

习题答疑课（二）

习题答疑课（二）

- 一、预习资料-【数据结构】-最短路简化
- 二、预习资料-【数据结构】-游戏分组
- 三、预习资料-【数据结构】-字符串旋转矩阵
- 四、预习资料-【数据结构】-灌溉
 - 最小生成树问题-Kruskal 算法
- 五、Leetcode-05-最长回文子串
 - Manacher 算法

一、预习资料-【数据结构】-最短路简化

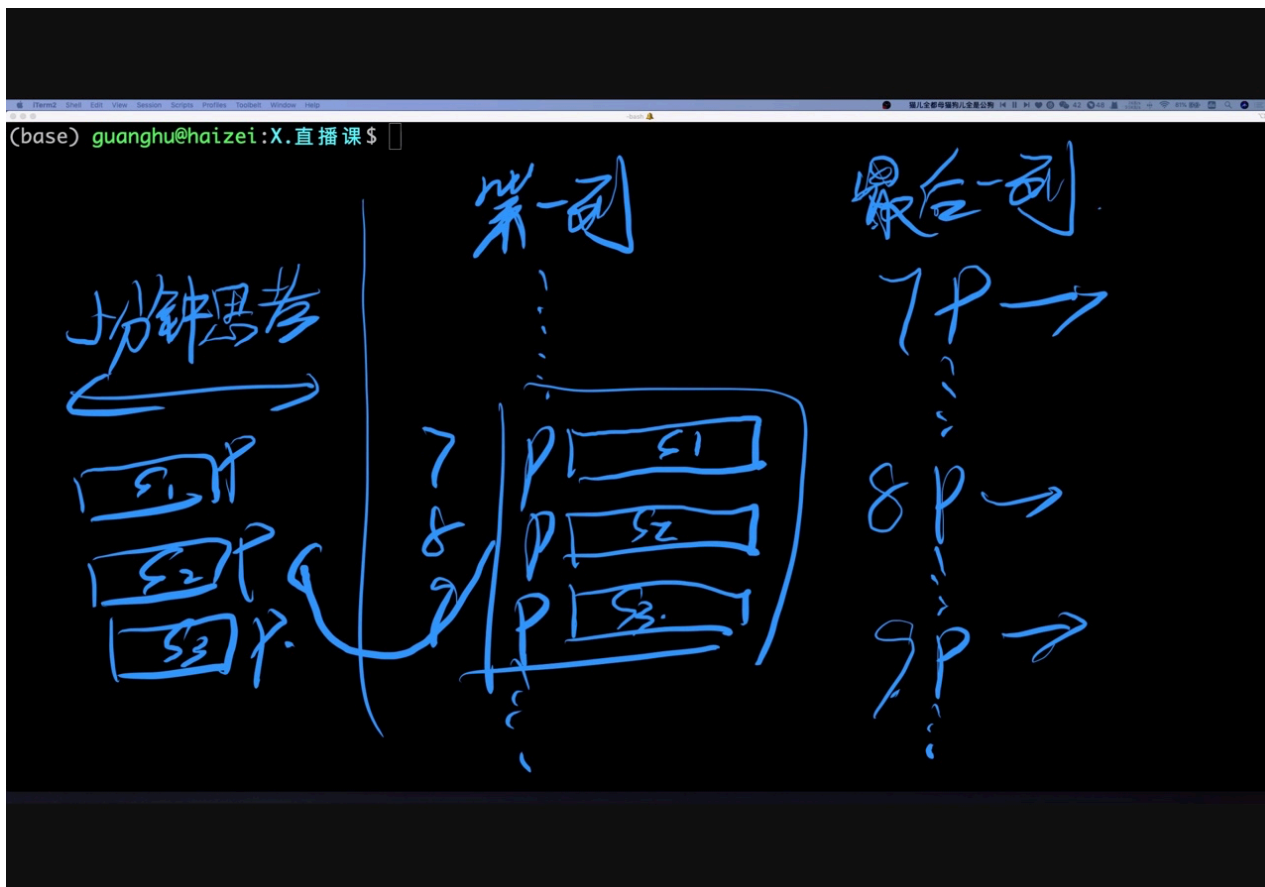
1. 广搜裸题

二、预习资料-【数据结构】-游戏分组

1. 并查集裸题
2. 并查集的简化实现方式
3. 路径压缩：代码实现短，优化效果明显
4. 按秩优化：代码实现稍复杂，对于随机数据的优化效果一般，还需要占用额外的存储空间
5. 在快速编码的场景中，只需要实现路径压缩即可

