

关于 RM 夏令营机器人的技术报告大纲(5 组)

组员及分工:

一、需求分析 (abstract)

分析比赛题目,例如需要从哪一个方向抓取砖块,是否设计存储机构,算法部分如何配合,算法部分如何处理第一二阶段,如何设计 ui,是否需要添加巡线,避障,状态及等功能。

分析题目后,关于机器人的设计,我们组的成员提出了四种方案:

- 1.夹子夹起砖块后翻到传送带上储存,同时外加吸盘用于搭建和拆除;
- 2.利用摩擦轮将砖块“吸”入车体储存;
- 3.小车正面装一可旋转的垂直于地面的圆盘,圆盘上两个夹子均可以夹取和储存积木;
- 4.利用楔子和摩擦轮两个相向的力将积木分离并存入体内。

考虑到实现难度、稳定性和实用性等因素,我们采用了第一种方案;

后续我们又进行了细节上的讨论,确定了从长面抓取砖块(因为这样能减少转向的操作),吸盘和夹子配合使用(便于搭建和拆除城堡),独立气动与零件系统双备份等细节。

而算法方面,考虑到实现难度和数据范围,小组成员打算采取分类+深度优先搜索的方法来计算出第一阶段铺设的地基颜色和路线。第二阶段则留待后续考虑。

二 所需技术点,关键词

对所使用的技术点及关键词进行描述

机械:

*系统双备份:车体所有的可活动零件都有电子限制和触碰开关两种校对方式。气动气阀有分气阀和总控两种控制气动方面甚至还有物理的单向阀来做最后的备用方案。

*系统可靠性:所有的机械结构都尽力向减少零件的方向发展,以减少可动,增加可调的设计方向进行制作,力求提高适应性和可靠性。

嵌入式:

*半自动化:将复杂的操作简化为一键式操作,使效率得到极大的提升,同时保留基本操作键位,作为备用方案,提高系统的稳定性。

*PID:所有电机均运用嵌套闭环控制,稳。

算法:

*深度优先搜索:朴素的枚举铺设路线的方法。

*多线程:多于一个线程同时进行运算,利用空间换取时间,在配置较高的电脑上,能有效地缩短运算时间。

三 总体方案

我们使用一块电池和一块中控板,来控制总共 6 枚 3508 电机。除了车体驱动使用了 4 枚之外,快速夹取结构(前端夹子)与吸盘抬起机构所需要的扭力最大,所以在这些地方也使用了 3508 电机。

前端的夹取结构除了 3058 之外,还有两个金属齿轮舵机来配合连杆进行夹取方块的工作。车体前方还配备瞄准器和对位板来配合夹取。

吸盘平台则使用 3508 配合同步带来进行高度调节。吸盘模块还有一个 2206 来进行角度的调整。以完成更加高级的城堡搭建。因为吸盘与夹子的位置是对齐的,所以可以通过夹子来获取仓库的方块。

吸盘控制系统由真空泵，总控电磁阀（直下式）和 4 个小型分控电磁阀组成。期间，机器人可以通过关闭不必要的吸盘来提升吸力。值得一提的是，经过设计的气路可以在分控电磁阀全部失效的情况下继续使用。当然效果会大打折扣，但是依然可以完成搭建工作。

仓库模块则使用了定制的同步带与 2006 驱动。方块被放置在大致仓库入口后，斜坡会把它定位到正确的地方。之后同步带就会把它输送到仓库后方进行储存。最大储存数量为 4。

