

EX4 实验报告

王嵘晟 PB1711614

EX4-1

这道题明显使用贪心算法可以解决问题。由于输入的村庄坐标是乱序，所以先对坐标进行从小到大排序，使用快排，时间复杂度为 $O(N\lg N)$ ，然后使用循环来遍历每个村庄的坐标，来判断是否放置基站。贪心选择为放置的一个基站能尽可能多的覆盖村庄，由此可证这种贪心算法可以得到最优解。贪心选择的时间复杂度为 $O(N)$ 。

总的时间复杂度为 $O(N\lg N)$ 。

EX4-2

这道题可以使用贪心算法来解决问题。由于输入的任务到达时间是乱序的，所以先对任务到达时间和执行时间进行排序，到达时间相同的执行时间少排在前面，使用快速排序，则时间复杂度为 $O(N\lg N)$ 。贪心选择为每次选择当前时间结点，到达的任务中剩余执行时间最短的任务执行，即抢占，这样可以保证总的响应时间最短。由于要维护每次选择执行时间最短，所以要选择一个合理的数据结构来存储结点。这里我使用了二叉搜索树 **BST**，当时间流动时当前任务没有执行完，则把剩余的任务插入到数里，当当前时间树中最小剩余执行时间结点小于到达任务的执行时间时，则执行树中的任务。插入结点的过程时间复杂度为 $O(1)$ ，找最小值时间复杂度为 $O(\lg N)$ ，删除结点的时间复杂度为 $O(\lg N)$ 。由于这些对于 **BST** 的操作都是在 **N** 重循环中执行的，所以总的时间复杂度为 $O(N\lg N)$ 。

EX4-3

这道题可以使用贪心算法来解决。贪心选择在于缓存替换时选择被替换掉的为下次出现在输入序列中最晚的数据。为了解决这个问题，我使用了斐波那契堆作为存储缓存

到的数据的数据结构。当输入一个数据且缓存槽没有满，且 **MISS** 了时，将数据插入斐波那契堆，同时记录下一次这个数据出现的位置。当缓存槽满了，且 **MISS** 了，需要替换时，找到缓存槽中下一次出现的最晚的数据，做替换。当缓存成功 **HIT** 时，则无需替换，只需要更新缓存槽中每个数据下一次出现的位置。建立空的斐波那契堆时间复杂度 $O(1)$ ，插入一个数据时间复杂度为 $O(1)$ ，删除最小结点的时间复杂度为 $O(\lg N)$ ，斐波那契堆查找的时间复杂度为 $O(1)$ 。这些操作都是在 N 重循环中进行的，所以总的时间复杂度为 $O(N \lg N)$ 。但是由于更新下一次出现的时间时最坏情况需要 $O(N)$ ，所以程序的总时间复杂度为 $O(n^2)$ 。

EX4-EX

这道题可以使用贪心算法来解决。不过改收回衣服为放置衣服。贪心选择在于在放置衣服间隔尽量大的前提下放足够多的衣服，即 $N-M$ 件。为了实现这一目的，我令间隔从 $mid = \frac{clothes[0] + clothes[N-1]}{2}$ 二分递减，直到恰好能放置 $N-M$ 件衣服，这时要让间隔尽量大，所以令 $left = mid + 1$ ，令间隔变为原来的 1.5 倍再执行一遍看能不能放置的下。最后可以得到最大的最小间隔。这种方法的时间复杂度为 $O(N \lg N)$ 。