三、预习资料-【数据结构】-字符串旋转矩阵

1. 首要解决的问题：还原原字符串
2. **Step1**：通过最后一列的字符，确定相应的第一列的字符
3. **Step2**：得出结论，相同字符的相对顺序在第一列和最后一列中相同
4. **Step3**：确定了每一个字符的编号，以及编号与编号之间的前后关系，即可还原原字符串
5. 由于反复标记 fail 指针链上的状态，导致时间超限，在这里有一个特殊的程序优化

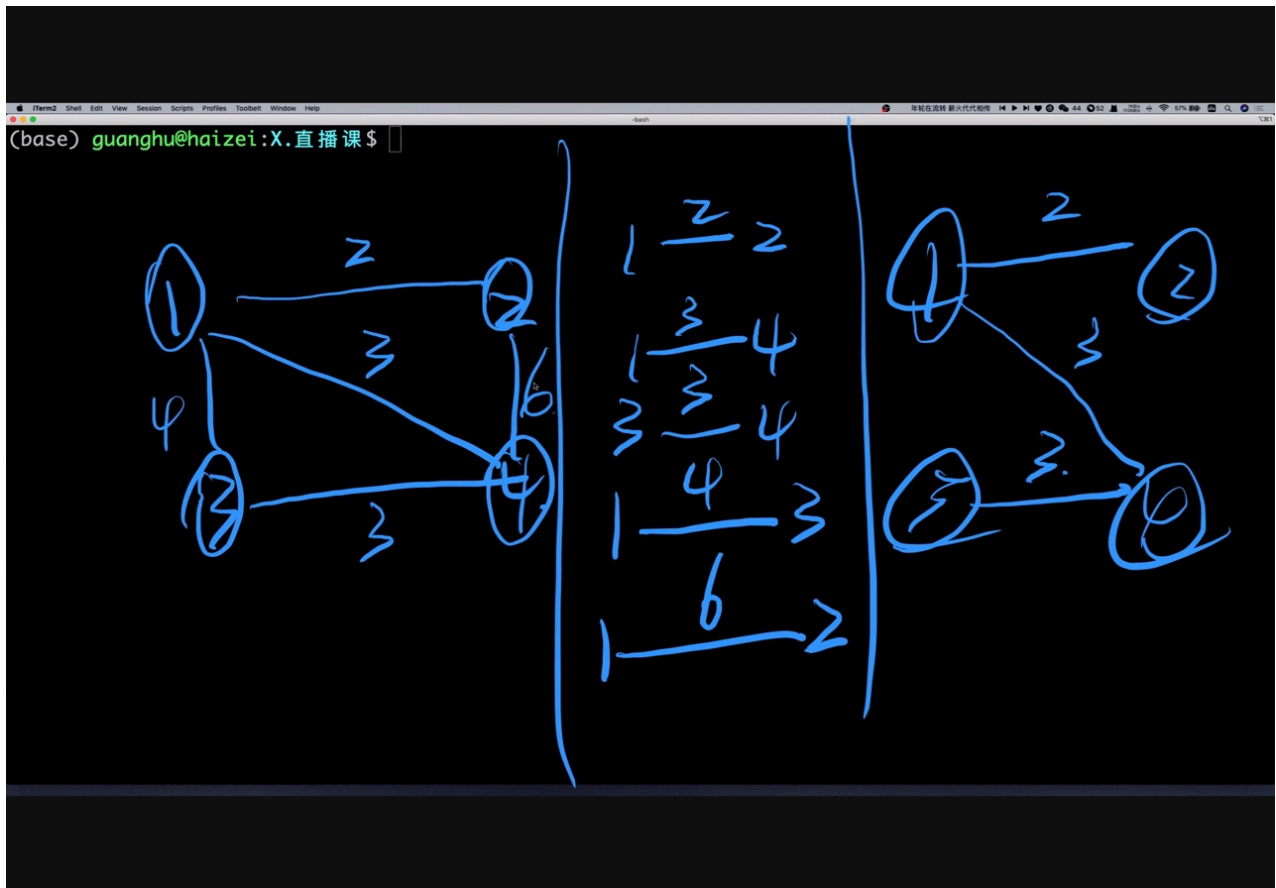