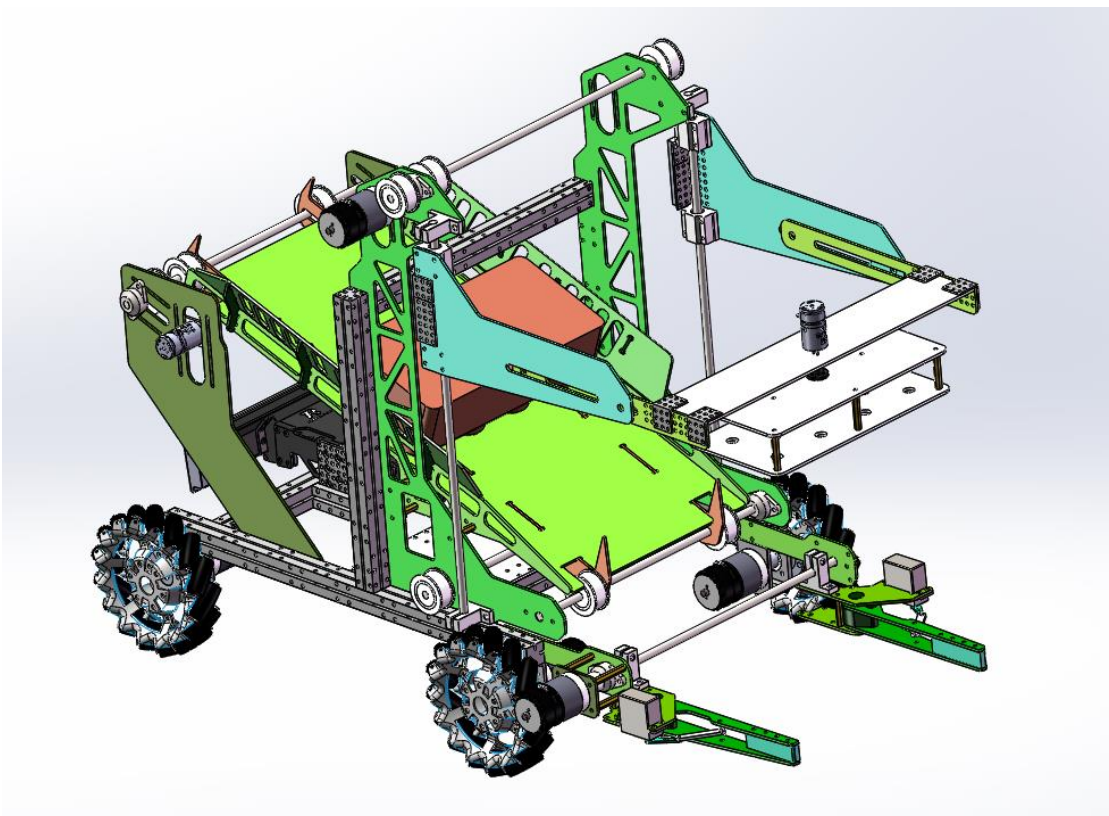
图传被定位在仓库的上方。角度经过调整图传可以清晰地看清所有机械原件的工作状态，并保留清晰的视野。

