数学建模经验分享

——以去年的比赛经验为例

汇报人　：　苏博

汇报日期：　2023 年 5 月 23 日（周二）

经验小谈：压缩

提前掌握（批量）压缩图片的方式

- 起因：需提交的文件夹有600M以上，但提交要求在约几十兆之内

- 步骤一：写代码或使用工具遍历文件夹压缩图片

```
% 批量压缩图片
mainDir='E:\B22100550001\第二问\第二问结果效果图\';
subDir=dir(mainDir);
for i=3:length(subDir)
    fprintf('********正在处理第%d个子文件夹********\n',i-2);
    path=strcat(mainDir,subDir(i).name);
    subsubDir=dir(path);
    for j=3:length(subsubDir)
        fprintf('****处理第%d个子文件夹的第%d个子子文件夹****\n',i-2,j-2);
        path_1=strcat(path,'\',subsubDir(j).name);
        pic_all=dir(path_1);
        if(length(pic_all)>2)
            for k=3:length(pic_all)
                path_2=strcat(path_1,'\',pic_all(k).name);
                A=imread(path_2);
                B=imresize(A,0.38);
                imwrite(B,path_2);
            end
        end
    end
end
```

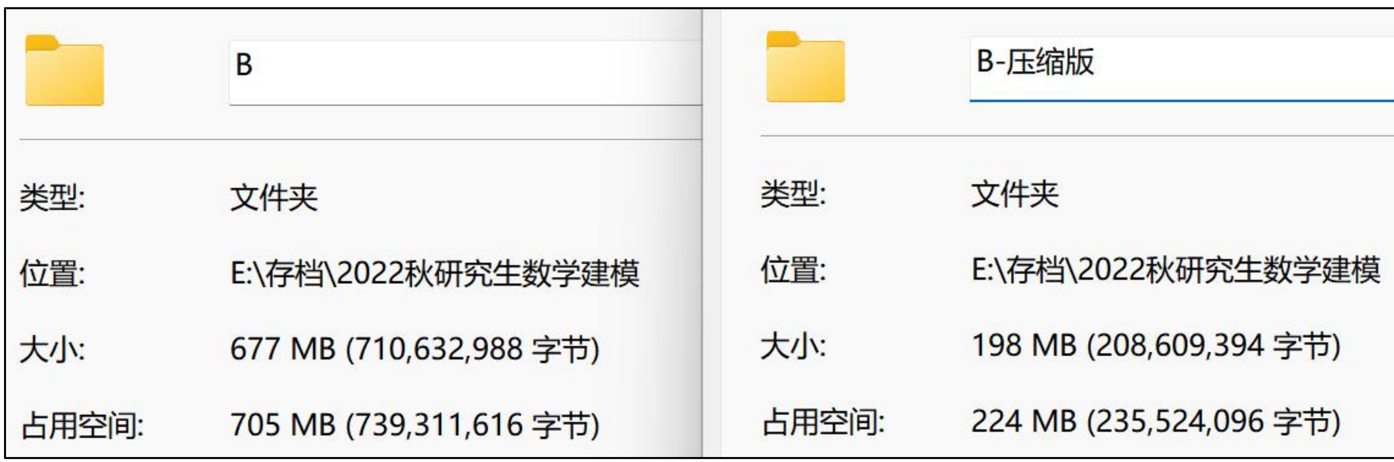
经验小谈：压缩