四、预习资料-【数据结构】-灌溉

最小生成树问题-Kruskal 算法

1. Step1: 对于所有边从小到大排序
2. Step2: 依次取出每一条边，试着加入图中，边上的两个点不联通的情况下，才将边加入图中

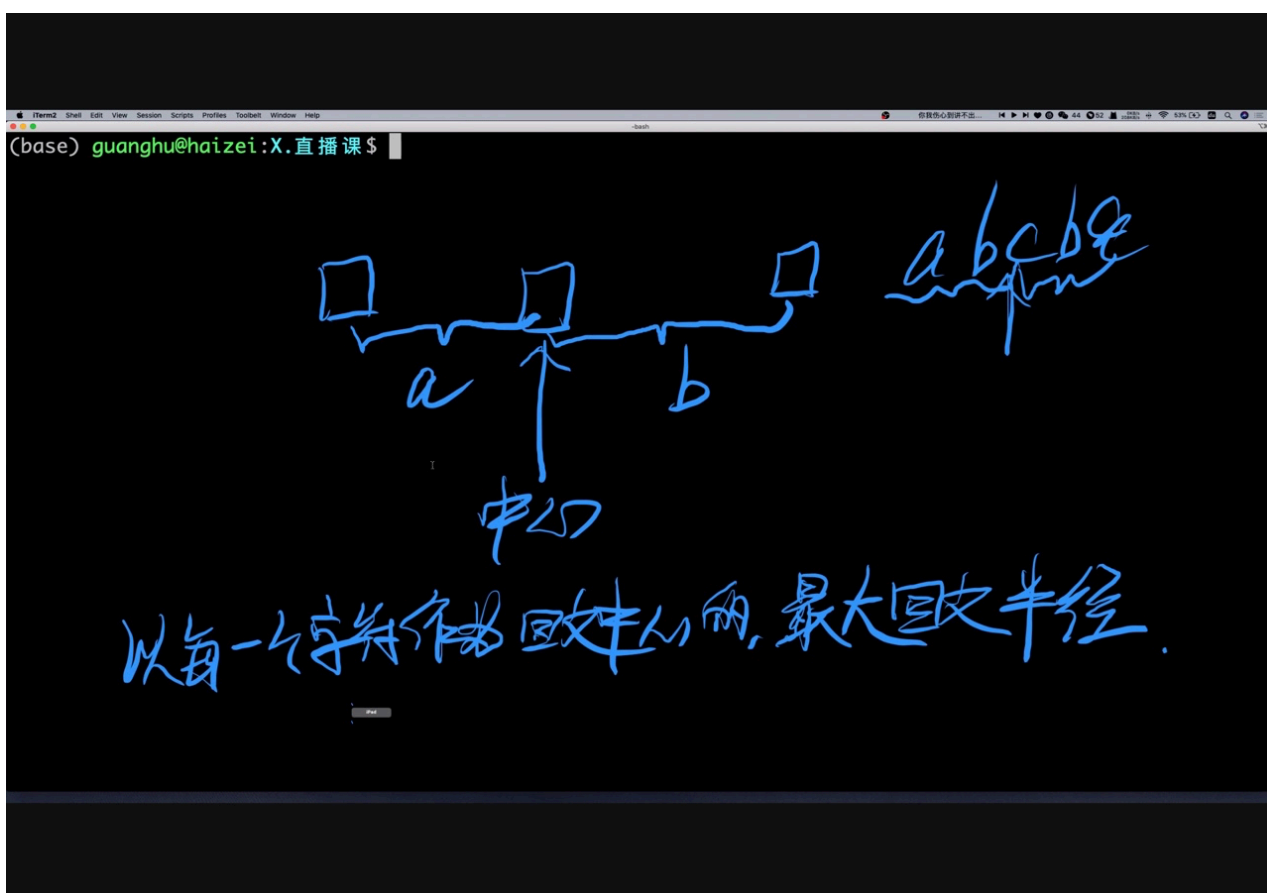
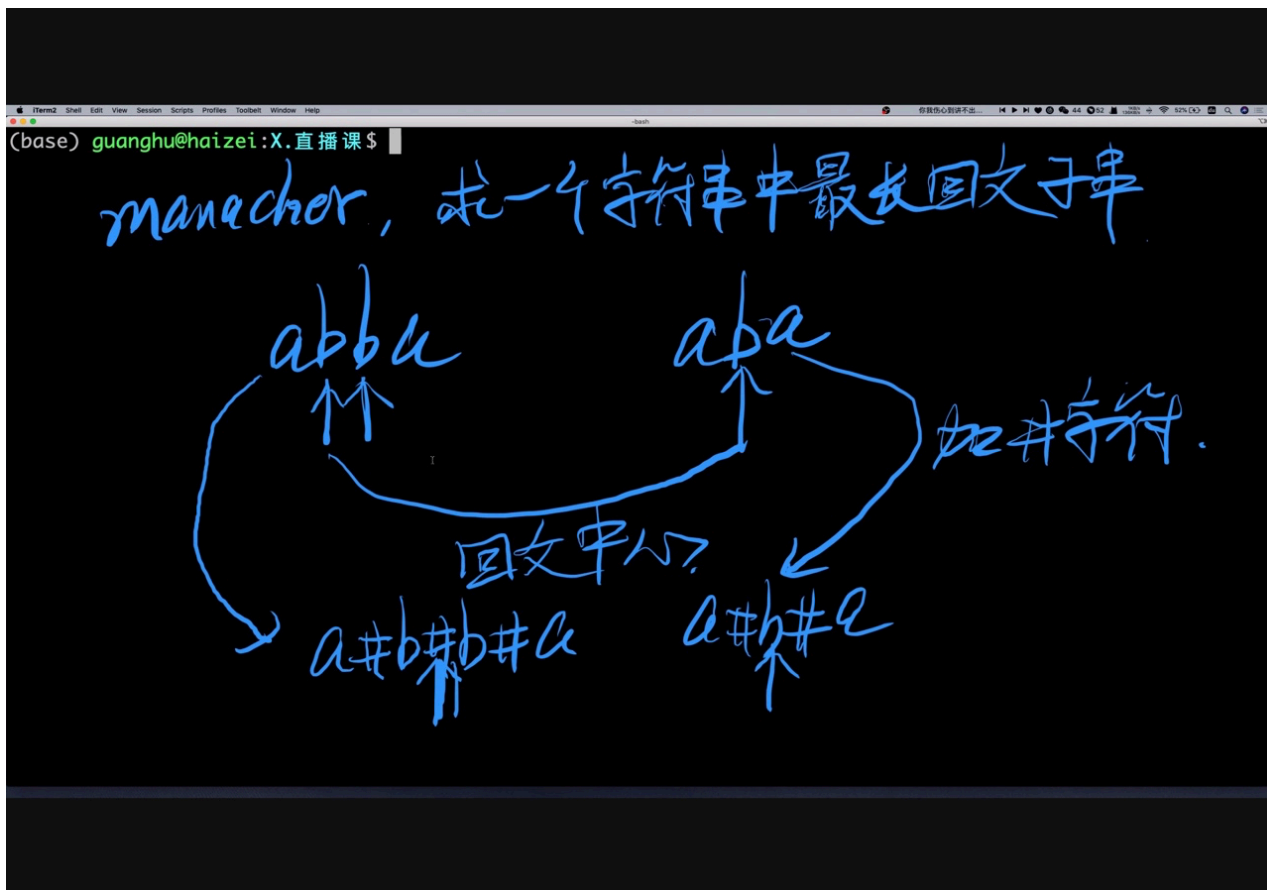
除了学习一个新算法以外，这道题目，一无是处。