四 各模块方案

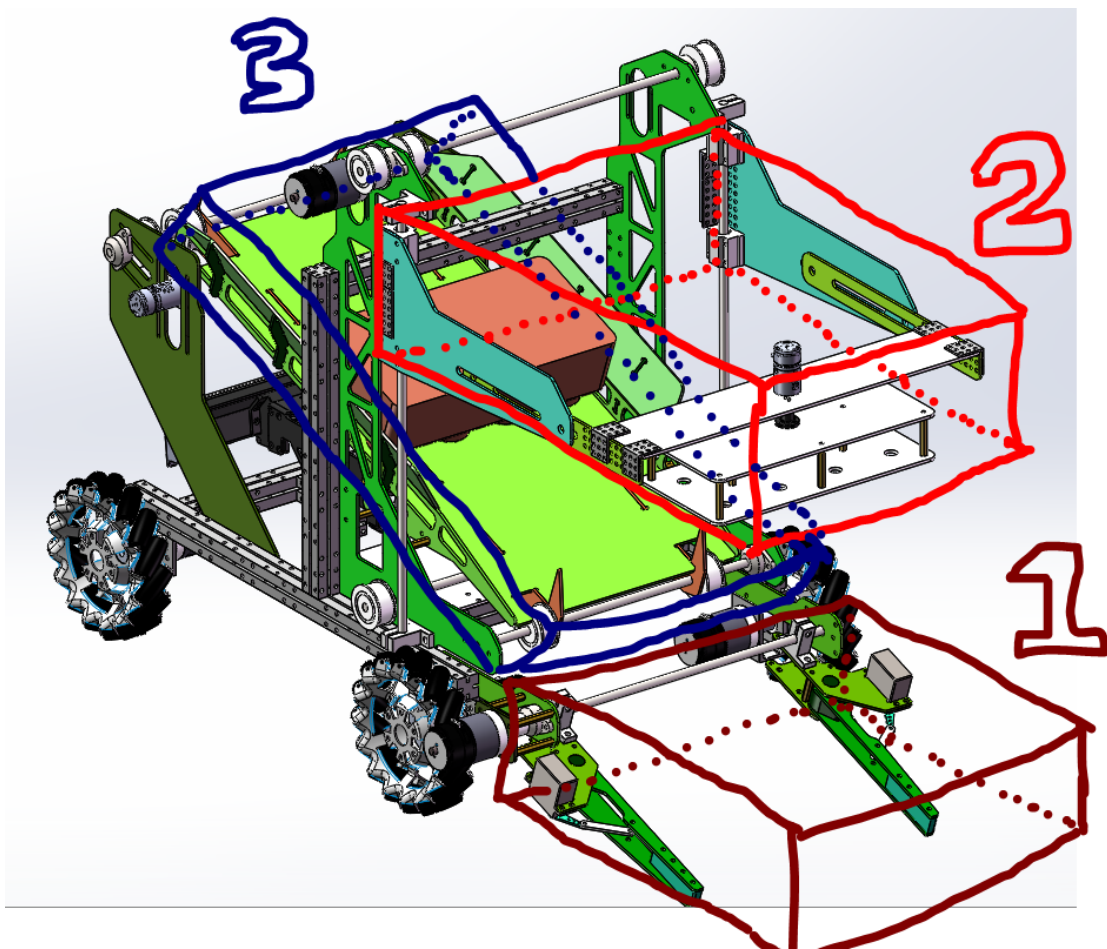
- 1、机械部分出结构连接草图，介绍图纸的整体布局，各模块排布（设计者，及分工）。
- 2、嵌入式写出逻辑框图，然后分模块介绍工作，比如电磁阀通信，can 通信，调 pwm，调 pid 最好按照你们的逻辑顺序来做，从小模块到整体。硬件的写下硬件框图，搞清楚哪跟线连哪根线，从底盘到云台怎么走线。以及搞清楚板子的接口情况
- 3、算法部分写出流程图，思路和 UI 设计图

