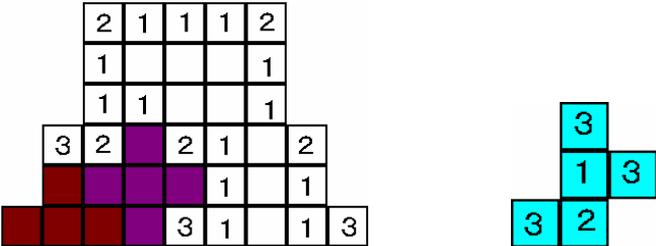


## 応募内容(競技部門用)

<b>No.1</b>	タイトル	オレンジ
<b>No.2</b>	1) 予定開発期間： 5ヶ月 2) 予定開発人数： 3人	
<b>No.3</b>	<p>実現方法</p> <h3 style="margin: 0;">オークション</h3> <ul style="list-style-type: none"> <li>・オークションに出ている残りの石の個数</li> <li>・どのチームがどの石を持っているか</li> <li>・どのチームがどの石に入札したか</li> <li>・どのチームがどの石にいくらで入札したか</li> </ul> <p>などの情報を入力。そこから</p> <ul style="list-style-type: none"> <li>・相手がどの石を欲しがっているのか</li> <li>・相手がいくらで石を入手しようとしているのか</li> <li>・買えそうに無い石はどれか</li> <li>・簡単に買える石はどれか</li> </ul> <p>といった項目を考察させる。(たとえば、対戦相手が高い優先順位をつけていながら落札できなかった石がある場合は、対戦相手は次もその石に入札する可能性が高い。)そして、最終的に自分の持っている石やツヤマなどの情報から、どの石にいくらで入札すればよいのかを算出させる</p> <h3 style="margin: 0;">石垣作り</h3> <p>総当りを基本として、最適な石垣の探索を行う。さらに、以下の案による効率化を施す。</p> <ul style="list-style-type: none"> <li>・石垣枠の隙間大きさより、石の大きさが大きな場合、探索を行わない。</li> <li>・石垣枠や、石の一コマ(以下ブロック)の外側に接している辺の数を利用する。(図を参照)石垣枠の数字よりブロック側の数字が小さいと、その位置にそのブロックを入れることは出来ないため、無駄な探索を省くことが出来る。</li> <li>・石が地面に接しているかの判定は、探索途中では行わない。(無駄な計算を省く。)</li> </ul> <div style="text-align: center; margin: 10px 0;">  </div> <p style="text-align: center;">図.石垣枠と石</p>	
<b>No.4</b>	使用ソフト Microsoft Visual Studio .NET  使用言語 C++	
全国高等専門学校 第18回プログラミングコンテスト：津山		