提前掌握（批量）压缩图片的方式

- 起因：需提交的文件夹有**600M**以上，但提交要求在**约几十兆之内**

- 步骤一：写代码或使用工具遍历文件夹压缩图片

效果图

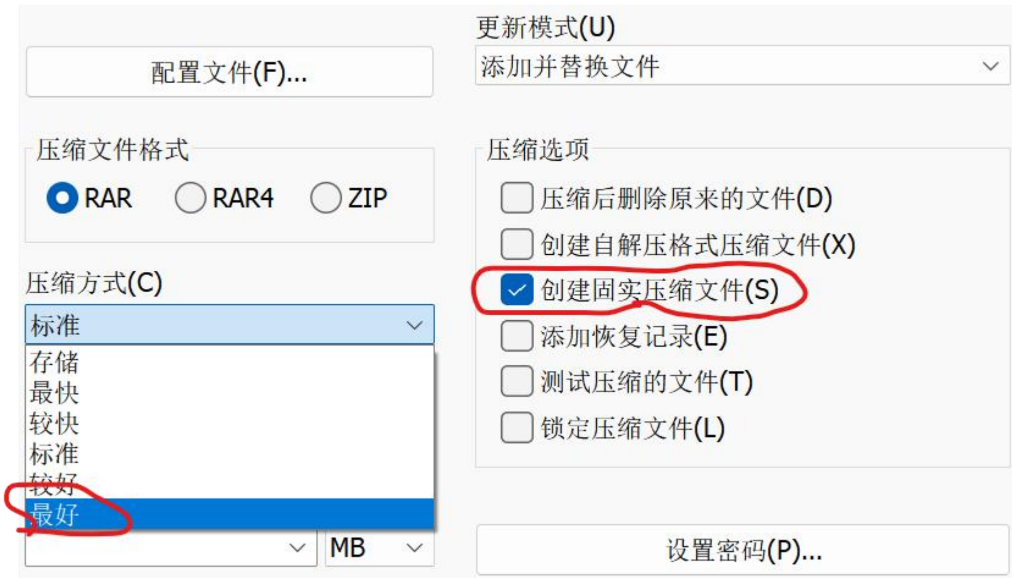


经验小谈：压缩

提前掌握（批量）压缩图片的方式

- 起因：需提交的文件夹有**600M**以上，但提交要求在约几十兆之内

- 步骤二：打压缩包时选择“最好”及“固实压缩”



经验小谈：压缩

提前掌握（批量）**压缩**图片的方式

- 起因：需提交的文件夹有**600M**以上，但提交要求在**约几十兆之内**

- 步骤二：打压缩包时选择“最好”及“固实压缩”

效果图

经验小谈：通宵

尽量不通宵，但做好最后一天通宵准备

1. 提前找好讨论的教室或酒店或宿舍等并完成相应预约工作。
2. 尽量在白天及晚上11点前以高效率完成工作，保证睡眠与精力。
3. 不要在前几天通宵，在最后一天可以通，但此时可能精力不足，剩下没完成的内容又是比较重要的内容，需提前想到这一点。
4. 尽量在前几天高效率地完成绝大部分工作，尽量别拖到ddl。

经验小谈：摘要

摘要最重要，需要好好写

1. 留足写摘要的时间
2. 摘要一般格式：
   - 第一段概括本文内容
   - 针对问题一，
   - xxxx(简述解决问题用的模型、方法、过程)
   - xxxx(简述最终得到的结果)
   - 针对问题二（三、四），xxxx
   - 综上所述，xxxx（这一段视情况而定，看是否需要）
3. 不要含糊其辞，要语句清楚明确
4. 可适当写一些套话。但要提到关键算法、模型、步骤

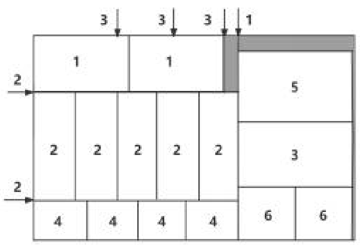
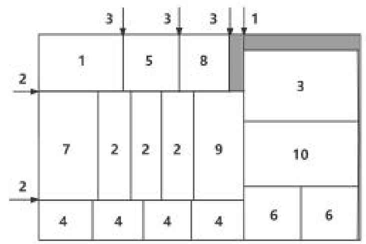
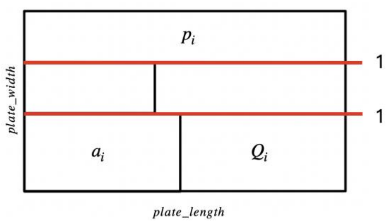
经验小谈：摘要