这一部分介绍各模块的方案框架，写清楚谁提出了什么方案，详细写出谁做了哪些工作。

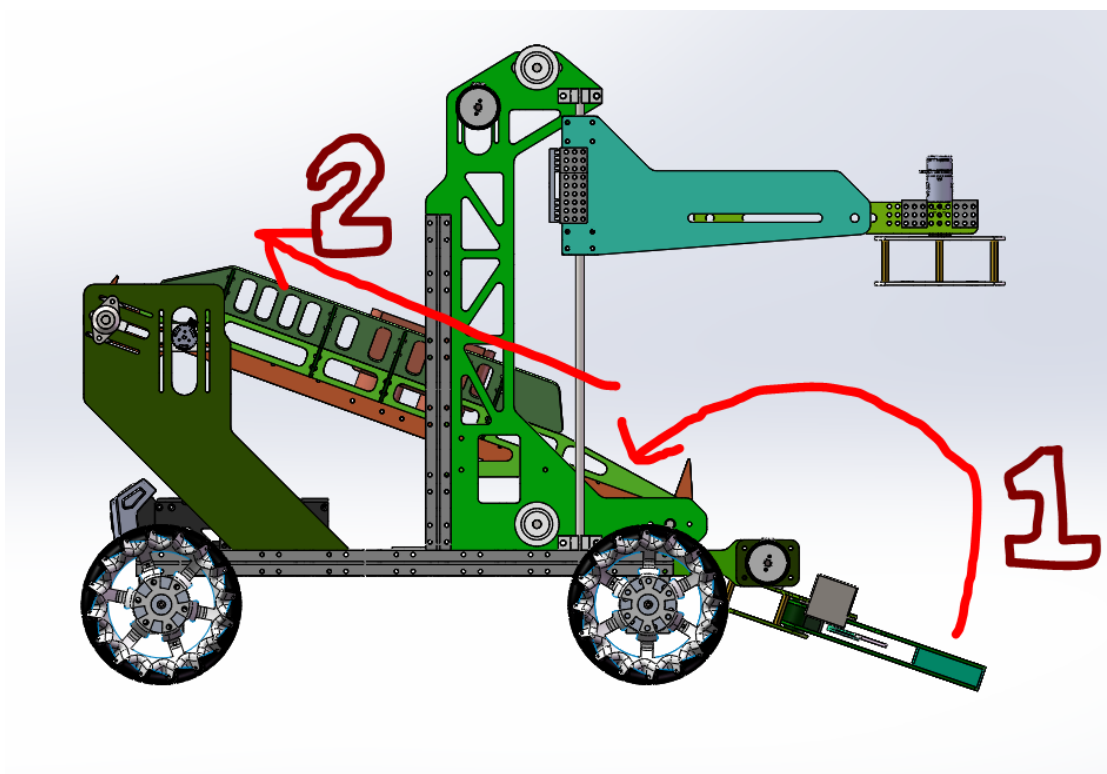
1. 机械：



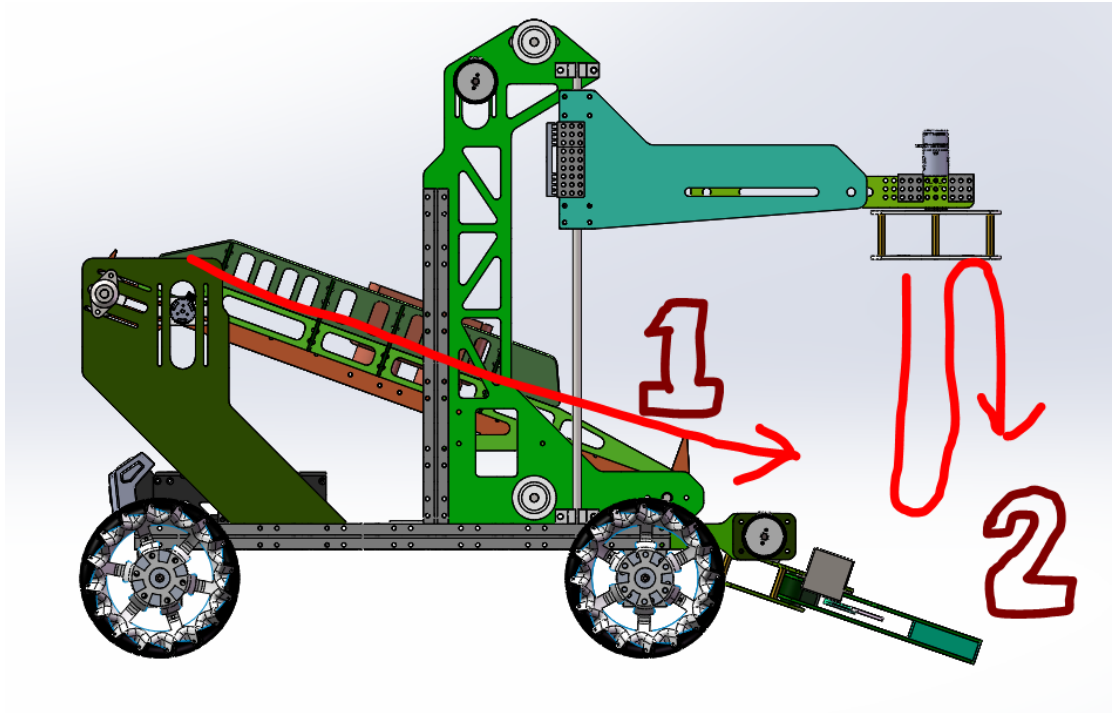
总装图



模块分解：1，前置夹取放置系统 2，精准城堡搭建系统 3，块存储系统



块收集夹取示意图



块取出搭建示意图

车体的前夹取机构的概念由欧阳曜钧提出，曹瑞翔等人经过讨论改良。并投票通过。

升降平台的概念由曹瑞翔提出，全组投票通过后并改良。

吸盘结构原本是由曹瑞翔提出的机械手，后期被欧阳曜钧提出的吸盘替代。

气路系统由欧阳曜钧设计制作。

仓库系统为曹瑞翔提出的斜坡，全组经过投票通过并改良。

车体结构强度不足的问题由葛鸿与梁德麟于热身赛发现，并改良。

传送带张紧力不足的问题由葛鸿发现，并由曹瑞翔解决。

仓库对位不齐的问题解决。

零件购置由曹瑞翔与欧阳曜钧共同负责。

工作台的整理，零件的分类与工具分方放，整体问题的记录，板材切割与二次加工由刘一思负责。因为他的工作，我们的工作十分顺畅。可以说他是我们的加工担当。

2. 嵌入式



嵌入式的工作由张思成负责。

01.首先，底盘主要采用修改后的官方函数进行控制。

02.夹取与放置主要用前方的机械臂完成，机械臂的纵向角度由电机 M3508 P16 进行控制，横向角度即张开的角度由一堆舵机进行控制。电机的控制利用 angle 与 speed 的嵌套闭环控制。

03.仓库的存储用的传送带利用 M2006 进行控制，利用 PID 闭环稳定转动速度。

04. 吸取装置由吸盘与升降装置以及气泵组成，吸盘由一 M2006 P35 电机控制角度，利用了 PID 算法对输入电机的电流进行角度与速度嵌套的闭环控制，实现了在 0 度与 90 度两种方向之间稳定切换。气泵及放气气阀分别由继电器进行控制，升降装置由 M3508 P19 电机进行控制，同样利用 PID 分别实现对位置与速度的闭环控制。同时利用上下安装限位开关返回的 IO 值控制吸盘的行程。

特点 1，半自动化：将复杂的操作简化为一键式操作，如一键“夹取+存储+归位”，使效率得到极大的提升，同时保留基本操作键位，作为备用方案，提高系统的稳定性。

特点 2，稳：所有电机均运用嵌套闭环控制，先由 angle 的偏差值计算出所需的 speed，再由实际 speed 与计算的 speed 值的偏差计算出发送给电机的电流，从而使控制非常的稳。

3. 算法：

