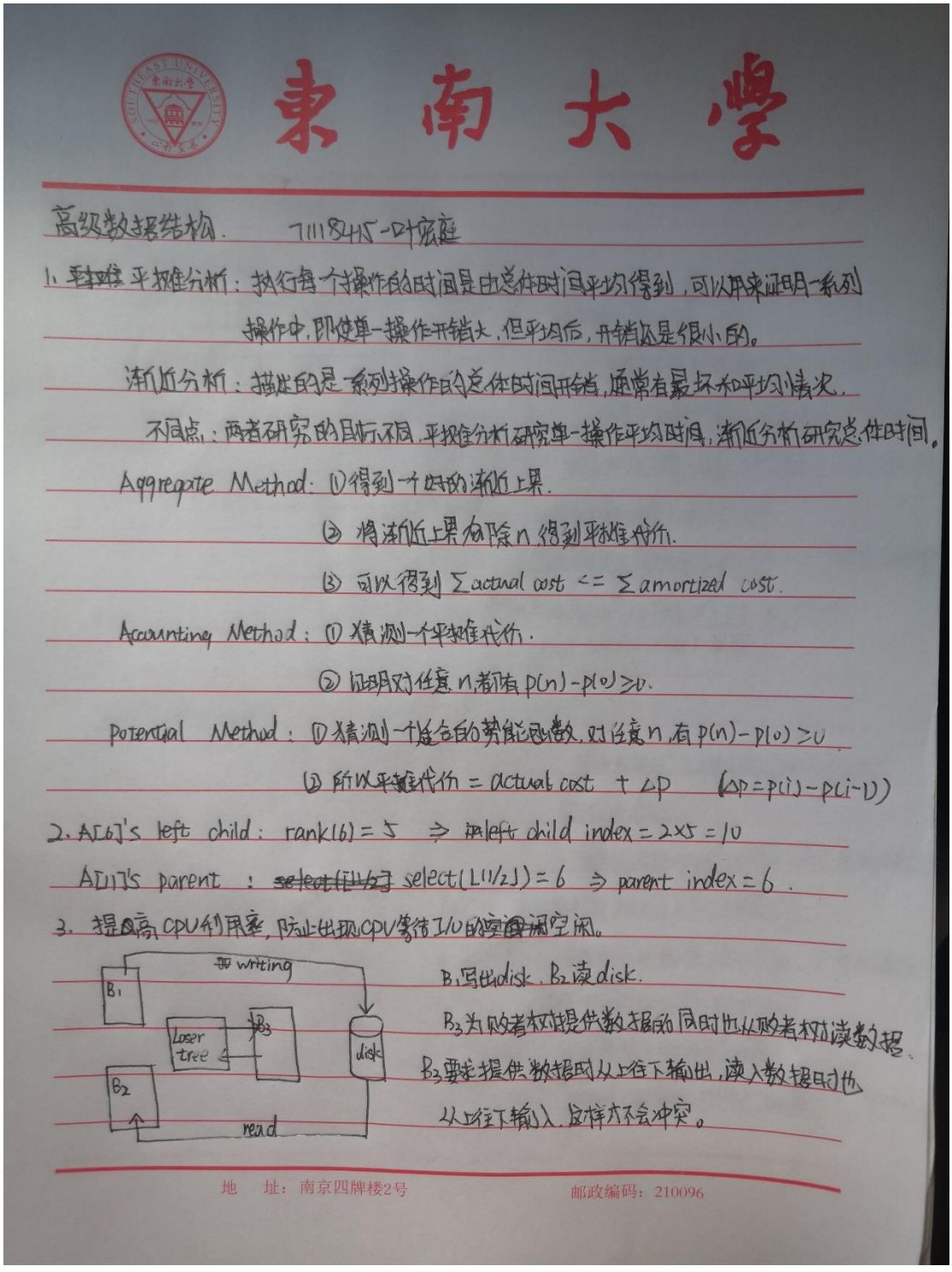
高级数据结构考试

71118415 叶宏庭



东南大学

高级数据结构. 71118415-叶宏庭

1. 平摊分析：执行每一个操作的时间是由总体时间平均得到，可以用来证明一系列操作中，即使单一操作开销大，但平均后，开销还是很小的。

渐近分析：描述的是一系列操作的总体时间开销，通常有最坏和平均情况。

不同点：两者研究的目标不同，平摊分析研究单一操作平均时间，渐近分析研究总体时间。

Aggregate Method: ①得到一个好的渐近上界.

②将渐近上界除n，得到平摊代价.

③可以得到 $\sum$ actual cost <= $\sum$ amortized cost.

Acounting Method: ①猜测一个平摊代价.

②证明对任意n，都有 $p(n)-p(0)\geq 0$.

Potential Method: ①猜测一个适合的势能函数，对任意n，有 $p(n)-p(0)\geq 0$.