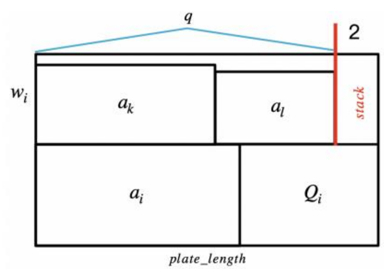
摘要最重要，需要好好写

针对问题一，要求建立混合整数规划模型，在满足生产订单需求和相关约束条件下，尽可能减少板材用量。本文以最小化板材用量为优化目标，建立了两种混合整数规划模型，提出了给定排样规则的生成算法、3H 排样生成算法、混合遗传算法等三种算法来对该排样优化问题进行求解，并针对各模型与算法的优缺点提出了对算法的优化思路，将给定排样规则的生成算法和 3H 排样生成算法结合起来，提高了板材利用率、降低了算法复杂度。基于各算法的复杂度、实际效果等特点，最终使用混合遗传算法来进行对该问题的求解。经过编程实现，数据集 A1、A2、A3、A4 在所求得对应排样方案下的板材利用率分别为

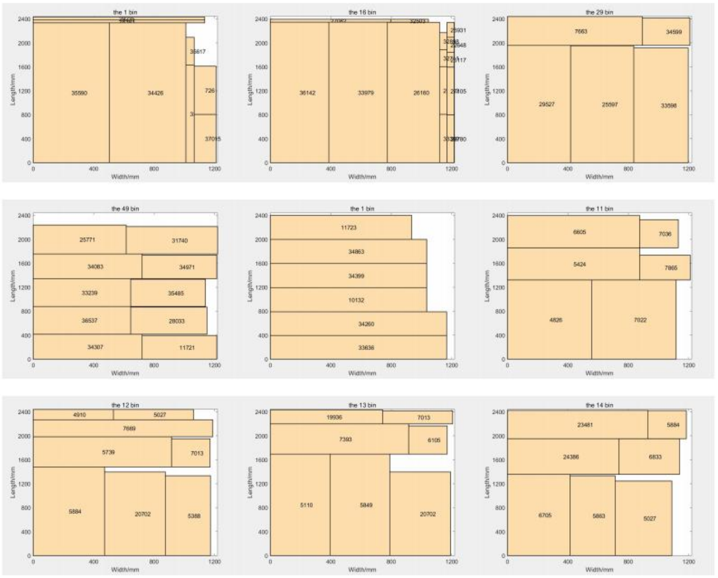
经验小谈：图表

涉及的**图表**需要好好画，尽量**简洁**又**好看**

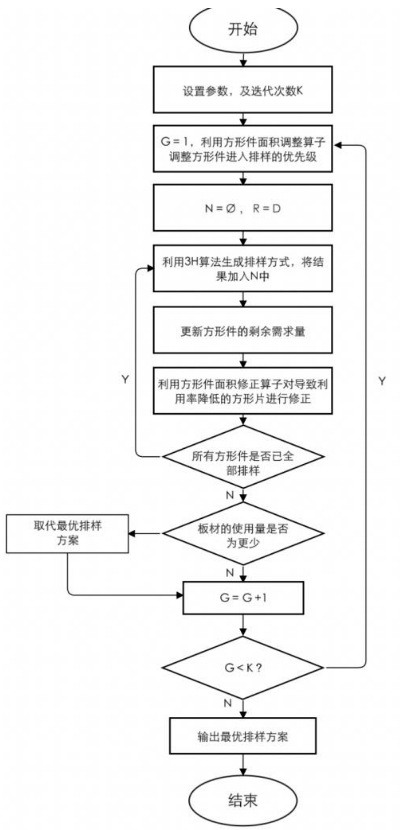


由于问题二求解所得效果图过多，故此处仅列出少部分效果图以作展示，全部效果图见附件。



经验小谈：图表

可适当放一些**流程图**，**step**型说明等



Step 1: 初始化设置要求的批次列表 *list* 为空，设置待处理的批次列表 *list_todo* 为每个订单独自组成一个批次进入该列表，并计算 *list_todo* 中每两个批次之间的距离，得到 *list_todo* 的距离矩阵 D；