对于每种颜色的四座城堡进行两两配对，从一个点开始搜索，连接至对应的另一个点为止，当所有点均连接且铺满棋盘时结束程序。另外将 81 种配对情况设为多个 thread，进行多线程的操作。最后编译生成 exe，通过命令行传参实现与 UI 的对接。（梁德麒；张思成指导完成多线程）

UI：使用 html 作为前端，ajax 传参给服务器调用 exe 程序进行运算（张思成）

五 理论分析 (analysis)

机械上，作简单的动力选型，比如电机功率，气缸参数，根据场地需求，给出轮距，轴距等参数。

算法，可行性分析，及运算速度。

嵌入式，给出控制参数。

写出遇到的困难和问题，重点写出走过的弯路，例如没有进行负载的估计，导致动力选型不合理等（介绍各部分任务的完成者）

1. 机械：

零件购买时，因为缺乏检查，所以淘宝买少零件（同步轮）而无法在热身赛中使用仓库模块。

由于使用真空泵，发放的电磁阀无法使用导致电磁阀选型不合理，无法使用，只能重新购买直动式的电磁阀

由于计算失误，传送同步带的长度过长只能通过另外加装张紧轮使其正常工作

由于经验不足，同步带驱动电机安装位上打了键槽，导致电机固定不稳

由于对加工流程不够理解，打了大量减重孔位，部分孔位直径为 3mm，导致加工速度缓慢甚至无法加工

2. 嵌入式（张思成）：

基本没有什么问题，但在装限位开关的时候，发现 io 的值一直飘来飘去非常的不稳定，花了好长时间还是找不到原因，最后问了导师说一定要接三根线，然后才意识到是下拉电阻的问题，因为我以前用的板子自带下拉电阻所以限位开关接两根线就没有任何问题，而限位已经装在车上很难接第三根线所以最终加焊了一个电阻，从而解决这个问题。

3. 算法（梁德麒）：

可行性：由于搜索相当于枚举所有的可能性，故对于有解的地图，时间充足的情况下一定能算出结果；实际测试中对于随机数据，运算时间均在 1s 内。

深度优先搜索时间复杂度较高，但加入多线程后较好地解决了运算时间过长的问题，具体运行时间取决于电脑配置。

五 制作与测试流程

这一部分主要记录制作机器人与进入场地测试的经历，讲讲初始规划中与实际偏差较大的部分，分析问题原因，以及解决方案（讲讲是谁提出了解决问题的方案）

仓库的传送带与设计的偏差最大。我们在原先的规划中，传送带不仅可以运送方块，还可以让没有放整齐的方块回复整理，方便夹取。可是在制作的过程中，发现传送带与方块的摩擦力不够，难以完成这个操作。为了让传送带可以流畅工作，我们改进了夹子，在上面加装了辅助瞄准器，夹取的时候就让方块恢复整理，从而解决问题。（曹瑞翔完成）

