算法设计与分析

课 程 设 计 报 告

课程名称 算 法 分 析 与 设 计

实验学期 2016 年至 2017 年 第 2 学期

所在学院 数学与信息学院 年级 2014

专业班级 信息与计算科学 3 班

学生姓名 何 唯 学号 201430120308 成绩 ___

学生姓名 刘晓婷 学号 201430120318 成绩 ___

指导教师 赵 峰

《算法分析与设计》课程设计报告

设计题目　**巴士司机的加班费**

设计时间　2017 年 05 月 25 日　设计性质　　☐应用性 ☑设计性 ☑综合性

设计成绩

教师评阅：

☐ 设计目的明确；☐操作步骤正确；☐设计文稿（表格、程序、数据库、网页）符合要求；

☐ 设计结果正确；☐设计分析总结全面；☐设计报告规范；

课程设计答辩情况记录：

☐ 思路清晰；语言表达准确，概念清楚。

☐ 准备工作充分，具备必要的报告资料；报告在规定的时间内完成。

☐ 回答问题有理论依据，基本概念清楚。主要问题回答简明准确；

☐ 对前人工作有改进或突破，或有独特见解。

评阅教师签名：

目　　录

1 题目描述......4
2 符号说明......5
3 算法设计思想......6
4 算法选择理由......6
4.1 贪心选择性质......6
4.2 最优子结构性质......15
5 求解过程......15
6 算法复杂度及其推导过程......18
6.1 算法时间复杂度......18
6.2 算法空间复杂度......19
7 实验截图......20
7.1 实验测试用例一输入......20
7.2 实验测试用例二输入......20
7.3 实验测试用例三输入......21
8 设计小结......22
8.1 何唯......22
8.2 刘晓婷......24
9 各人在小组中的完成的工作......25
10 自我评定成绩(按百分制)......25
参考文献......26

1 题目描述

巴士司机的加班费

有 n 个巴士司机，每个司机已经有了一个白天路线，每个白天路线有确定的一个行驶时间。现在巴士公司新增加了 n 条夜班线路，每个夜班线路也都有一个确定的行驶时间现在需要给每个司机再分配一条夜班线路。根据法律规定，每位司机工作时间（行驶时间）超过某个数值 d 后，公司需要为超出部分支付加班费。为简单起见，设每超出 1 小时，公司需要支付加班费 1。但公司不想指出太多的加班费，所以需要根据司机白天线路的行驶时间为其安排夜班线路。

问题：

如何分配夜班线路，使公司的加班费支出最小。

输入：

加班费的支出时间限 d ，司机数 n ，每个司机的白天线路行驶时间，n 个夜班线路的行驶时间。

输入格式：

- 第一行为加班费的支出时间限 d
- 第二行为司机数 n
- 第三行有 n 个数，对应各司机白天线路的行驶时间
- 第四行有 n 个数，对应第 i 个夜班线路的行驶时间

输出：

1~ n 的一个排列（第 i 个数 k 表示为第 i 为司机安排第 k 个夜班线路），以及公司需要支付的共加班费

2 符号说明

符号名称	意义
n	司机总个数
d	加班费的支出时间限
i	序号
D_i	原始输入顺序的第 i 个司机 $i=1,2,...,n$
$\mathrm{NIGHTTIME}_{\mathrm{i}}$	原始输入顺序的第 i 个夜班线路行驶时间 $i=1,2,...,n$
$\mathrm{DAYTIME}_i$	原始输入顺序的第 i 个司机白天路线行驶时间 $i=1,2,...,n$
d_i	按照白天工作时间 $\mathrm{DAYTIME}_i$ 与支出时间限 d 的差降序排序后的第 i 个司机 $i=1,2,...,n$
$\mathrm{daytime}_i$	上述差值排序后的第 i 个司机白天路线行驶时间 $i=1,2,...,n$
$\mathrm{nighttime}_{\mathrm{i}}$	按照夜班线路时间降序排列后的第 i 个夜班 $i=1,2,...,n$ 线路时间
cost	问题最优解的数值

3 算法设计思想

本题目——巴士司机的加班费将设计**贪心算法**求解。贪心算法通过一系列的选择得到问题的解，它做出的每一个选择都是当前状态下局部最好选择，即贪心选择。对于问题能否用贪心算法求得最优解，将需要说明问题具有贪心选择性质以及最优子结构性质。因此下面将证明并说明这两个性质。

注意的是，本设计采用的**贪心选择策略**为：白天工作时间与加班费支出时间限的差越大的司机优先与夜间线路行驶时间长的线路匹配。可产生加班费总额最小的最优解，并获得司机与夜间线路匹配的序列。

4 算法选择理由

对于选择贪心算法的原因，首先，在于本问题贪心选择性质以及最优子结构性质可以得证；其次，在于贪心算法设计思路简单易懂，从局部最优到全局最优这种解题方式更加简单、更加直接、且解题效率更高。

4.1 贪心选择性质

对于贪心选择策略，**对于夜班行驶时间递减序列，优先安排可工作但不加班工作时间最长的司机，即对于差值排序后的司机序列，与夜班行驶时间递减序列先后进行匹配。**贪心算法通过自顶向下的方式进行，做出相继贪心选择策略后，每次都将所求问题简化为规模更小的子问题。

类似于文献[1]中提到的活动安排问题的贪心选择性质证明方法，要确定本设计加班费问题是否具有贪心选择性质，只需证明每一步做出的贪心选择最终导致问题的整体最优解。

证明过程：

设贪心算法得到的解为 $(\mathrm{d}_1,\mathrm{nighttime}_1),(\mathrm{d}_2,\mathrm{nighttime}_2),\ldots,(\mathrm{d}_n,\mathrm{night\,time}_n)$ ，如图一所示。一个最优解 A 为 $(\mathrm{d}_1,\mathrm{nighttime}_1'),(\mathrm{d}_2,\mathrm{nighttime}_2'),\ldots,(\mathrm{d}_n,\mathrm{night\,time}_n')$ ，如图二所示。此

时，对于问题第一步贪心选择策略，即司机d_1的夜班线路匹配选择来说，贪心算法将其匹配到$nighttime_i$。此处，司机序列以及夜班线路序列为根据输入进行非增排序得到的序列，即

