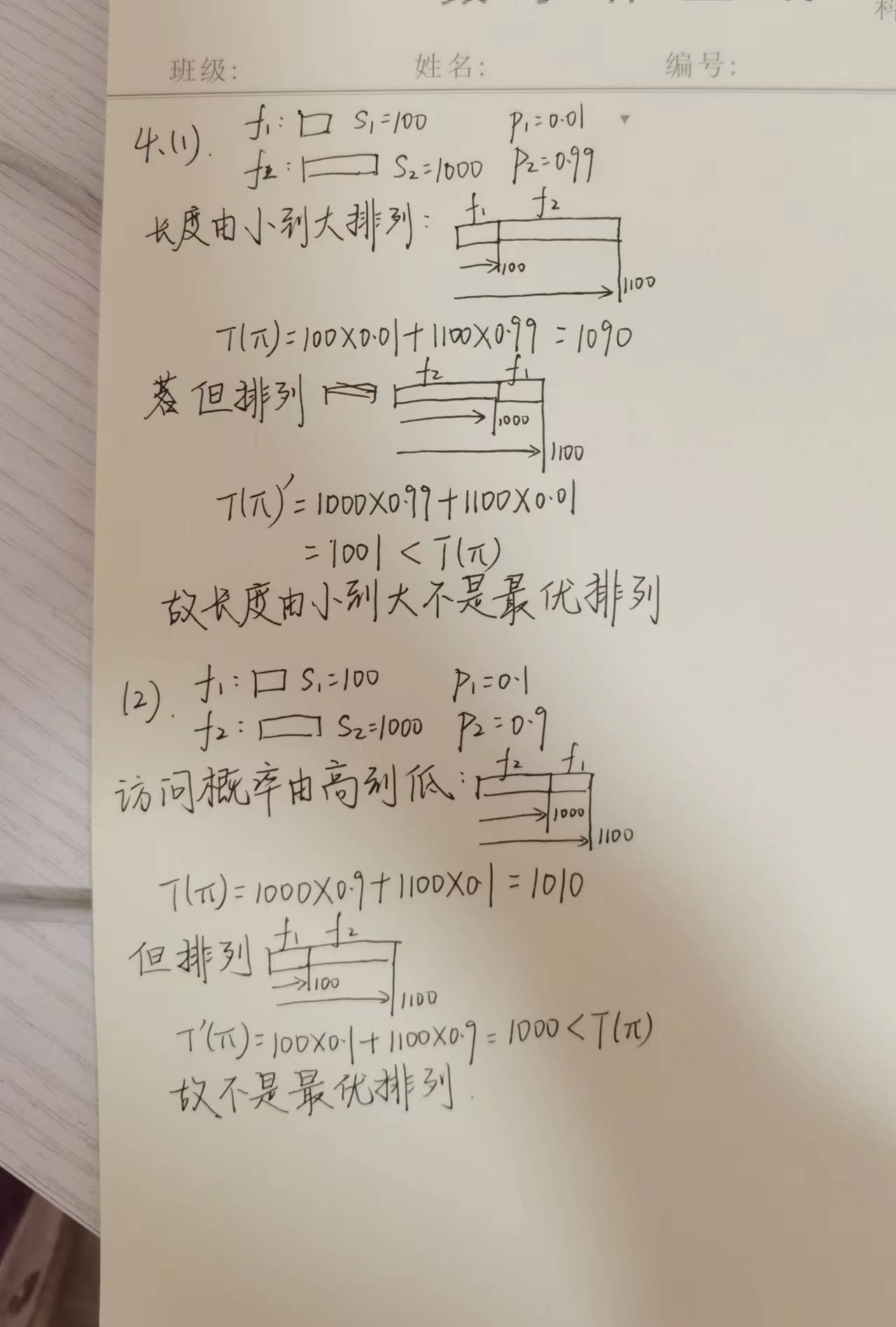
13.2 d

14.2 b

14.3 c

4.



（3）算法：首先计算每个文件的长度和访问概率的商，然后对序列按照si/pi升序排序。

正确性证明：令 G 为贪心算法产生的调度，令 O 为任意最优调度。如果G = O，那么贪心是最优的，我们就完成了。否则，O 必须包含至少一个反转，即至少有一对未按si/pi升序排序调度的文件，记为fi、fj。也就是说，设fi和fj是调度 O 中的前两个连续任务，使得 sj/pj < si/pi。设i文件前有长度s，则O的总成本T1=(s+si)\*pi+(s+si+sj)\*pj。在O中交换fi和fj，交换后的总成本T2=(s+sj)\*pj+(s+si+sj)\*pi，T1-T2=sipj-sjpi > 0，即交换后成本会降低，是更优的策略。以此类推，贪心算法G就是最优策略

运行时间：O（nlogn），为排序所需时间复杂度

5.（a）假设当前存在一个刺穿集，其中存在不是某个区间的右端点(bi)的刺穿点。则将第一个不是某个区间的右端点(bi)的刺穿点xi变为xi’，使其位于xi所刺穿的区间中最小的右端点bi处，则xi’没有超出任何一个xi所刺穿的区间，并且还可能因为经过某个区间的左端点而使刺穿的区间数增加。则相应的刺穿集中点的个数要么不变，要么因为xi’刺穿的区间增加而减少。以此类推，对于任何一组区间，都存在一个最小尺寸的刺穿集，其中每个刺穿点都是某个区间的右端点(bi)。

（b）算法思想：将所有右端点按从小到大的顺序排序，然后每两个右端点合为一组，选择组内深度更大的端点作为该组的端点（若深度相等则选择更小的端点），得到n/2个端点，然后再两两分组以此类推，最终得到深度最大的端点。

MD启发式算法不是最优的

（c）T = nlogn+(k-1)n/k\*log((k-1)n/k)+……+2n/k\*log(2n/k)+ n/k\*log(n/k)

（d）

