear Lists	238
2.2.1. Stacks, Queues, and Deques	238
2.2.2. Sequential Allocation	244
2.2.3. Linked Allocation	254
2.2.4. Circular Lists	273
2.2.5. Doubly Linked Lists	280
2.2.6. Arrays and Orthogonal Lists	298
2.3. Trees	308
2.3.1. Traversing Binary Trees	318
2.3.2. Binary Tree Representation of Trees	334
2.3.3. Other Representations of Trees	348
2.3.4. Basic Mathematical Properties of Trees	362
2.3.4.1. Free trees	363
2.3.4.2. Oriented trees	372
*2.3.4.3. The “infinity lemma”	382
*2.3.4.4. Enumeration of trees	386
2.3.4.5. Path length	399
*2.3.4.6. History and bibliography	406
2.3.5. Lists and Garbage Collection	408
2.4. Multilinked Structures	424
2.5. Dynamic Storage Allocation	435
2.6. History and Bibliography	457

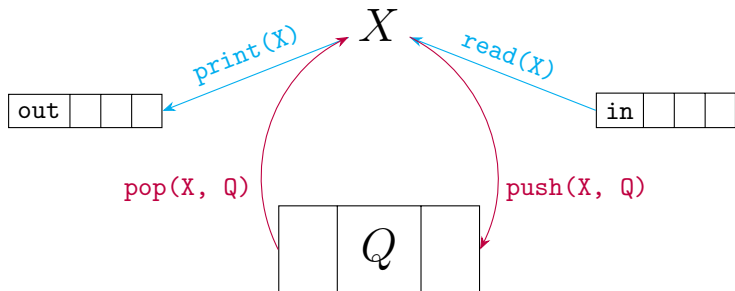




Stackable/Queueable Permutations

DH 2.14: Queueable Permutations

$$\text{out} = (a_1, \dots, a_n) \xleftarrow[\substack{Q=\emptyset \\ X=0}]{Q=\emptyset} \text{in} = (1, \dots, n)$$

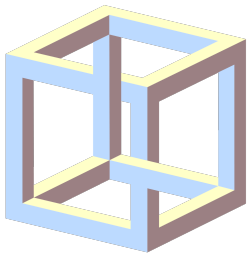


DH 2.14: Queueable Permutations

(a) Show that the permutations given in Exercise 2.12(b) are queueable.

(i) $(3, 1, 2)$

(ii) $(4, 5, 3, 7, 2, 1, 6)$



DH 2.14: Queueable Permutations

(b) Prove that every permutation can be obtained by a queue.

Alg here.

DH 2.14: Queueable Permutations

(c) Prove that every permutation can be obtained by **two stacks**.

DH 2.15: Algorithm for Queueable Permutations

Extend the algorithm you were asked to design in Exercise 2.13, so that if the given permutation cannot be obtained by a stack, the algorithm will print the series of operations on two stacks that will generate it.