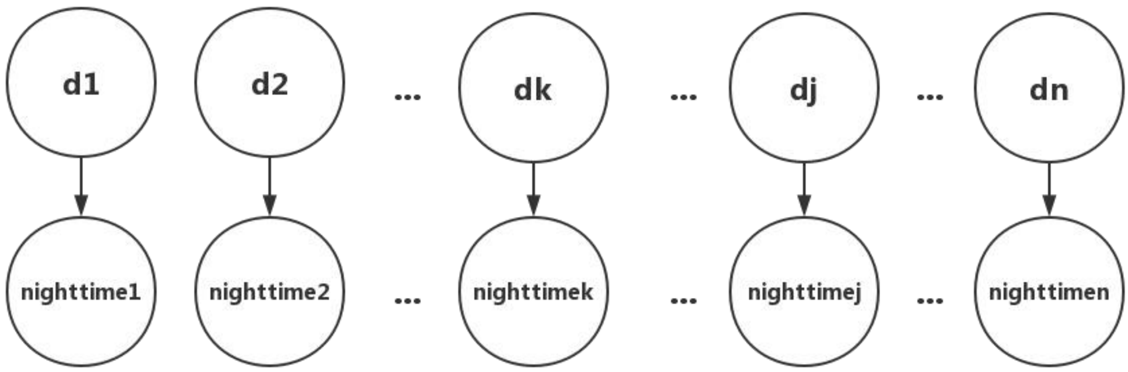
$$d\text{-}daytime_1 \ge d\text{-}daytime_2 \ge \ldots \ge d\text{-}daytime_n, nighttime_1 \ge nighttime_2 \ge \ldots \ge nighttime_n \quad ①$$

而对于最优解 A，

1) 若 A 中含有匹配$(d_1, nighttime_1)$，则说明贪心选择策略正确，选择结果$(d_1, nighttime_1)$,其中$nighttime_1 \ne nighttime_1'$，包含在某个最优解中。

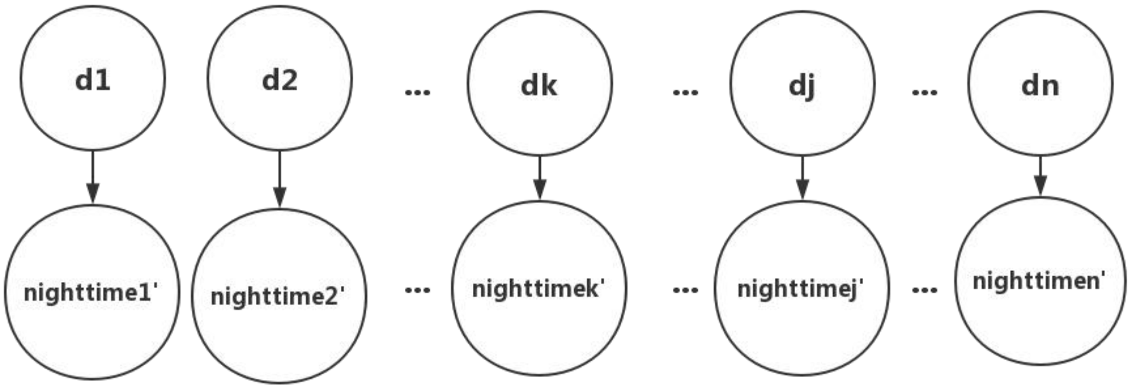
2) 若 A 中不含有匹配$(d_1, nighttime_1)$，则必然有司机d_1的另一个匹配$(d_1, nighttime_1')$,其中$nighttime_1 \ne nighttime_1'$。这里设$nighttime_1' = nighttime_k$,其中$k \ne 1$，即司机$d_1$匹配夜班线路为贪心解中司机$d_k$匹配的夜班线路。而夜班线路$nighttime_1$在 A 中匹配到司机$d_j$。如图三所示，即

贪心解：



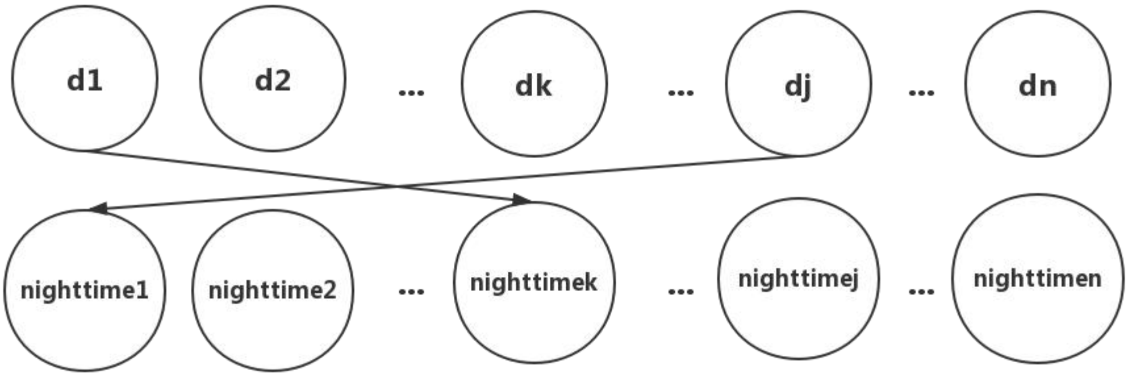
图一 贪心解

某个最优解 A (其中， $nighttime_i'$ 对应任意一个 $nighttime_i$):

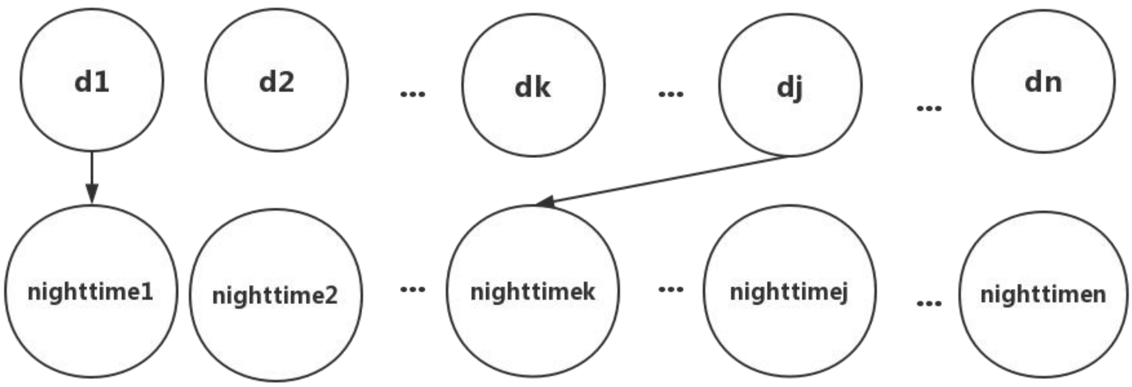


图二 最优解 A

假设最优解 A 对应司机 d_1 以及司机 d_1 在贪心解中原对应的夜班线路 $nighttime_1$ 在最优解 A 中，对应的是以下匹配：



图三 最优解 A 对应原序列结果



图四 最优解 A 交换后结果

要证明此问题贪心选择性质，此处考虑交换 $\mathrm{nighttime}_1$ 与 $\mathrm{nighttime}_k$ 的匹配司机，只需证明交换后也是对于司机 1 选择匹配的一个最优解，即构造出贪心解并说明对比后总加班花费不多于本最优解 A 即可。这里设最优解 A 得到解大小为 cost。

