Hwk6.

17.2-2 核算法:

	实际代价	摊还代价
i为2的幂	2^j	3
其它	1	3

下证 该数据结构中的信用总为非负.

考虑 $c_{2^{k-1}+1} \dots c_{2^k}$ $k \in \mathbb{Z}$

此过程中产生的信用 $= 2 \cdot 2^{k-1}+1-2^k = 1$

而 $c_{2^{k-1}+1} \dots c_{2^k-1}$ 的过程中显然信用单调增.

∴ 信用总非负.

17.3-5. 定义势能函数为计数器中1的个数.

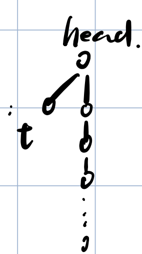
$$\sum_{i=1}^{n} c_i = \sum_{i=1}^{n} \hat{c}_i - \Phi(D_n) + \Phi(D_0)$$

$$= 2n - b_n + b = 2n - b_n + \Omega(n) = O(n)$$

19.4-1 当斐堆中已有一条长为k的单链表, 进行如下操作:

① 插入3个数, 其中两个数比单链表小.

② extractmin. 合并为一个度为2的树. 形如:



③ 将t decrease-key 至比head小.

④ extractmin 则构造出长为k+1的单链表.

初始 insert 一个较大的数, 重复上述方法 n-1 次, 可满足题意

19-2. a. 1. 易知 $B_{k+1}=2B_k$ 且 $B_0=1$

$\therefore B_k=2^k$

2. 设 B_k 的高度为 H_k

$H_k=H_{k-1}+1$ $H_0=0$

$\therefore H_k=k$

3. 设 B_k 中深度为 i 的结点数为 t.

$(D_k)_0=1$ $(D_k)_i=t$

易知 $(D_k)_i=(D_{k-1})_i+(D_{k-1})_{i-1}$

假如对 B_{k-1}, $(D_{k-1})_i=C_{k-1}^i$

对 B_1 $(D_1)_0=1$ $(D_1)_1=1$ 满足.

对 $(D_k)_i=(D_{k-1})_i+(D_{k-1})_{i-1}=C_{k-1}^i+C_{k-1}^{i-1}=C_k^i$ 满足.

$\therefore$ 原命题成立.

4. 设 B_k 的根的度数为 R_k

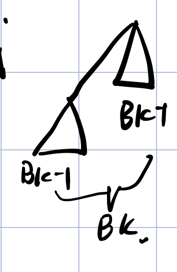
$R_k=R_{k-1}+1$ $R_0=0$ $\therefore R_k=k$

对二项树的任意结点及其子结点，都构成二项树，

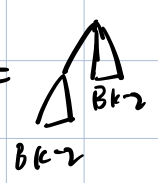
而根结点对应的是最大的二项树

$\therefore$ 根结点的度数最大

易知任意一个二项树可写成如下形式



又 $B_{k-1}=$



$\therefore$ 可以递归分解为题中的形式.

b. 设 n 对应的二进制表示中从右往左第k位为1、

则 H 包含二项树 B_{k-1}

又 n 最多有 ⌊lg n⌋+1 位

∴ H 中最多有 ⌊lg n⌋+1 棵二项树.

c. 对于个 node x 定义 x.s x.p x.r x.d x.v

分别代表子结点. 父结点. 兄弟结点. 度数, value 值

指针指向的值不存在时为 NIL

二项堆由二项树的根节点链表组成.

其它操作与斐波堆大致相同; 除了 Decrease-key, Insert, extract-min

对于 Insert 和 extract-min: 需要马上合并堆中相同度数的二项树.

对于 Decrease-key: 这里需要将调整 key 后的 node 不断向上翻转直到其父节点的值更小.

d. 相似之处在于都根据根节点的度数合并.

不同在于斐波堆根节点度数相同的时候不一定相同.

本题中的斐波堆中全为二项树, 而二项树的度为 lg k. (k 为二项树结点数)

所以二项堆的最大度为 ⌊lg n⌋

e. 其它操作和斐波堆最坏操作时间相同,

对于 Insert $O(\log n)$

合并 $O(\log n)$