②所以平摊代价 = actual cost + Δp ($\Delta p = p(i)-p(i-1)$)

2. A[6]'s left child: rank(6)=5 ⇒ left child index = 2×5 = 10

A[11]'s parent: ~~select(11/2)~~ select(⌊11/2⌋)=6 ⇒ parent index = 6.

3. 提高CPU利用率，防止出现CPU等待I/O的空闲。

B_1写出disk，B_2读disk.

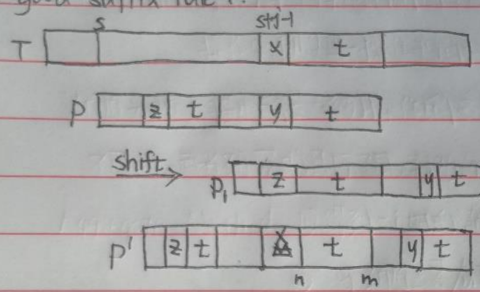
B_3为的者树提供数据的同时也从的者树读数据。

B_3要求提供数据时从上往下输出，读入数据时也从上往下输入，这样不会冲突。

地 址：南京四牌楼2号 邮政编码：210096

高级数据结构　71118415-叶宏庭

4. good suffix rule 1.



图中 P_1 展示了正确右移，下面证明。

证明：反证法：

假设存在 p' 如图所示，p' 位置比 P_1 靠左。

如果 p' 正确匹配，那么 p' 与 T 相对应的这一段一定正确匹配 (p'[n:m] = t)

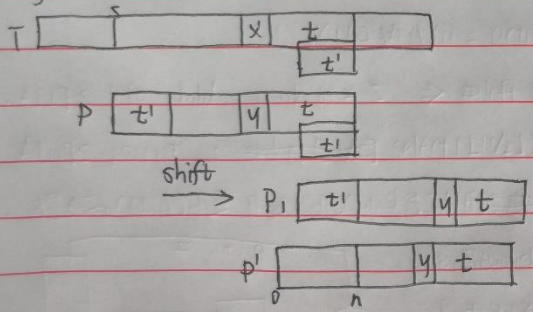
这与我们的正确右移产生矛盾。

正确右移说了，要找到 index 最大的 t，并移到该位置。

所以 p' 中 p'[n:m] 必不能等于 t，这产生矛盾。

所以 good suffix rule 1 正确。

good suffix rule 2.



图中 P_1 展示了正确右移，证明如下。

证明：反证法：

首先，对于 good suffix rule 2 是 P 中不存在 t

且 t' 是 P 的最大好后缀。

若存在 p' 如图，在 P_1 左边。p' 若正确匹配，则 len(p'[0:n]) > len(t')

这与 t' 是最大好后缀矛盾。

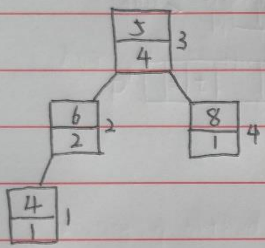
所以 good suffix rule 2 正确。

高级数据结构 71118415-叶宏庭

5. 采用二叉搜索树的原形。(ele并不遵循二叉搜索树，搜索的key本该是index，但我们不能存储index，我们存储size可以更好效率)。

ele	数值
size	子树大小

利用size即可获得当前任意node的index。原理同order statistic trees.



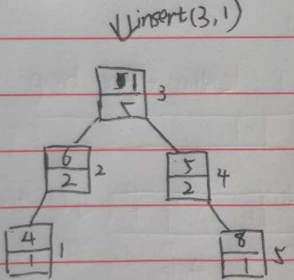
(1) search(int idx)

if(idx == leftchild.size+1). return root.

if(idx < leftchild.size+1). 以leftchild为root. search(idx).

if(idx > leftchild.size+1) 以rightchild为root. search(idx-leftchild.size-1)

时间复杂度 O(logn).



(2) insert(int idx, T ele)

if(idx == leftchild.size+1). 插入到root位置，原root向右下方旋转.

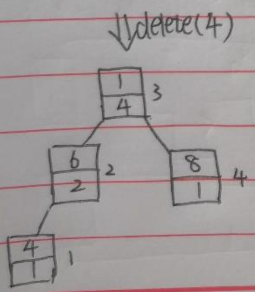
if(idx < leftchild.size+1). 以leftchild为root. insert(idx, ele).

if(idx > leftchild.size+1). 以rightchild为root. insert(idx-leftchild.size-1, ele).

向上更新size.

例: insert(3, 1).

时间复杂度 O(logn)



(3) delete(int idx)

先search(idx)找到指定结点.

再删除结点，用其右子孩子代替他，并且向上更新size.

例: delete(4).

时间复杂度 O(logn)

备注: index = ~~leftchild.size~~ leftchild.size +1 + parent的回溯影响.