需要说明的是，最优解的数值可以通过如下函数求和得到，

$$\cos t=\sum_i \frac{\mathrm{sgn}(\mathrm{nighttime}_i-(\mathrm{d}-\mathrm{daytime}_i))+1}{2}\cdot(\mathrm{sgn}(\mathrm{nighttime}_i-(\mathrm{d}-\mathrm{daytime}_i)))$$

其中

$$\mathrm{sgn}(\mathrm{x})=\begin{cases}1 & x>0\\ 0 & x\le 0\end{cases},$$

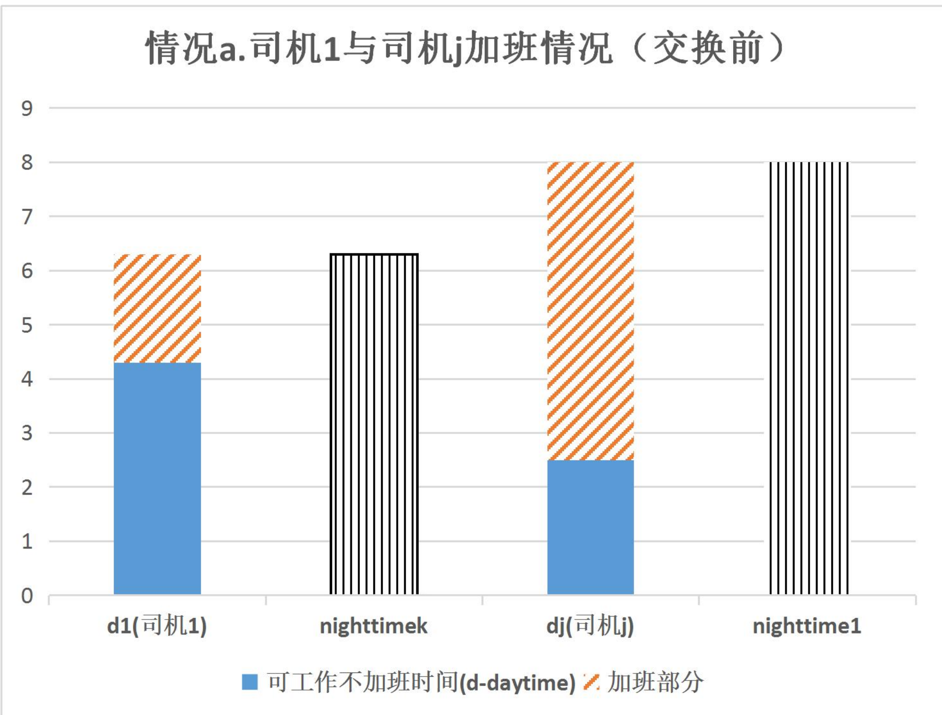
首先，讨论任意情况，即 d_k 不一定等于 d_j，在最优解 A 中，司机 d_1 匹配到司机 d_k 的贪心解中的夜间线路，而司机 d_j 匹配到司机 d_k 的贪心解中的夜间线路。

交换分如下情况讨论：

a. 交换前司机 d_1 与司机 d_j 匹配夜班线路后均要加班

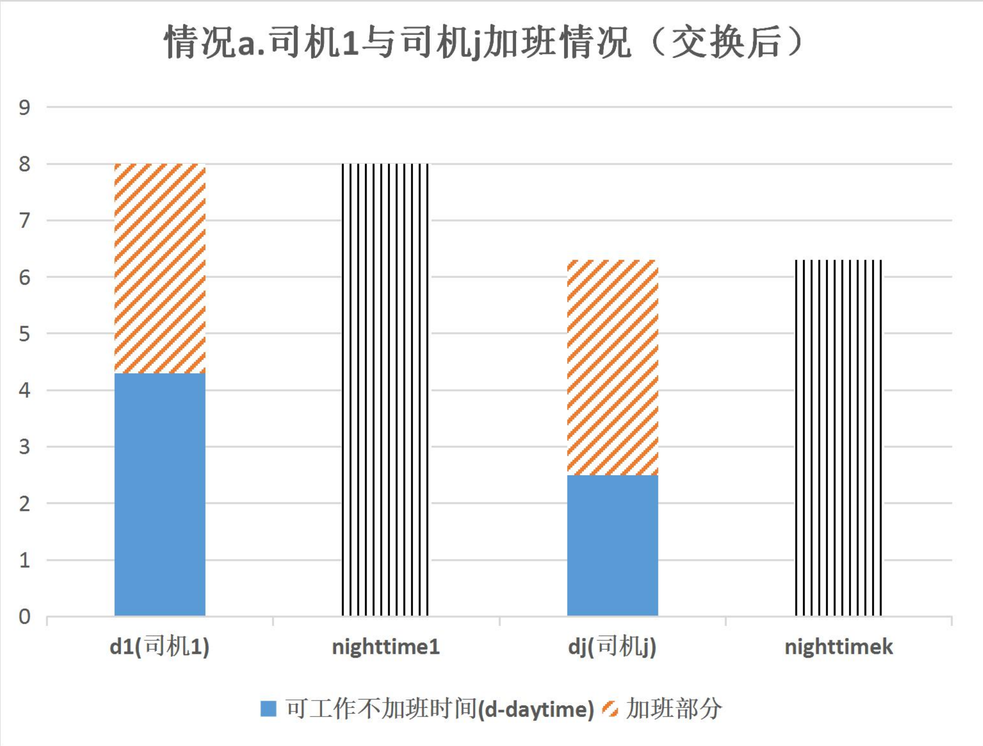
易得 $\mathrm{d\text{-}daytime}_1\le \mathrm{nighttime}_k$，$\mathrm{d\text{-}daytime}_k\le \mathrm{nighttime}_1$，如表一所示，

