§1. 数据结构绪论.

Program Design = Data Struct + Algo.

数据结构：相互间存在一种或多种特定关系的数据元素的集合.

逻辑结构. 集合. 线. 树. 图.

物理结构. 顺序. 链式.

数据类型. atom / struct.

ADT (Abstract Data Type).

§2. 算法 (Algorithm).

算法：解决特定问题求解步骤的描述.

用指令的有限序列实现.

Input. Output.

Finite. Definite. Effective.

事后统计 / 事前分析. 时空复杂度. 最坏与平均.

↳ Asymtopic O.

§3. 线性表 (List).

ADT List

Data

DataType {a1, ——, an}.

Operation.

InitList. IsEmpty. [i]. (get). Search(e).

ins(i, e). del(i, p). len().

顺序：[i] O(1) (随机存储结构).

ins del O(n).

链式：[i] O(n)

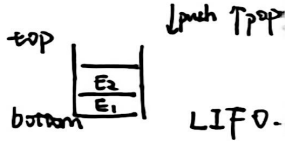
ins del O(1).

(单链表). (静态、循环、双向).

§4. 栈与队列. (Stack & Queue).

ADT Stack.

top. push. pop. length. empty.



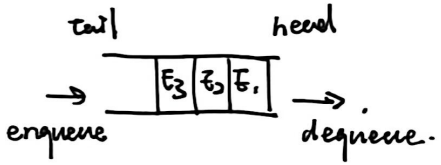
LIFO.

顺序、链式.

ADT Queue.

head. enqueue. dequene.

length. empty.



顺序(循环). 链式.

§5. 串 (string).

Alphabet Σ = {a, b, c}. $w \in \Sigma^*$. 空串 ε.

Equal: 1. $|u| = |v|$

2. $u_i = v_i$.

Lexicographic Order: (total order $< \Sigma$).

let $k = \min(|u|, |v|)$.

1. $\forall$ is the smallest $u(i) \neq v(i)$: $u(i) < v(i) \Rightarrow u < v$
2. $\forall$ doesn't exists: $|u| < |v| \Rightarrow u < v$.

ADT String.

Operation

copy, len. cmp. concat.

substring (S. pos. len).

index (S. T. cpos)). 模式匹配.

replace (S. T. V). insert (S. pos. T).

delete (S. pos. len).

朴素模式匹配: O(nm) worst case.

KMP: O(n+m). space O(m).

next [j] : 最长相同前后缀长度.

改良 KMP: nextval.

§6. 树. (Tree).

①无环 ②连通图 / N个点. ③N-1条边. 的连通图.

递归定义.

terms:

root. child. parent. leaf. siblings.

ancestors. descendants. degree.

表示: children / FirstChild - NextSibling

二叉树: ≤2 棵子树, 有左右.

斜、满、完全.

性质. (2^{i-1} / 2^k-1 / $n_0=n_2+1$)/($\lfloor \log_2 n \rfloor +1$ / $\lfloor \frac{i}{2} \rfloor$, $2i$, $2i+1$).

存储: 顺序 (2k, 2k+1), 链式. 遍历: 前. 中. 后. 层.

Huffman 树 (complete)

§7. 图 (Graph).

G(V, E).

terms:

(un)directed. complete. ($\frac{n(n-1)}{2}$ / $n(n-1)$).

sparse / dense. weight. subgraph. simple.

adjacent / incident. (in/out)degree.

path. cycle. path-len.

connected. Connected Component. (strong)

Spanning Tree.

存储: (无指 O(n+e))

邻接矩阵. 邻接表. 十字链表. 邻接多重表.

边集数组.

遍历:

DFS. BFS.

(recurse) (queue)

Minimum Spanning Tree.

Prim.

(S. V-S). 初始 S 一个点.

利用割性质逐步扩大 S.

Kruskal.

从小到大排 e. 逐步连通 n 个顶点

(割性质: 最小割边在 MST 上. (反证)).

The Shortest Path.

Dijkstra. (单源)

1. S. T. dis(s)=0, dis(i)=∞.
2. while True:
3. e ← 当前 dis 最小的点 (可能再被 relax)
4. S+=e. relax(e).

Floyd (全源).

f_{ij}^0 = 0/w/+∞.

for k, i, j

$f_{ij}^k = \min(f_{ij}^{k-1}, f_{ik}^{k-1} + f_{kj}^{k-1})$.

Topological Sort.

AOV网. Activity On Vertex.

1. while True:
2. v ← an vertex that has degree 0. (maintain a set.)
3. print v.
4. remove v & all (ui, v).

critical path.

AOE (Activity on Edge).

// TODO. (感觉略过. 跳了).

§8. Search.

Search Table. Key. Primary Key / Secondary Key.

Searching. Static / Dynamic search Table.

1. Sequential.

2.1. Binary Search. (in order). $mid = \frac{l+r}{2}$.

2.2 Interpolation Search. $mid = l + k(r-l)$.

(in order. uniform) $k = \frac{key - arr[l]}{arr[r] - arr[l]}$

$O(\log \log n)$ in the best case.

2.3. Fibonacci Search. // TODO.

$O(\log n)$ $O(\sqrt{n})$

3. Index: Dense. Block. Inverted.

4. BST. Balance: AVL. RBT.

Search. Insert. Delete:

- 0~1 child: OK.
- 2 child: 左最大 / 右最小 → 1个 0~1 child. 换位.

5. MST. B. B+.

6. Hash Table.

index = f(key)

f: Hash function. 通常使用 mod.

Collision.

开放定址: Linear / Quadratic Probing.

Rehashing.

拉链. 公共溢出.