

応募内容(競技部門用)

No.1	タイトル	みそ - 利用価値がよく分からない問題を解くためのソフトウェア -																																								
No.2	<p>1) 予定開発期間： 7ヶ月 2) 予定開発人数： 3人</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 30px;"></td> <td style="width: 20px;">4</td> <td style="width: 20px;">5</td> <td style="width: 20px;">6</td> <td style="width: 20px;">7</td> <td style="width: 20px;">8</td> <td style="width: 20px;">9</td> <td style="width: 20px;">10</td> </tr> <tr> <td>問題分析</td> <td>←→</td> <td></td> <td>←→</td> <td></td> <td>←→</td> <td></td> <td></td> </tr> <tr> <td>設計</td> <td></td> <td>←→</td> <td></td> <td>←→</td> <td></td> <td>←→</td> <td></td> </tr> <tr> <td>実装</td> <td></td> <td></td> <td>←→</td> <td></td> <td>←→</td> <td></td> <td>←→</td> </tr> <tr> <td>試用・トレーニング</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>←→</td> </tr> </table> <p style="text-align: center;">← フェーズ1 フェーズ2 フェーズ3 →</p>		4	5	6	7	8	9	10	問題分析	←→		←→		←→			設計		←→		←→		←→		実装			←→		←→		←→	試用・トレーニング							←→	
	4	5	6	7	8	9	10																																			
問題分析	←→		←→		←→																																					
設計		←→		←→		←→																																				
実装			←→		←→		←→																																			
試用・トレーニング							←→																																			
No.3	実現方法	<p>プロトタイプモデルを用いてアルゴリズムの構築を行う。上図では3つのフェーズでの開発日程を示したが、可能な限りフェーズ数を増やし、ソフトウェアの完成度の向上を目指す。</p> <p>現状はフェーズ1が終了したところで、次の3つのプログラムを試作した。</p> <ol style="list-style-type: none"> 1. 分枝限定法を用いて、厳密解¹を求める。 2. 焼きなまし法²を利用し、解を求める。スコアリング関数は単純に大会規定と同様のものを使用している。 3. 問題分析の補助ソフトウェアとして、1,2で得られた解を視覚化し、ソフトウェアが正常に動作しているかどうか、またアルゴリズムに改善の余地はないかどうかなどの検討に利用する。 <p>また次の問題点が見つかっている。</p> <p>1の方法では、4×4の大きさまでしか安定して解を得られないので、少なくとも大会規定の下限である5×5まで安定して解を出せるようにする必要がある。</p> <p>2の方法では、最終的な解は導けるものの、ステップ数が膨大な数となってしまうこと。3のツールを用いて分析し、スコアリング方法や探索順序の改善を行う必要がある。</p> <p>これらのアルゴリズムで実際に問題を解かせてみて分かった事は、この問題は厳密に解くためには、莫大なメモリを必要とすることである。このため、少なくとも厳密解は5×5以上の問題では求められない。</p> <p>以上のことを踏まえ、フェーズ2では次の方針に従い改良を行う。</p> <ol style="list-style-type: none"> 1. 分枝限定法でのデータ構造を見直し、メモリ効率の向上を図る。5×5の問題が安定して解けるようになることを目指す。 2. 焼きなまし法でのスコアリング関数を見直し、より早く解に収束するように改善する。解までのステップ数を現状の50%以下までに減少させることを目指す。 																																								
No.4	開発環境	<p>Visual Studio 2005/2008 Professional (C++, C#) Cygwin</p>																																								
全国高等専門学校 第20回プログラミングコンテスト：木更津																																										

¹ 最小のステップ数での解