表一 交换前司机 d_1 与司机 d_j 匹配夜班线路后均要加班



- 交换后只会发生两个司机都要加班的情况，且加班费明显不变；

表二 情况 a 交换后两个司机都加班



- 交换后不会存在司机 j 要加班，司机 1 不加班或者交换后司机 1 加班，司机 j 不加班的情况，原因在于：

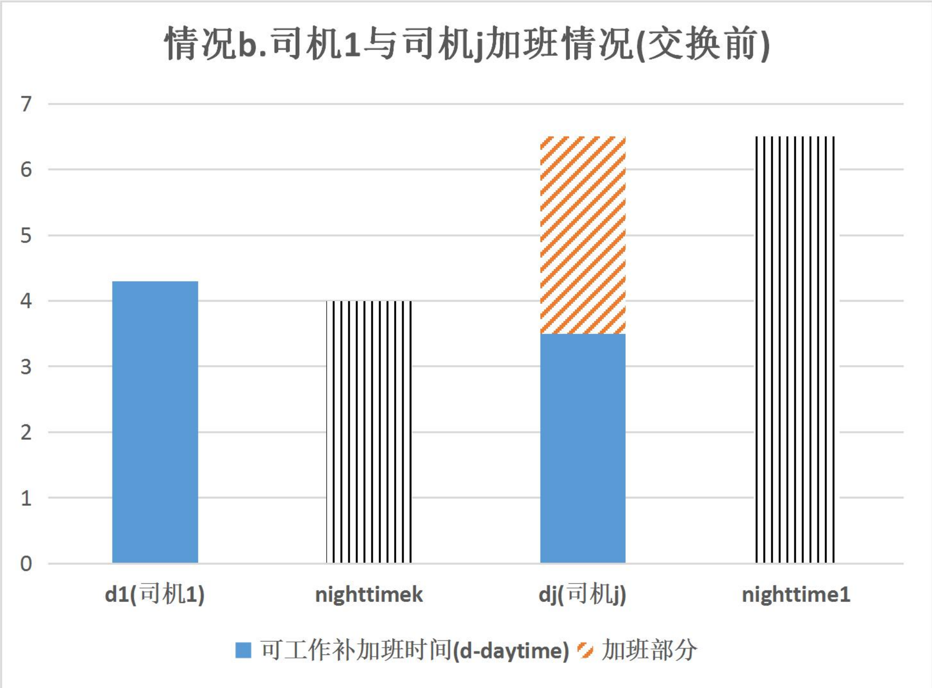
$$\text{d-daytime}_1 \le \text{nighttime}_k，\text{d-daytime}_j \le \text{nighttime}_1$$

而且由于①，$\text{nighttime}_k \le \text{nighttime}_1$，因此只会存在两个司机都加班的情况，因此可以证明交换后的解的大小不会增加(如表二)，因此贪心算法在此情况可以得到最优解。**这里设交换后得到的最优解大小为** $\cos t'$**，即** $\cos t - \cos t'=0$

综上所述，对于情况 a，$\cos t'=\text{cost}$，贪心选择策略可得某个最优解。

b. 交换前司机 d_1 不加班而司机 d_j 加班

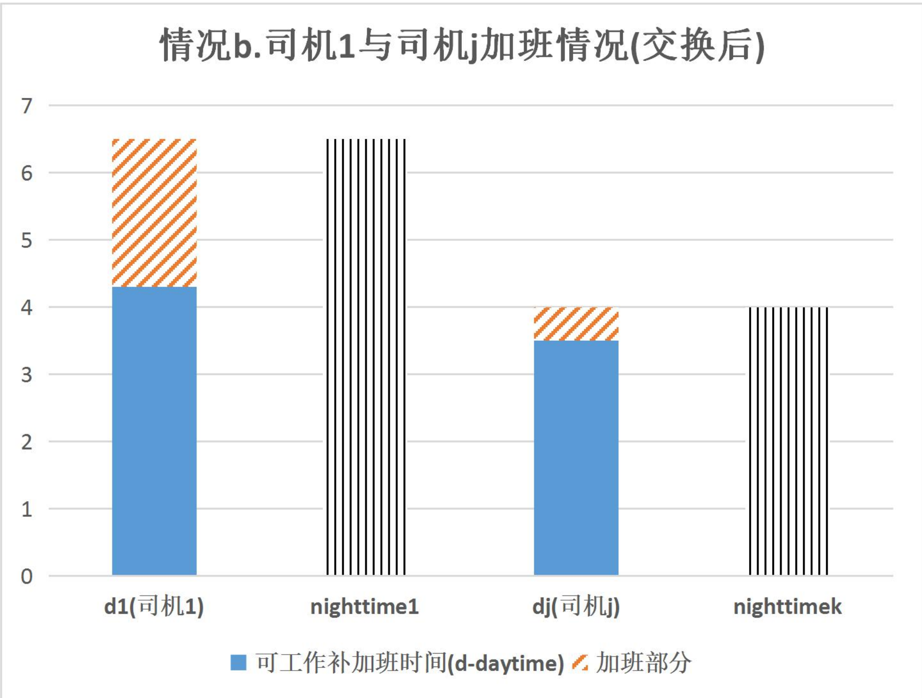
表三 情况 b 交换前司机 1 不加班，司机 j 加班



$$cos\,t = nighttime_1 - (d - daytime_j)$$

- **交换后如果司机 d_1 加班并且司机 d_j 也加班**，则

表四 情况 b 交换后司机 d_1 加班并且司机 d_j 也加班



$$\cos t'=\mathrm{nighttime}_1-(\text{d-daytime}_1)+\mathrm{nighttime}_k-(\text{d-day time}_j),$$

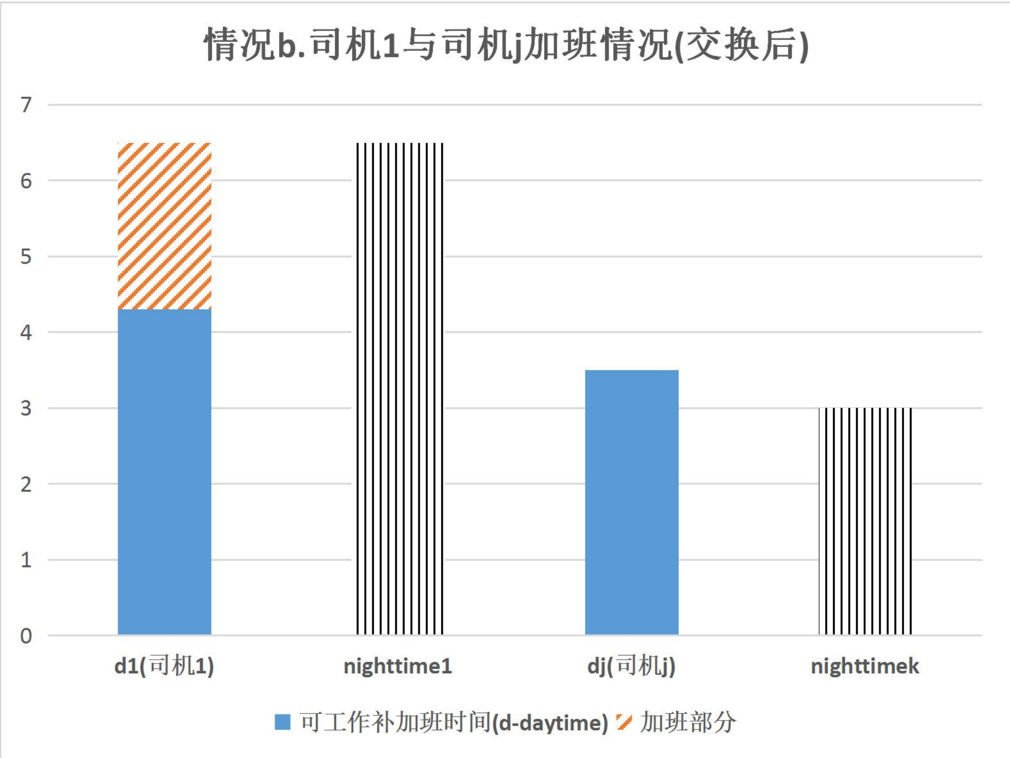
$$且\ \cos t'\text{-cost}=\mathrm{nighttime}_k-(\text{d-daytime}_1),$$