之前我们原本打算使用全手动控制机器人（手动控制所有的机械控制）以应对紧急情况。然而发现手动控制更加容易发生“紧急情况”，因为错误率很高。所以我们现在改为了半自动控制（张思成完成）

六 结果与评价

对项目的优劣，创新点及可以改进的地方。

1. 机械：相对来说，我们的结构简单，稳定可靠性强。但是正是因为结构过于简单，让我们没有办法转换图传视角，也没有办法夹取竖直的方块。在仓库的设计理念中，我们的设计储存量偏低，让我们有一定的劣势。
2. 嵌入式：创新点一个便是半自动化，其次便是特别的稳。至于可以改进的地方我一直觉得应该写视觉，但大家都认为不好，因为上年用视觉的组都翻车了，大家都觉得关键不在于追求极致，而在于稳定。我认为也很有道理，也就依照了大家的想法，但我自己的看法是只要有时间，视觉也是可以做到很稳的，因此，我觉得这就是可以改进的地方吧。
3. 算法：相对于我们组使用的深度优先搜索，插头 `dp` 时间复杂度更低，效果更好，在要求更高的情况下应该采用更优的算法；另外可以考虑资源区的位置使结果更优。

七 附录（Appendix）请将代码及机械制图等粘贴在附录中
见文件尾

八 感想与感悟

从团队的角度谈谈在夏令营的收获与感悟，也可以谈谈未来的规划与发展（畅所欲言）。

这个大纲从理论上来说是一个小项目初级的研发流程，按照这个流程你们会省去很多不必要的麻烦，规避很多不必要的风险。你们通过这一段时间的经历应该对这个感触很深（如果你们真正投入了这个夏令营的话）当然大家也可以自由发挥，想什么写什么。把自己的看法都加进去，这个会是你们之后拿出去给别人展示的一份技术报告。

欧阳曜钧：我是第一次参加一小组为单位的研发活动。一直以来都是单干的我第一次感受到了团队合作的魅力。特别是当我买错关键零件而拖延进度的时候，没有人来责怪我，反而都在着手解决问题。（其中用木棍当做同步轮的方案让我哭笑不得）大家分工合作，没有人在划水。这种互相帮助的感觉真的非常棒。这次活动中，我最大的获得不仅仅是看到自己有多么渺小（这也是我来的原因）还看到了团队合作应有的样子，让我明白共同努力的滋味。

曹瑞翔：今年是第二次参加 RM 的夏令营，去年的夏令营里因团队合作问题我们组最后比赛结果并不理想。抱着再试一次的心态我再一次来到了这个夏令营。为了避免重蹈覆辙，一开始我就战战兢兢，尽力避免冲突。而后来事实上并没有这个问题。这次分工明确，并没有出现去年一样的全组争执，对于有分歧的方案也都通过交流解决了。各方面都比较顺利。同时我也看到了很多组因为交流问题比赛前几天还在拆车，改方案。也有很多组速度极快，前几天就完成了车辆的设计制造。这次夏令营更加让我意识到了团队合作的重要，虽然我们可能不是最终胜利的队伍，但这次经历对我来说绝对是一次美好而难忘的经历。

张思成：这次来夏令营给我最大的感受就是一个好的团队有多么的重要。所有的事情都会在对比中产生差距，当我去别的团队交流时就发现我们的团队是多么和谐，多么高效，没有人

划水，一切都按部就班的运转着，我不太懂机械，在负责的嵌入编程范围之外也帮梁德麒（负责算法）写了 UI，并一起完成了算法与 UI 的对接，偶尔也去机械那边拧拧螺丝。我们队员之间都互帮互助，也经常有队员主动来帮助我，这份感觉真的特别棒。因此与别的组对比起来，我发现我们组在研发的时候真的很顺利，而这一切都得益于我们之间的团队合作。同时，在这次的夏令营中，我也接触到了很多大佬，也交了不少的朋友，在与其他组聊天时也发现了很多新的东西等待着我去学习。在交流中我也收到了 FRC 大佬们的邀请，非常高兴我能得到他们的肯定，想必与他们的合作能像这次的我们一样和谐与高效。总之，在这次的夏令营中我收获了很多。