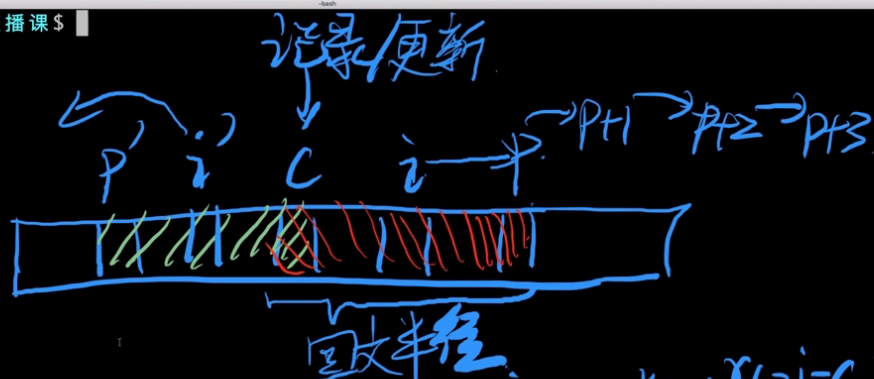
五、Leetcode-05-最长回文子串

Manacher 算法

1. Step1: 对原字符串进行特殊处理，在每两个字符中间加入 # 字符
2. Step2: 依次求解每一个位置的最大回文半径
3. 在求解过程中，记录一个 c 点， $c + r[c]$ 最大
4. 求回文半径时：要不然借助原有信息，要不然就是暴力匹配，暴力匹配过程，均摊时间复杂度 $O(1)$
5. 时间复杂度： $O(n)$



(base) guanghu@haizei: X.直播课 \$



欧氏半径

1. 欧氏半径长度 $\leq p$ 超不过 $p \rightarrow r(p) \rightarrow r(2i-c)$

2. 超过 p , 暴力遍历求解