由于之前司机 1 对于 $\mathrm{nighttime}_k$ 没有加班，因此 $\cos t'\text{-cost}=\mathrm{nighttime}_k-(\text{d-daytime}_1)\le 0$,可得，

$$\cos t'\le \mathrm{cost}$$

- **交换后如果司机 d_1 加班而司机 d_j 不加班**

表五 情况 b 交换后司机 d_1 加班，司机 d_j 不加班



$$\cos t'=\mathrm{nighttime}_1-(\text{d-daytime}_1),$$

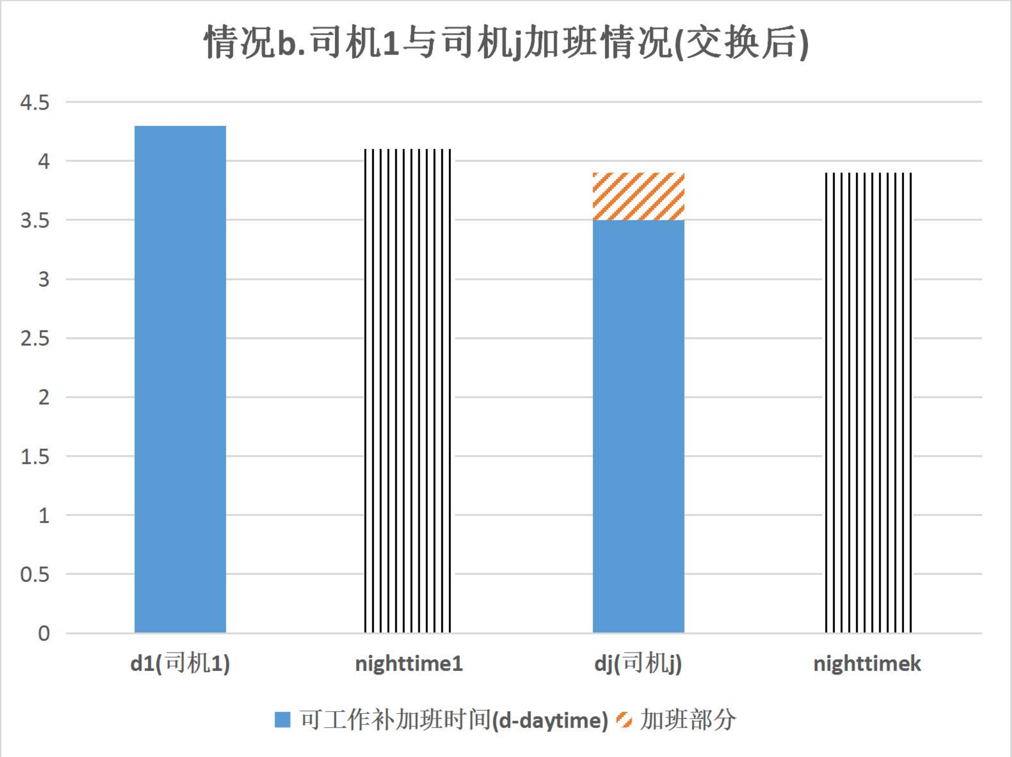
$$且\ \cos t'\text{-cost}=(\text{d-daytime}_j)-(\text{d-daytime}_1),$$

由①可得，$(\text{d-daytime}_j)\le(\text{d-daytime}_1)$，因此，$\cos t'\text{-cost}=(\text{d-daytime}_j)-(\text{d-daytime}_1)\le 0$ ，

$$\cos t'\le \mathrm{cost}$$

- **交换后如果司机 d_1 不加班，而司机 d_j 加班，则，**

表六 情况 b 交换后司机 d_1 不加班，司机 d_j 加班



$$\cos t' = nighttime_k - (d\text{-}daytime_j),$$

且 $\cos t'\text{-}cost = nighttime_k\text{-}nighttime_1$ 显然小于 0，易得，

$$\cos t' \le cost$$

- **交换后如果司机 d_1 和司机 d_j 均不加班，明显得到的 $\cos t' \le cost$**

综上所述，对于情况 b， $\cos t' \le cost$ ，贪心选择策略可得某个最优解。

c. 交换前司机 d_1 加班，司机 d_j 不加班

对于排序后的序列必定不会出现此情况，因为由①可得， $d\text{-}daytime_1 \ge d - daytime_j$ 以及 $nighttime_1 \ge nighttime_k$ ，

