DSA Final

n+e 抄录

2016.1

1 判断 (2'×20)

1. 若 AVL 树插入元素的过程中发生了旋转操作，则树高必不变。
2. 若红黑树插入一个元素后，黑高度增加，则双红修正过程中没有拓扑结构变换，只有重染色操作。
3. 若 KMP 算法不使用改进版的 next 表，最坏情况下时间复杂度可能达到 $O(mn)$。
4. 在 BST 中删除两个节点 (7B3)，则无论先删除哪个节点，最终 BST 的拓扑结构均相同。
5. 完全二叉堆删除元素在最坏情况下时间复杂度为 $O(\log n)$，但平均情况下仅为 $O(1)$。
6. 在任何情况下，伸展树总能保持每次操作 $O(\log n)$ 的平均复杂度。
7. 对于左式堆 A 和 B，合并后所得二叉堆的右侧链元素一定来自 A 和 B 的右侧链。
8. 如果元素理想随机，那么对二叉搜索树做平衡化处理，对改进其渐进时间复杂度并没有什么卵用。
9. 采用双向平方试探策略时，将散列表长度取作素数 $M = 4k + 3$，可以极大地降低查找链前 M 个位置冲突的概率，但仍不能杜绝。
10. 在使用 Heapify 批量建堆的过程中，改变同层节点的下滤次序对算法的正确性和时间效率都无影响。
11. 在 kd-search 中，查找区间 R 与任一节点的 4 个孙节点 (假设存在) 对应区域最多有 2 个相交。
12. 在 n 个节点的跳转表中，塔高的平均值为 $O(\log n)$。
13. 既然可以在 $O(n)$ 时间内找出 n 个数的中位数，快速排序算法 (12-A1) 即可优化至 $O(n \log n)$。
14. 将 N 个关键码按随机次序插入 B 树，则期望的分裂次数为 $O(\log^2 N)$。
15. 与二叉堆相比，多叉堆 delMax() 操作时间复杂度更高。
16. 若元素理想随机，则用除余法作为散列函数时，即使区间长度不是素数，也不会影响数据的均匀性。

17. 与胜者树相比，败者树在重赛过程中，需反复将节点与其兄弟进行比较。

18. 在图的优先级搜索过程中，每次可能调用多次 **prioUpdater**，但累计调用次数仍为 $O(e)$。

19. 若序列中逆序对个数为 $O(n^2)$,则使用快速排序 (12-A1) 须进行的交换次数为 $O(n \log n)$。

20. 如果把朋友圈视为一无向图，那么即使 A 君看不到你给 B 点的赞，你们仍可能属于同一个双联通分量。

2 选择 (3'×10)

1. 二叉堆中某个节点秩为 k，则其兄弟节点（假设存在）的秩为（　）

   A. $k+1$

   B. $k-1$

   C. $k+(-1)^k$

   D. $k-(-1)^k$

   E. 以上皆非

2. 由 5 个互异节点构成的不同的 BST 共有（　）个

   A. 24

   B. 30

   C. 36

   D. 42

   E. 120

3. 有 2015 个节点的左式堆，左子堆最小规模为（　）(不计外部节点)

   A. 10

   B. 11

   C. 1007

   D. 1008

   E. 以上皆非

4. 与 MAD 相比，除余法在（　）有缺陷

   A. 计算速度

   B. 高阶均匀性

   C. 不动点

   D. 满射性

   E. 以上皆非

5. 以下数据结构，在插入元素后可能导致 $O(\log n)$ 次局部结构调整的是（　）

A. AVL

B. B-树

C. 红黑树

D. 伸展树

E. 以上皆非

6. 对小写字母集的串匹配，KMP 算法与蛮力算法在（　）情况下渐进的时间复杂度相同

A. 最好

B. 最坏

C. 平均

D. 以上皆非

7. 对随机生成的二进制串，gs 表中 gs[0]=1 的概率为（　）

A. $\frac{1}{2^m}$

B. $\frac{1}{2^{m-1}}$

C. $\frac{1}{2^{m+1}}$

D. $\frac{1}{m}$

8. 以下数据结构，空间复杂度为线性的是（　）

A. 2d-tree

B. range tree

C. interval tree

D. segment tree

E. 以上皆非

9. 人类拥有的数字化数据数量，在 2010 年已达到 ZB($2^{70} = 10^{21}$) 量级。若每个字节自成一个关键码，用一颗 16 阶 B-树存放，则可能的高度为（　）

A. 10

B. 20

C. 40

D. 80

E. >80

10. 在 BST 中查找 365，以下查找序列中不可能出现的是（　）

A. 912, 204, 911, 265, 344, 380, 365

B. 89, 768, 456, 372, 326, 378, 365

C. 48, 260, 570, 302, 340, 380, 361, 365

D. 726, 521, 201, 328, 384, 319, 365

3 综合 (10'×3)

1. 在有向图 G 中,存在一条自顶点 V 通向 u 的路径,且在某次 DFS 中有 dTime[v]<dTime[u],则在这次 DFS 所生成的 DFS 森林中，v 是否一定是 u 的祖先？若是，请给出证明；若不是，请举出反例。

2. 对闭散列 $[0, M)$，$M = 2^S$，采用如下冲突排列解决方法：

   - 初始时，$c = d = 0$
   - 探查 key 冲突时，$c \leftarrow c + 1$，$d \leftarrow d + c$，探查 H[(key+d)%M]

   则这种算法是否可以保证空间能被 100% 利用？若是，请给出证明；若不是，请举出反例。

   ($\frac{(j+i+1)(j-i)}{2}\%M = 0$，又 $j+i+1$，$j-i$ 必一奇一偶，$\frac{j-i}{2} < \frac{j+i+1}{2} < M$，矛盾)

3. 在不改变 BST 和 BinNode 定义的前提下 (BinNode 仅存储 parent，data，lc，rc)，设计算法，使得从节点 x 出发，查找值为 Y 的节点 y 的时间复杂度为 $o(d)$，d 为节点 x 与 y 的距离。要求利用树的局部性，复杂度与总树高无关，否则将不能按满分起评。

   函数定义式:参量为 BinNode x,y,T,返回值为 BinNode 类型,函数名 **fingerSearch**

   (a) 说明算法思路

   (b) 写出伪代码

   (c) 在图中画出由值为 6 的点查找值为 17 的点的查找路径

   (d) 说明算法时间复杂度为 $O(d)$ (若无法达到，说明困难在哪)