Step 2: 若 *list_todo* 中有某一批与其他批的距离都是无穷大，则将其从 *list_todo* 中取出，并放入 *list* 中；

Step 3: 更新 D 并将 *list_todo* 中距离最小的两批合并为一批，并更新 D；

Step 4: 若 *list_todo* 中只有一批，则将其存入 *list*，程序结束。否则回到 *Step* 2。

由以上算法即可得到我们想要的订单组批方案。得到组批方案后，再对每一批的同种材质的产品项使用问题一的方法即可得到具体的排样方案与该方案所需的板材数量。

经验小谈：模型

多看一些相关文献，从中寻找思路与可用模型

别人的模型不一定能照搬，但至少可以借鉴

- $\beta_{k,j} \in \{0,1\}$, $k=1,\cdots,n$, $j=1,\cdots,n$: 其取值为 1 当且仅当栈 j 包含在条带 k 中，
- $\gamma_{l,k} \in \{0,1\}$, $l=1,\cdots,n$, $k=l,\cdots,n$: 其取值为 1 当且仅当条带 k 包含在板材 l 中，
- $\delta_{l,i,j} \in \{0,1\}$, $l=1,\cdots,n-1$, $i=l+1,\cdots,n$, $j=l,\cdots,i-1$: 其取值为 1 当且仅当 $\alpha_{j,i}=1$ 且 $\gamma_{l,j}=1$。

于是可得问题一的混合整数线性规划模型如下 [1]:

$$\min \sum_{l=1}^{n} \gamma_{l,l}$$

$$\text{s.t.} \begin{cases} \sum_{j=1}^{i} \alpha_{j,i} = 1,\ \forall i=1,\cdots,n, \\ \sum_{i=j+1}^{n} \alpha_{j,i} \le (n-j)\alpha_{j,j},\ \forall j=1,\cdots,n-1, \\ \alpha_{j,i}=0,\ \forall j=1,\cdots,n-1, \forall i>j \text{且满足} l_i \ne l_j \text{或} w_i+w_j>W, \\ \sum_{k=1}^{n} \beta_{k,j} = \alpha_{j,j},\ \forall j=1,\cdots,n, \\ \sum_{i=j}^{n} w_i\alpha_{j,i} < \sum_{i=k}^{n} w_i\alpha_{k,i} + (W+1)(1-\beta_{k,j}),\ \forall k=2,\cdots,n,\ \forall j=1,\cdots,k-1, \\ \sum_{i=j}^{n} w_i\alpha_{j,i} \le \sum_{i=k}^{n} w_i\alpha_{k,i} + W(1-\beta_{k,j}),\ \forall k=1,\cdots,n-1,\ \forall j=k+1,\cdots,n, \\ \sum_{j=1}^{n} l_j\beta_{k,j} \le L\beta_{k,k},\ \forall k=1,\cdots,n, \\ \sum_{l=1}^{k} \gamma_{l,k} = \beta_{k,k},\ \forall k=1,\cdots,n, \\ \sum_{i=l}^{n} w_i\gamma_{l,i} + \sum_{i=l+1}^{n} w_i \sum_{j=l}^{i-1} \delta_{l,i,j} \le W\gamma_{l,l},\ \forall l=1,\cdots,n-1, \\ \alpha_{j,i}+\gamma_{l,i}-1 \le \delta_{l,i,j} \le (\alpha_{j,i}+\gamma_{l,j})/2,\ \forall l=1,\cdots,n-1,\ \forall i=l+1,\cdots,n,\ j=l,\cdots,i-1, \\ \sum_{k=l+1}^{n} \gamma_{l,k} \le (n-l)\gamma_{l,l},\ \forall l=1,\cdots,n-1, \\ \alpha_{j,i} \in \{0,1\},\ j=1,\cdots,n,\ i=j,\cdots,n, \\ \beta_{k,j} \in \{0,1\},\ k=1,\cdots,n,\ j=1,\cdots,n, \\ \gamma_{l,k} \in \{0,1\},\ l=1,\cdots,n,\ k=l,\cdots,n, \\ \delta_{l,i,j} \in \{0,1\},\ l=1,\cdots,n-1,\ i=l+1,\cdots,n,\ j=l,\cdots,i-1. \end{cases}$$