若交换前司机 d_1 加班，即 $nighttime_k > d\text{-}daytime_1$ ，则必有，

$$nighttime_1 \ge nighttime_k > d\text{-}daytime_j,$$

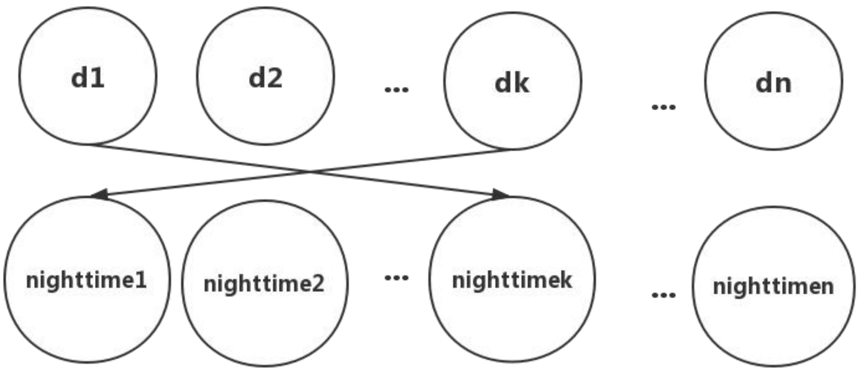
司机 d_j 交换前也必然加班。综上所述，情况 c 不存在。

d.　交换前司机 d_l 不加班而且司机 d_j 不加班

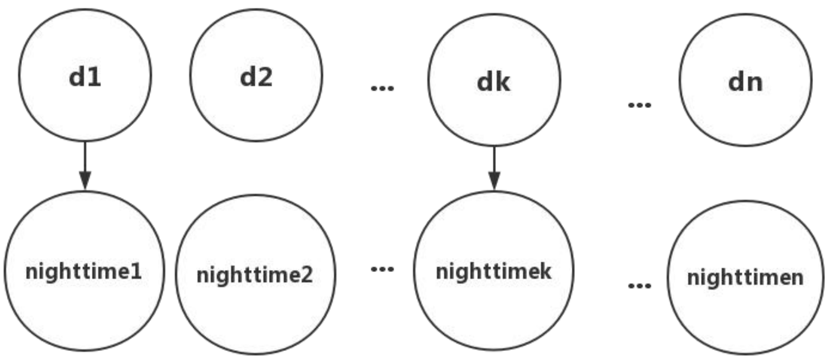
由①可得， $\text{d-daytime}_l \ge d - \text{daytime}_j$ 以及 $\text{nighttime}_l \ge \text{nighttime}_k$ ， 交换后，必然两者均不加班，易得 $\cos t' = \text{cost}$ 。

综上所述，情况 a,b,c,d 的分析可以得到 $\cos t' \le \text{cost}$ ，即无论 d_k 是否等于 d_j ，都可以说明贪心解是一个最优解，即本问题具有贪心选择性质。并且，每一步都按照贪心选择进行求解，最终可以全局的最优解(每一步都可以通过这样的交换得证)。

然后，下面分析比较简单的情况，当 $d_k = d_j$ 的时候，如图五所示。



图五 特殊的最优解中



图六 构造贪心解交换后

在此特殊情况下，每次交换都能得到两个最优匹配，多次交换后也可以得到贪心解，并同样能通过上述说明证明其贪心选择性质。

证明完毕。

4.2 最优子结构性质

当一个问题的最优解包含其子问题的最优解时，称此问题具有最优子结构性质。本巴士司机加班费问题的最优子结构性质也容易说明，具体如下：

利用 4.1 问题贪心选择性质证明用到的贪心解中，我们可以知道除去司机 1 匹配 $(d_1,nighttime_1)$ 后，得到子问题 $\{d_2,d_3,\ldots,d_n\}$ 与 $\{nighttime_2,nighttime_3,...,nighttime_n\}$ 匹配问题。且对于子问题，要证明匹配序列 $(d_2,nighttime_2),(d_3,nighttime_3),...,(d_n,nighttime_n)$ 是最优解即可。

证明过程：

假设 $(d_2,nighttime_2),(d_3,nighttime_3),...,(d_n,nighttime_n)$ 不是子问题最优解，那么存在一个子问题最优解为 $(d_2,nighttime_2'),(d_3,nighttime_3'),...,(d_n,nighttime_n')$ 。

对于这个子问题最优解，我们加上除去的司机 1 匹配 $(d_1,nighttime_1)$ 后，将得到问题一个最优解 $(d_1,nighttime_1),(d_2,nighttime_2'),(d_3,nighttime_3'),...,(d_n,nighttime_n')$ ，这样与前提贪心解 $(d_1,nighttime_1),(d_2,nighttime_2),(d_3,nighttime_3),...,(d_n,nighttime_n)$ 是最优解矛盾。因此，子问题的最优解为 $(d_2,nighttime_2),(d_3,nighttime_3),...,(d_n,nighttime_n)$ 。

证明完毕。

5 求解过程

首先将接收输入数据(司机总数 n、加班费的支出时间限 d、各个司机白天工作时间 $DAYTIME_i$ 以及夜间线路 $NIGHTTIME_i$)，并且记录处理后的数据序列：n 个司机晚上可工作但不加班时间 $(d\text{-}DAYTIME_i)$ 序列

$$(d\text{-}DAYTIME_1),(d\text{-}DAYTIME_2),...,(d\text{-}DAYTIME_n),$$

以及夜间线路序列

$$NIGHTTIME_1,NIGHTTIME_2,\ldots,NIGHTTIME_n\text{。}$$

接着，对上述两个序列进行排序(本设计采用快速排序)，根据$\left(\text{d-DAYTIME}_i\right)$的差值非增排序后的序列

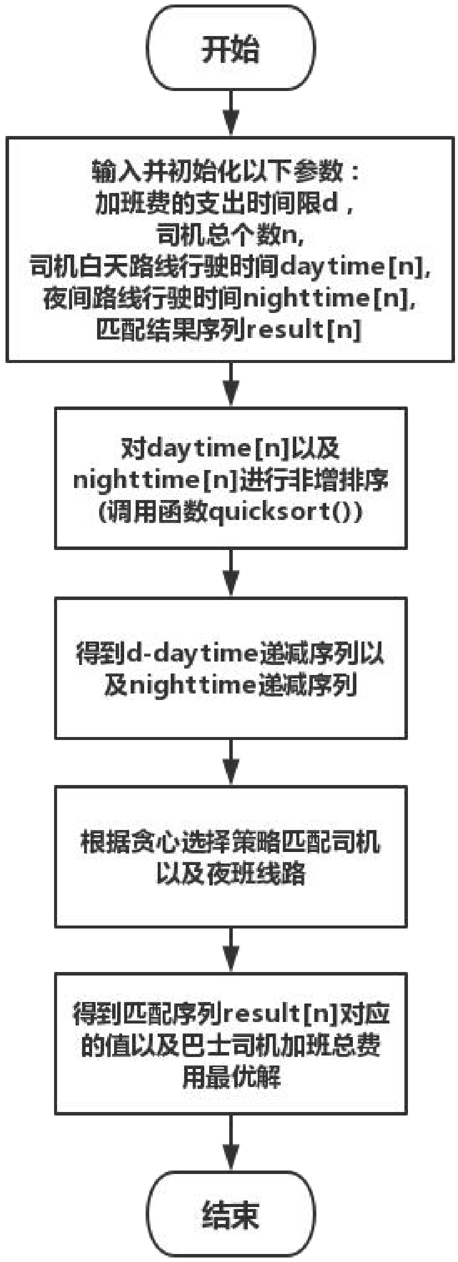
$$\left(\text{d-daytime}_1\right),\left(\text{d-daytime}_2\right),\ldots,\left(\text{d-daytime}_\text{n}\right),$$

