

```
Point p2 = info.b.getPoint(minfo.j));  
Point dp = Util.subPoint(p1, p2);
```

俺たちはプロコンに行ったら まず真っ先にバグを確認する

```
if (Util.doubleEqual(angle, Math.PI)) {  
  
    ;  
}
```

高専プロコンとは.txt

- 1 全国の高専生が一堂に会して、プログラミング能力や開発能力、プ
- 2 レゼンテーション能力を競う大会。
- 3
- 4 課題部門・自由部門・競技部門の三部門が存在し、
- 5 「俺達はプロコンに行ったらまず真っ先にバグを確認する」は
- 6 競技部門に出場した。

```
continue;
```

```
} else if (Util.doubleEqual(angle, Math.PI * 3 / 2)) {
```

競技部門のルール.txt

- 1 一枚の板を切り抜いたパズルが与えられるのでできるだけ早く、こ
- 2 のパズルを完成させる問題。
- 3
- 4 ピースの形状や、あるピースがどこに置かれるかといったヒントが
- 5 QRコードで与えられるが、これを使用すると減点。
- 6
- 7 減点が少なく、早くパズルを完成させたチームが勝利

```
}  
  
Piece merge = Piece.merge(info.a, minfo.b, dp.x, dp.y, info.a.p);  
for (int i = 0; i < merged.size(); i++) {  
    Piece e1 = merged.get(i);  
    if (e1.isMerged, i < merged.size()) {  
        this.es.add(e1);  
        continue;  
    }  
}
```

```
for (Piece e2 : a) {
```

```
for(Piece p:pieces){
    int id=1;
```

俺たちはプロコンに行ったら まず真っ先にバグを確認する

```
if(count==0){
    c++;
}
```

戦略.txt

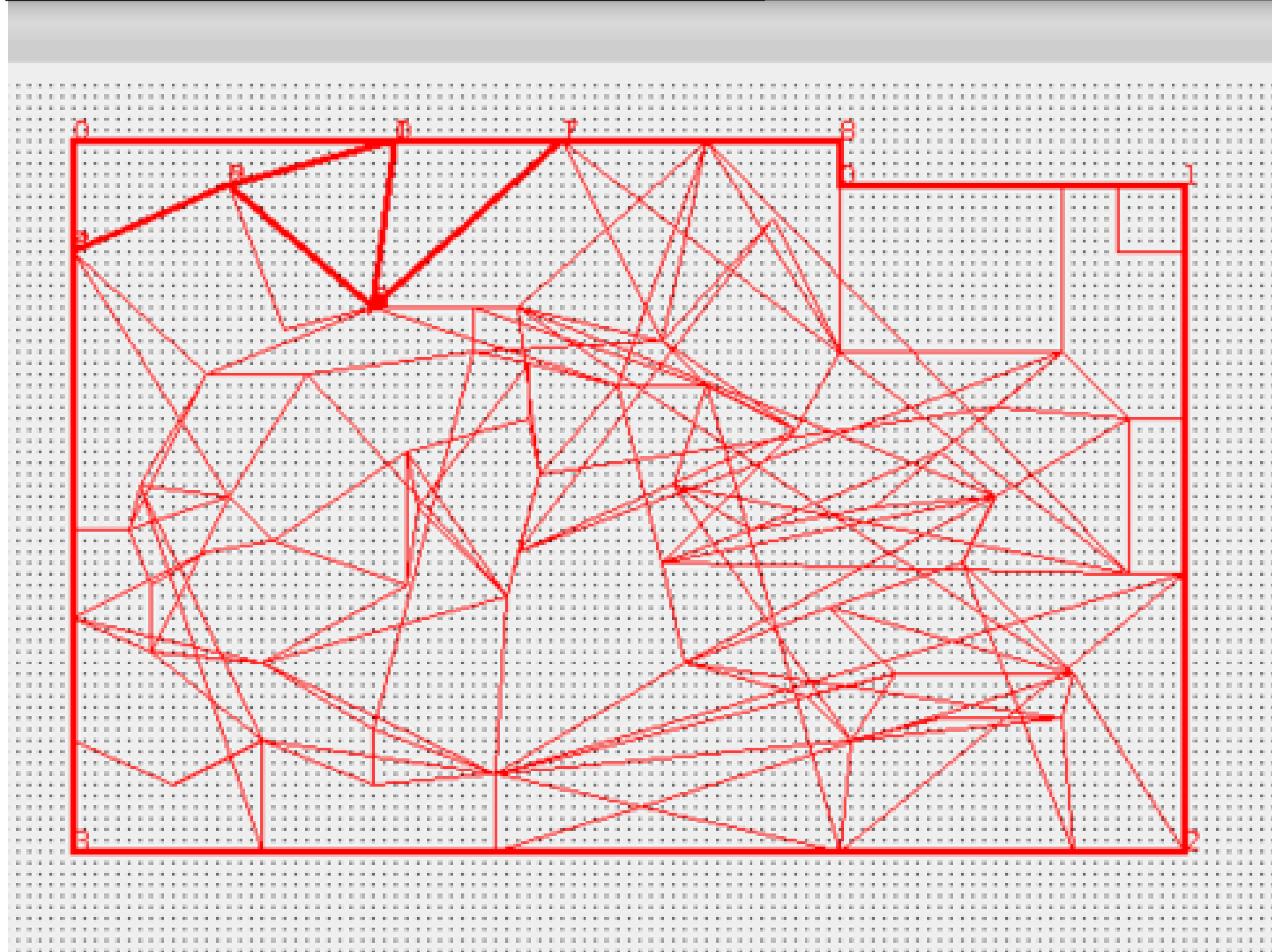
- 1 ピースの情報を画像処理で取得するのは**つらい**ので、QRコードを
 - 2 使う
 - 3
 - 4 頂点の角度と辺の長さから、**明らかにくっつきそう**なピースを探し
 - 5 て**結合してみる**
 - 6
- これを繰り返すと最後には一つの大きなピースになるはず

```
p2.update(temp.candidate.id,p2.size,temp
break;
```

default:

成果物.png

```
p2.update(temp.candidate.id,0,0,temp.p);
```



なんかすっごく汚いのは
バグを確認してなかったから。

変な線を取り除いて心の眼で見
ると、ちゃんとパズルが復元され
ている

```
id=r.parent.id;
```

成績.txt

- 1 予選 ==> 8位。**準決勝進出！**
- 2
- 3 準決勝 ==> 12位。**敗退……**
- 4