但注意到该模型使用的变量数为 O(n^3)，约束数也为 O(n^3)，其中 n 为需求的产品项的数目。其数量级太大，极难求解，故此我们转而考虑更为简单的模型。

The complete ILP for the (unrestricted) 3-stage 2BP looks as follows:

$$\text{minimize} \quad \sum_{l=1}^{n} \gamma_{l,l} \tag{16}$$

$$\text{subject to} \quad \sum_{j=1}^{i} \alpha_{j,i} = 1, \quad \forall i=1,\ldots,n, \tag{17}$$

$$\sum_{i=j+1}^{n} \alpha_{j,i} \leqslant (n-j)\alpha_{j,j}, \quad \forall j=1,\ldots,n-1, \tag{18}$$

$$\alpha_{j,i}=0, \quad \forall j=1,\ldots,n-1,\ \forall i>j \mid w_i \ne w_j \vee h_i+h_j>H, \tag{19}$$

$$\sum_{k=1}^{n} \beta_{k,j} = \alpha_{j,j}, \quad \forall j=1,\ldots,n, \tag{20}$$

$$\sum_{i=j}^{n} h_i\alpha_{j,i} < \sum_{i=k}^{n} h_i\alpha_{k,i} + (H+1)(1-\beta_{k,j}), \quad \forall k=2,\ldots,n,\ \forall j=1,\ldots,k-1, \tag{21}$$

$$\sum_{i=j}^{n} h_i\alpha_{j,i} \leqslant \sum_{i=k}^{n} h_i\alpha_{k,i} + H(1-\beta_{k,j}), \quad \forall k=1,\ldots,n-1,\ \forall j=k+1,\ldots,n, \tag{22}$$

$$\sum_{j=1}^{n} w_j\beta_{k,j} \leqslant W\beta_{k,k}, \quad \forall k=1,\ldots,n, \tag{23}$$

$$\sum_{l=1}^{k} \gamma_{l,k} = \beta_{k,k}, \quad \forall k=1,\ldots,n, \tag{24}$$

$$\sum_{i=l}^{n} h_i\gamma_{l,i} + \sum_{i=l+1}^{n} h_i \sum_{j=l}^{i-1} \delta_{l,i,j} \leqslant H\gamma_{l,l}, \quad \forall l=1,\ldots,n-1, \tag{25}$$

$$\alpha_{j,i}+\gamma_{l,j}-1 \leqslant \delta_{l,i,j} \leqslant (\alpha_{j,i}+\gamma_{l,j})/2, \quad \forall l=1,\ldots,n-1, \quad \forall i=l+1,\ldots,n,\ j=l,\ldots,i-1, \tag{26}$$

$$\sum_{k=l+1}^{n} \gamma_{l,k} \leqslant (n-l)\gamma_{l,l}, \quad \forall l=1,\ldots,n-1, \tag{27}$$

$$\alpha_{j,i} \in \{0,1\}, \quad j=1,\ldots,n,\ i=j,\ldots,n, \tag{28}$$

$$\beta_{k,j} \in \{0,1\}, \quad k=1,\ldots,n,\ j=1,\ldots,n, \tag{29}$$

$$\gamma_{l,k} \in \{0,1\}, \quad l=1,\ldots,n,\ k=l,\ldots,n, \tag{30}$$

$$\delta_{l,i,j} \in \{0,1\}, \quad l=1,\ldots,n-1,\ i=l+1,\ldots,n. \tag{31}$$

经验小谈：模型

多看一些相关文献，从中寻找思路与可用模型

别人的模型不一定能照搬，但至少可以借鉴

第一问修改后的模型：

$$\min \sum_{j=1}^{K} x_j$$

$$\text{s.t.} \begin{cases} \sum_{j=1}^{K} a_{ij} x_j = d_i,\ i = 1, \cdots, n, \\ x_j \in \mathbb{N},\ j = 1, \cdots, K. \end{cases}$$

第二问模型：

（借鉴了之前非常复杂的模型的思想）

$$\min \sum_{i=1}^{t} q_i$$

$$\text{s.t.} \begin{cases} \sum_{j=1}^{t} \gamma_{j,i} = 1,\ \forall i = 1, \cdots, n, \\ \sum_{i=1}^{t} \gamma_{j,i} m_i d_i \le N_{max},\ \forall j = 1, \cdots, t, \\ \sum_{i=1}^{t} \gamma_{j,i} \sum_{k=1}^{m_i} l_{ik} w_{ik} d_{ik} \le S_{max},\ \forall j = 1, \cdots, n, \\ \gamma_{j,i} \in \{0, 1\}, \forall i, j = 1, \cdots, n. \end{cases}$$