以及夜间线路序列

$$\text{nighttime}_1,\text{nighttime}_2,\ldots,\text{nighttime}_\text{n}$$

排序的同时，记录输入时的对应司机序号，此处通过创建 Match 对象完成保存排序前序列对应元素的位置。

最后，根据贪心算法的贪心选择策略，优先匹配$\left(\text{d-daytime}_i\right)$数值大的司机与较长的夜间行驶线路$\text{nighttime}_\text{i}$，并得到问题在贪心算法求解得到的最优解$\cos t$以及对应的匹配结果序列$\left(\text{d}_1,\text{nighttime}_1\right),\left(\text{d}_2,\text{nighttime}_2\right),\ldots,\left(\text{d}_\text{n},\text{nighttime}_\text{n}\right)$。



图七 算法求解过程总流程图

6 算法复杂度及其推导过程

6.1 算法时间复杂度

对于本问题巴士司机的加班费来说，假如事先输入的司机白天工作线路与加班工作时间限 d 的差，是没有按照 d-$daytime_i$ 进行非增排序的话，程序需要调用 quicksort()方法对其进行排序；同理，输入的夜间线路工作顺序若没有降序排列也是需要调用 quicksort()进行排序。本课程设计采用的排序算法为快速排序，其平均时间复杂度为 O(nlogn)，平均性能最好，但是若输入序列有序或基本有序时，其时间复杂度将和冒泡排序一样，为 $O(n^2)$。

代码中采用递归方式实现快速排序，具体如下

```
while (i != j) {
        // 顺序很重要，要先从右边开始找
        while (a[j].getValue() <= temp.getValue() && i < j)
            j--;
        // 再找右边的
        while (a[i].getValue() >= temp.getValue() && i < j)
            i++;
        // 交换两个 Match 对象在数组中的位置
        if (i < j) {
            t.copy(a[i]);
            a[i].copy(a[j]);
            a[j].copy(t);
        }
        }
```

最优情况下，快速排序类似折半查找，将问题分为两个规模相等的子问题进行求解，推导过程如下，得到 O(nlogn)。

$$T(n)=2T(\frac{n}{2})+n,$$
$$T(n)=2T(2T(\frac{n}{4})+\frac{n}{2})+n=4T(\frac{\text{n}}{4})+2n,$$
$$\ldots$$
$$T(n)=nT(1)+(\log(n))\cdot n=O(n\cdot\log(n))$$
$$(\because T(1)=0)$$

最坏情况下，即当输入序列均为正序或完全逆序的时候，每次划分只得比上一次划分少一个元素的子序列。比较次数为

$$(n-1)+(n-2)+...+1=\frac{\text{n}(n-1)}{2}$$

得到$\text{O}(\text{n}^2)$。

一般情况下，

$$\text{T(n)=}\frac{1}{n}\sum_{k=1}^{n}(T(\text{k}-1)+T(n-k))+n=\frac{2}{n}\sum_{k=0}^{n-1}T(k)+n\ ,$$

由此[2]，可得算法时间复杂度为$\text{O}(\text{nlogn})$。

6.2 算法空间复杂度

从上面算法可知 ，用到了三个 Match 数组 ，分别为白天工作时间数组 daytime[n]，夜间工作行驶时间数组 nighttime[n],结果序列数组 result[n]。而 Match 对象含有两个字段，origin 以及 value，因此上述数组可看做数组，因此空间复杂性主要由此三个 Match[2][n]决定，为 O(3×2×n)，即$\text{O}(\text{n})$。因此，该算法的空间复杂性为$\text{O}(\text{n})$，可以应用到即使是小内存、低速度的计算机上。因此本算法有广阔的应用前景。

7 实验截图

7.1 实验测试用例一输入

第一行：加班费支出时间限 d=10，司机总数 n=3，

第二行：白天行驶时间序列(1,1),(2,8),(3,5)，

第三行：夜间线路时间(1,18),(2,10),(3,28)。

输出：

匹配序列(1,3),(2,2),(3,1)，

最优解为 40

图八 实验测试一

7.2 实验测试用例二输入

第一行：加班费支出时间限 d=15，司机总数 n=5，

第二行：白天行驶时间序列(1,2),(2,3),(3,77),(4,55),(5,18)，

第三行：夜间线路时间(1,3),(2,9),(3,58),(4,17),(5,15)。

输出：

匹配序列(1,3),(2,4),(3,1),(4,2),(5,5)

最优解为 182

图九 实验测试二

7.3 实验测试用例三输入

第一行：加班费支出时间限 d=20，司机总数 n=10，

第二行：白天行驶时间序列(1,15),(2,10),(3,25),(4,5),(5,17),(6,3),(7,18),(8,47),(9,50),(10,13)，

第三行：夜间线路时间(1,20),(2,5),(3,78),(4,1),(5,10),(6,55),(7,17),(8,8),(9,13),(10,8)。

输出：

匹配序列(1,3),(2,4),(3,1),(4,2),(5,5)

最优解为 218

图十 实验测试三

需要说明的是，首先，除了考虑白天工作时间没有超过 d 以外，还需考虑司机 i 在白天已经加班了的情况，此处在以上实验用例二和实验用例三可以体现到。因为这里 $\left(\text{d-daytime}_i\right)$ 为负号的时候，$\text{nighttime}_{\text{i}}-\left(\text{d-daytime}_i\right)$ 会自动加上白天就加班的费用，并且当 $\left(\text{d-daytime}_i\right)$ 小于零的时候，快速排序也会将此司机 i 放到非增序列末尾，从而匹配夜间线路最短的班次，体现出程序的鲁棒性。

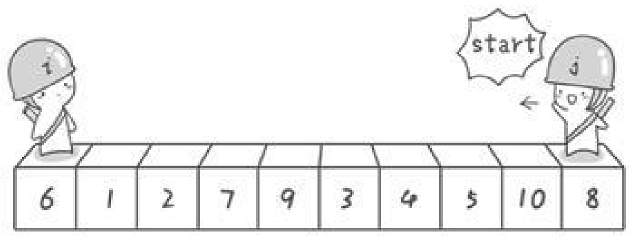
其次，本设计代码具有健壮性，当有非法输入的时候会抛出异常，并输出给用户进行调整。