葛鸿：这次强者如云的夏令营，让我清楚地认识到了自己与他们的差距。对于比自己厉害的人，一直以来都只有一个模糊的概念，看清这份差距让我有些受挫，但是都说知耻而后勇，我希望通过自己之后的努力，能与那些我羡慕的队友们并肩站在一起。

梁德麒：这是我第一次参加机械相关的研发活动，也是独自离开家人时间最长的一次。在这之前，我更喜欢独自一人钻研，但后来发现效率并不高，而且和别人对比时总会感觉自己各个方面都不如别人。但参加这次夏令营，让我意识到了团队的力量，我虽然在机械、嵌入式等方面一窍不通，但在算法方面却能用自己擅长的方式帮助小组完成任务。而且值得庆幸的是组内关系十分和谐，我有得到队友们的支持、建议和帮助，也有感觉到自己为小组出了一份力，这让我对团队合作有了新的看法，每个人发挥自己的长处之余还能给予队友帮助，相信这份经历对我会有长期的、积极的影响。另一方面，和队友在各个方面的对比以及和其他组的算法的对比也让我意识到了我的不足，未来还有更大的提升空间。

附录:

1. 第一阶段算法代码 (梁德麒)

```
/*
 * dji summer camp task 1
 * 0=blank 1=red 2=yellow 3=blue 4=green
 * usage: tl.exe [input] [output]
 * input (eg):
 * 0 0 0 0 3 0 0 1
 * 0 0 0 2 0 0 0 0
 * 0 0 0 3 0 0 2 0
 * 3 3 0 0 4 0 4 0
 * 0 0 0 0 1 0 0 0
 * 0 0 0 1 0 0 0 0
 * 0 4 1 0 0 0 0 0
 * 4 2 2 0 0 0 0 0
 */
#include<bits/stdc++.h>
#include<thread>
#include <windows.h>
using namespace std;
#define NUM_THREADS 81
const int n=8;
struct hp{
    int x,y;
}aa[10][5];
int fx[5][2]={{0,0},{0,1},{1,0},{0,-1},{-1,0}};
int pd[6][2]={{1,2},{1,3},{1,4},{3,4},{2,4},{2,3}};
int mapl[10][10],b[5],c[100][5],co[5];
int VI[10][10],MAPS[10][10],RD[100],SE[100],flag,flag2,sx,sy,ths=1,thcnt=0;
void dfs(int x,int y,int now,hp a[10][3],int maps[10][10],int vi[10][10],int cnt,int rd[100],int se[100]){
    if(flag)return;
    cnt++;
    rd[cnt]=x*8+y-9;
    //if(!vi[x][y]){
    //    road[now][0]++;
    //    road[now][road[now][0]]=x*8+y-9;
    //}
    vi[x][y]=now;
    if((x==a[now][2].x)&&(y==a[now][2].y)&&(!flag)){
        if(now<8){
            se[now+1]=cnt+1;
            if(flag)return;
            dfs(a[now+1][1].x,a[now+1][1].y,now+1,a,maps,vi,cnt,rd,se);
            if(flag)return;
        }
        else{
            if(flag)return;
            if(cnt>=64/*>50?*/){
                flag=1;
                // printf("%d\n",cnt);
                for(int i=1;i<=8;i++){
                    for(int j=1;j<=8;j++){
                        VI[i][j]=vi[i][j];
                        MAPS[i][j]=maps[i][j];
                    }
                }
                for(int i=0;i<=9;i++)SE[i]=se[i];
                for(int i=0;i<=70;i++)RD[i]=rd[i];
                return;
            }
        }
    }
    return;
}
for(int i=1;i<=4;i++){
    x+=fx[i][0],y+=fx[i][1];
    if( (x>0)&&(x<=n)&&(y>0)&&(y<=n) )
        if( ( !vi[x][y] )&&( !maps[x][y] ) || ((x==a[now][2].x)&&(y==a[now][2].y)) )
            dfs(x,y,now,a,maps,vi,cnt,rd,se);
    x-=fx[i][0],y-=fx[i][1];
}
if(!flag){
    vi[x][y]=0;
    /*cnt--*/
}
}

void th(int t){
    if(flag)return;
    hp a[10][3];
    int maps[10][10],vi[10][10];
    // printf("thread[%d]ok\n",t);
    for(int i=1;i<=4;i++){
        a[i][1]=aa[i][pd[c[t][i]-1][0]];
        a[i][2]=aa[i][pd[c[t][i]-1][1]];
    }
}
```

```