经验小谈：常数

在一些常数的给定上，或可尝试给出取**不同值**时的情况并可做出相应图像以进行分析并取较优解等。

- 有了以上两个相似度的定义之后，我们考虑将以上两个相似度综合起来考虑，并给定一个常数 $\lambda \in \{0,1\}$，定义最终的相似度 $\rho(i,j)$ 如下：

$$\rho(i,j) = \lambda\rho_1(i,j) + (1-\lambda)\rho_2(i,j).$$

2. 其次，对两个订单之间的最终的相似度的定义中，采用了两种相似度的线性组合的形式，其中使用了常数 $\lambda \in \{0,1\}$。一方面我们不知道那两种相似度对于最终所需板材数量的影响程度哪个大，且时间有限，另一方面，问题一的求解方案本身也只是近似最优解，并没有严格地求出精确的最优解，故此我们直接将 λ 的值设置为 0.5 来进行求解。或许可以尝试令 λ 取 0 到 1 之间的不同的值，并得到相对应的所需板材数，进而可做出所需板材数与 λ 值之间的曲线，找出其最小值点，令 λ 取该值以得到更优的结果。

经验小谈：文献

可放置一些中文及英文的参考文献

其中至少有一些是确实参考了思想

参考文献

[1] Puchinger, Jakob; Raidl, Günther R, Models and algorithms for three-stage two-dimensional bin packing, European J. Oper. Res, 183(3):1304-1327, 2007.

[2] Hifi, Mhand; Roucairol, Catherine, Approximate and exact algorithms for constrained (un) weighted two-dimensional two-staged cutting stock problems, J. Comb. Optim. 5(4):464-494, 2001.

[3] Hifi, Mhand; M'Hallah, Rym; Saadi, Toufik, Algorithms for the constrained two-staged two-dimensional cutting problem, INFORMS J. Comput, 20(2):212-221, 2008.

[4] 陈秋莲. 二维剪切下料问题的三阶段排样方案优化算法研究 [D]. 华南理工大学,2016.

[5] 刘睿, 严玄, 崔耀东, 矩形件三阶段带排样问题的遗传算法 [J], 计算机工程与应用, 46(33):221-224, 2010.

[6] Chen Q L, Chen Y, Cui Y D, et al, A heuristic for the 3-staged 2D cutting stock problem with usable leftover, 2015 International Conference on Electrical, Automation and Mechanical Engineering, 770-773, 2015.

[7] G.F. Cintra, F.K. Miyazawa, Y. Wakabayashi, E.C. Xavier, Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation, European Journal of Operational Research, 191(1):61-85, 2008.

[8] 陈秋莲, 宋仁坤, 崔耀东, 考虑余料价值的三阶段二维剪切下料算法 [J]. 图学学报,38(01):10-14, 2017.

[9] 扈少华, 武书彦, 潘立武, 基于两段排样方式的矩形件优化下料算法 [J], 图学学报, 39(01):91-96, 2018.

[10] 杨卫波, 王万良, 张景玲, 赵燕伟, 基于遗传模拟退火算法的矩形件优化排样 [J], 计算机工程与应用, 52(07):259-263, 2016.

[11] Elsa Silva, Filipe Alvelos, J.M. Valério de Carvalho, An integer programming model for two-

$$
\begin{aligned}
&\text{minimize} && \sum_{l=1}^{n} \gamma_{l,l} && (16)\\
&\text{subject to} && \sum_{j=1}^{i} \alpha_{j,i} = 1, \quad \forall i = 1,\dots,n, && (17)\\
& && \sum_{i=j+1}^{n} \alpha_{j,i} \leqslant (n-j)\alpha_{j,j}, \quad \forall j = 1,\dots,n-1, && (18)\\
& && \alpha_{j,i} = 0, \quad \forall j = 1,\dots,n-1, \ \forall i > j \mid w_i \neq w_j \vee h_i + h_j > H, && (19)\\
& && \sum_{k=1}^{n} \beta_{k,j} = \alpha_{j,j}, \quad \forall j = 1,\dots,n, && (20)\\
& && \sum_{i=j}^{n} h_i \alpha_{j,i} < \sum_{i=k}^{n} h_i \alpha_{k,i} + (H+1)(1-\beta_{k,j}), \quad \forall k = 2,\dots,n, \ \forall j = 1,\dots,k-1, && (21)
\end{aligned}
$$