8 设计小结

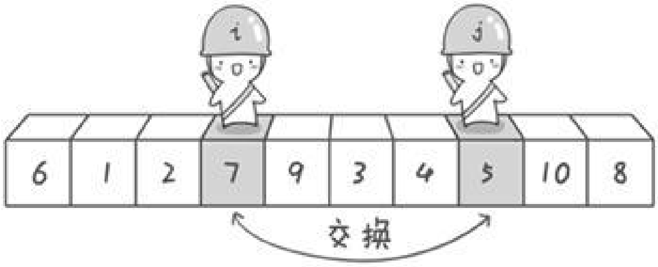
8.1 何唯

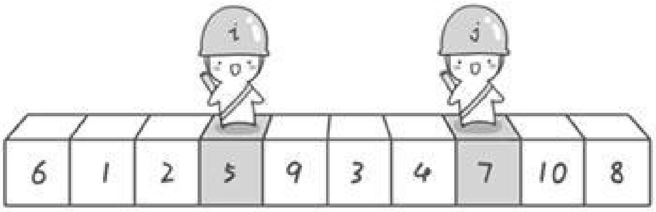
这里需要说明本设计贪心算法的实现算法复杂度十分依赖排序算法，这里使用了不稳定的快速排序，因为其适用范围比较广，总结其实现原理如下（详见本人博客 http://blog.csdn.net/erichhhhho/article/details/51066135）：

基本思想就是在序列中选择一个基准值，把小于基准值的元素移到右边，大于基准值的移去左边，然后一直递归下去。具体图示如下：

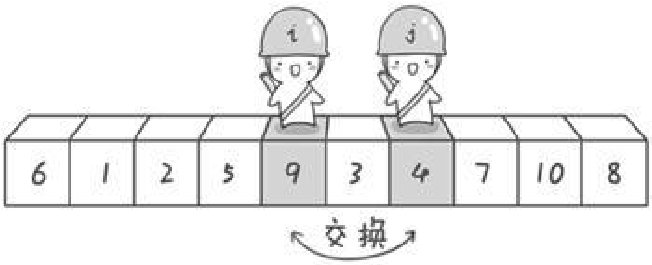


从序列两端开始探索，从序列右边找到第一个比基准值小的元素，再找到序列左边第一个比基准值大的元素，这里设置 i 为指向最左边的哨兵，j 为指向最右边的哨兵

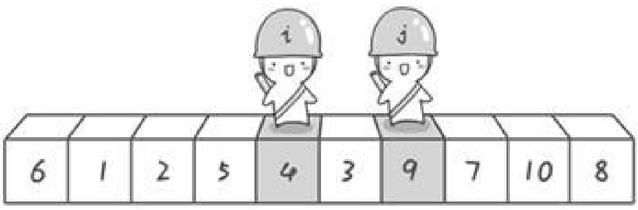


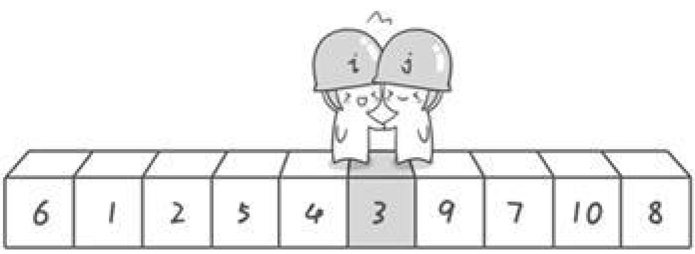


然后交换两个元素，

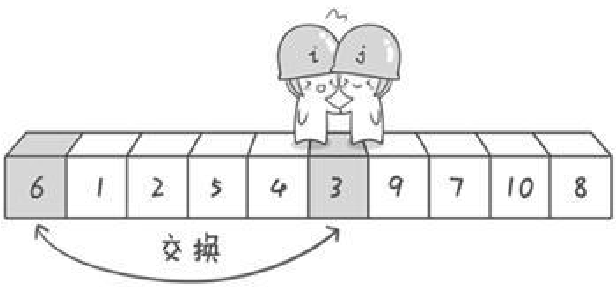


以此类推。

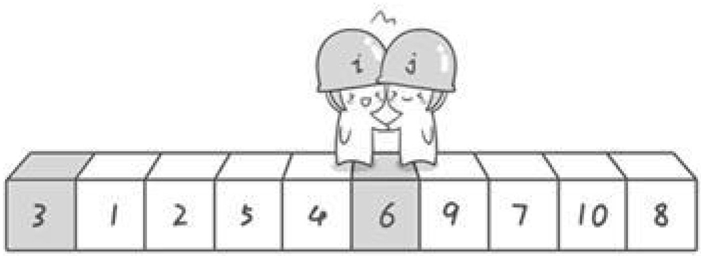




直到 i 烧饼与 j 烧饼相遇，就停止探索，



然后交换 i,j 烧饼所在元素和基准值,



这就完成一趟快排，接下来还要在基准值左边的区间和右边的区间分别递归进行下一趟快排，直到区间只有一个元素。

当然这里可以选择其他排序算法，比如堆排序(最坏以及最好的时间复杂度均为 O(nlogn) 但仍为不稳定排序算法)，归并排序(最坏以及最好的时间复杂度均为 O(nlogn) 且为稳定排序算法)。这里是本设计优化的一点建议和设想，可以根据一些用户输入偏好进行排序算法的使用。

完成本设计遇到的问题是贪心算法对于问题的正确性比较难证明，经过与老师的讨论后，自己结合课堂上的知识，很快就解决了，并能分析各种情况下的加班费用总和最优解。比如说，如何说明交换后得到的解不比所设的最优解大，一开始我只注意到简单的情况，即当 $d_k=d_j$ 的时候。但是其实要说明交换后也可以得到最优解，并不要求 $d_k=d_j$，其实只要是问题非首位的司机即可(首位的司机 (d-daytime_i) 最大)，即除了司机 1 外的司机都可以说明交换后解不变大。

得到的收获或则说教训就是，当面对证明问题不容易求解或者完成的时候不要轻易放弃，有时候不经意的思考也会得到很大的进展的。最后也感谢老师对于解题思路提供的意见，以及算法设计与分析这一段课程的学习。课程结束以后，希望自己也能学习更多的算法设计思想，增加自己的竞争力。

8.2 刘晓婷

9 各人在小组中的完成的工作

成员	负责完成的工作
何唯	➢ 负责算法设计思想的选择 ➢ 贪心算法正确性(问题的贪心选择性质以及最优子结构性质)的证明 ➢ 对于本设计问题的求解程序设计以及编码 ➢ 本设计报告文档内容的撰写 ➢ 统筹小组成员工作
刘晓婷	➢ 求解问题的叙述以及总结 ➢ 程序代码的调试以及测试 ➢ 程序算法复杂度分析 ➢ 部分设计报告文档撰写

10 自我评定成绩(按百分制)

成员	自我评定成绩
何唯	
刘晓婷	

参考文献

[1]王晓东. 算法设计与分析(第 3 版)[M]. 清华大学出版社，2015.10.

[2]Thomas H. Cormen.算法导论.机械工业出版社，2013.01.

[3]邹哲讷. 贪心算法及其应用[J]. 计算机光盘软件与应用,2015,(03):85-86.