        a[i+4][1]=aa[i][pd[c[t][i]+3-1][0]];
        a[i+4][2]=aa[i][pd[c[t][i]+3-1][1]];
//printf("[%d,%d %d,%d %d,%d %d,%d]\n",a[i][1].x,a[i][1].y,a[i][2].x,a[i][2].y,a[i+4][1].x,a[i+4][1].y,a[i+4][2].x,a[i+4][2].y);
    }
    memset(maps,0,sizeof(maps));
    for(int i=1;i<=8;i++){
        for(int j=1;j<=2;j++){
            maps[a[i][j].x][a[i][j].y]=i;
        }
    }
    memset(vi,0,sizeof(vi));
    int rd[100],se[10];
    memset(rd,0,sizeof(rd));
    memset(se,0,sizeof(se));
    se[9]=65;
    dfs(a[1][1].x,a[1][1].y,1,a,maps,vi,0,rd,se);
    if(flag)return;
    thcnt++;
}

void try1(int x,int y){
    b[x]=y;
    if(x<4)try1(x+1,1);
    if(x>=4){
        for(int i=1;i<=4;i++)c[ths][i]=b[i];
        ths++;
    }
    if(y<3)try1(x,y+1);
}

int main(int a1,char * args[]){
//printf("%s",args[1]);
freopen(args[1],"r",stdin);
freopen(args[2],"w",stdout);
int tmp;
for(int i=1;i<=n;i++){
    for(int j=1;j<=n;j++){
        scanf("%d",&tmp);
        map1[i][j]=tmp;
        if(tmp){
            int k=1;
            while(aa[tmp][k].x)k++;
            aa[tmp][k].x=i;aa[tmp][k].y=j;
        }
    }
}
//read
flag=0;
try1(1,1);
for (int i=1;i<=NUM_THREADS; i++)
{
    thread t(th, i);
    t.detach(); //join

/*for (int i=1;i<=9; i++)
{
    for(int j=i*9-8;j<=i*9;j++){
        thread t(th, j);
        t.detach();
    }
}*/
//for(int i=1;i<=81;i++)th(i);
// Sleep(5000);
while(1){
    if((thcnt==81)&&(!flag)){
        printf("FAILED");
        fclose(stdin);
        fclose(stdout);
        return 0;
    }
    if(flag){
        memset(co,0,sizeof(co));
        printf("\ncolor\":[");
        for(int i=1;i<=8;i++){
            for(int j=1;j<=8;j++){
                if(VI[i][j]>4)VI[i][j]==4;
                printf("%d",VI[i][j]);
                co[VI[i][j]]++;
                if(!((i==8)&&(j==8)))printf(",");
            }
        }
        printf(",");
        printf("\npoint\":[");
        for(int i=1;i<=8;i++){
            for(int j=1;j<=8;j++){
                if(!MAPS[i][j])printf("0");else printf("1");
                if(!((i==8)&&(j==8)))printf(",");
            }
        }
        printf("]");
        printf("\nroute\":[");
        SE[1]=1;
        for(int i=1;i<=8;i++){
            printf("[");
            for(int j=SE[i];j<SE[i+1];j++){
                printf("%d",RD[j]);
                if(j<SE[i+1]-1)printf(",");
            }
            if(i<8)printf(",");
            else printf("]");
        }
        /* printf("\nR:%d",co[1]-4);
        printf("\nY:%d",co[2]-4);
        printf("\nB:%d",co[3]-4);
        printf("\nG:%d",co[4]-4);*/
        fclose(stdin);
        fclose(stdout);
        return 0;
    }
    else Sleep(100);
}
}
}

```