$$
\min \quad z = \sum_{j=1}^{K} \delta_j x_j
$$

$$
s.t. \begin{cases} \sum_{j=1}^{K} a_{ij} x_j \geq d_i, & i = 1,2,\cdots,m \\ x_j \in \mathrm{N}, \ j = 1,2,\cdots,K \end{cases}
$$

经验小谈：代码

把控好时间，提前规划好大概**每天做些什么**并尽量实现
可以优先给出一个小问的思路或者方法等赶紧写**代码**

- 代码实现比较麻烦
- 调各种莫名其妙的bug？会很费时间

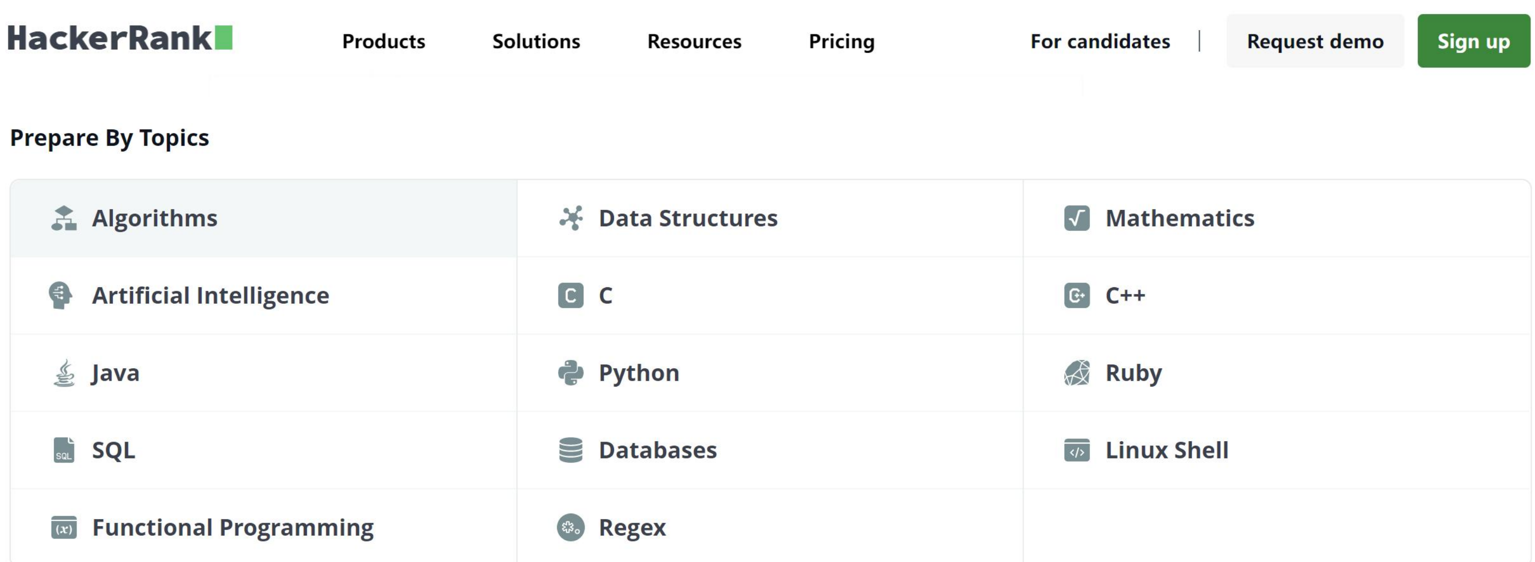
需要提前知道一些经典**算法**，并有能力进行简单**实现**

- 包括一些传统算法，如排序、DFS、动态规划等
- 也包括一些新型算法，如机器学习深度学习领域的部分经典算法

经验小谈：代码

这里推荐几个**代码算法**相关网站(没有收广告费啊)：

- Hackerrank: https://www.hackerrank.com/ 各种代码题



经验小谈：代码

这里推荐几个**代码算法**相关网站(没有收广告费啊)：

- 力扣：https://leetcode.cn/problemset/all/ 各种代码题

经验小谈：代码

这里推荐几个**代码算法**相关网站(没有收广告费啊)：

- OI Wiki : https://oi-wiki.org/ 一些常见算法的简要介绍

OI Wiki

搜索

OI-wiki/OI-wiki ☆14.5k 2.9k

简介 比赛相关 工具软件 语言基础 算法基础 搜索 动态规划 字符串 数学 数据结构 图论 计算几何 杂项 专题 关于 Hulu

算法基础
- 冒泡排序
- 插入排序
- 计数排序
- 基数排序
- 快速排序
- 归并排序
- 堆排序
- 桶排序
- 希尔排序
- 锦标赛排序
- tim排序
- 排序相关 STL
- 排序应用

前缀和 & 差分
二分
倍增
构造

希尔排序

本页面将简要介绍希尔排序。

定义

希尔排序（英语：Shell sort），也称为缩小增量排序法，是 插入排序 的一种改进版本。希尔排序以它的发明者希尔（英语：Donald Shell）命名。

过程

排序对不相邻的记录进行比较和移动：

1. 将待排序序列分为若干子序列（每个子序列的元素在原始数组中间距相同）；
2. 对这些子序列进行插入排序；
3. 减小每个子序列中元素之间的间距，重复上述过程直至间距减少为 1。

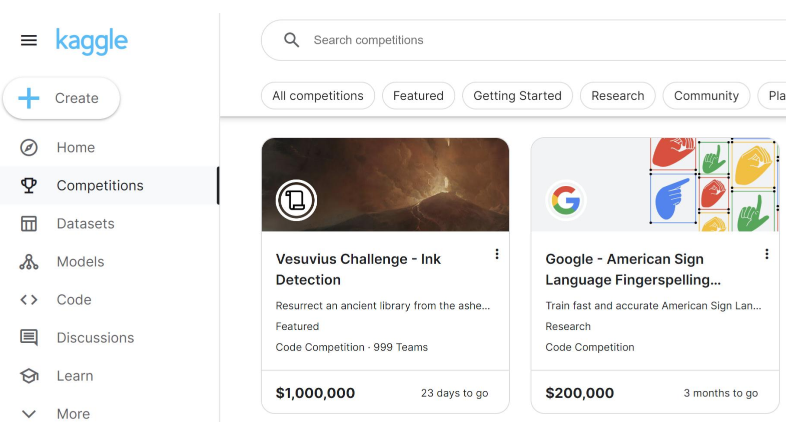
目录
- 定义
- 过程
- 性质
  - 稳定性
  - 时间复杂度
  - 空间复杂度
- 实现
- 参考资料与注释

经验小谈：代码

这里推荐几个**代码算法**相关网站(没有收广告费啊)：

一些在线的数据分析比赛的网站：

- Kaggle : https://www.kaggle.com/
- 天池：https://tianchi.aliyun.com/competition/activeList



经验小谈：好处

保研可加分： https://math.nankai.edu.cn/2022/0830/c5598a470285/page.htm

2022年数学学院推免细则公示

2022-08-30

综合排名成绩=BCD类课程平均学分绩×95%+综合素质成绩×5%

全国及国际数学建模竞赛	特等奖	**15**
	一	**12**
	二	**10**
天津市数学建模竞赛	一	**7**
	二	**5**
国家级其它奖	二等奖以上	**5**
省部级其它奖	二等奖以上	**3**

经验小谈：好处

本科生及研究生奖学金可加分：

可获得许多奖金

- 赛事组织方可能有奖金
- 国家奖学金、公能奖学金、学业优秀奖学金等
- 校友捐赠类的各种奖学金
- 陈省身奖学金

经验小谈：好处

对个人**能力**有一定提升

尤其是对**统计**、**金融**相关专业及

打算从事**算法研发**、**数据分析**岗位的同学

谢 谢 观 看