Cプログラミング

大レポート

課題４

指名：　　　樋口陽祐

所属：　基幹理工学部

学籍番号：1w1523146

出題日：　 1月9日

提出日： 1月16日

担当者：青柳滋己先生

課題内容：

対角線ナンバープレースを解くプログラムの作成

実行結果：

numplace.txtを参照。

考察：

人はナンプレを解く上で、あるマスにおける数が、そのマスの行、列、3×3エリアにおける数とかぶらないように盤面を埋めていく。ある盤面のマスに入りうる数の候補が1に絞られる場合、その数がそのマスに入る。しかし、この解法のみだと、入る数の候補が1つだけに定まるマスが盤面上に無くなった時、できなくなってしまう。そこで、コンピュータの処理速度を活かし、あるマスに入りうる数によって盤面の分岐を行うことで、解を見つけるプログラムを書いた。分岐方法は、数学における順列と同じ考えである。空白のマスが3つあったとし、それぞれの候補数が4、2、7個だったとする。この場合、盤面のマスにおける数の被りを考慮しないと、4×2×7通りの盤面の表し方があり、この中から、数が全て被っていない場合の盤面を探せばよい。

解法：

あるマスに入る候補の数を一つずつ確かめていき、再帰処理により最終的にすべてのマスが埋まれば、それが入力された盤面の解答と出力するというのが大雑把な流れである。ソースコード（numplace.c）より、init関数では課題4で形式が指定された盤面の入力を行い、入力された盤面を2次元配列（9\*9）のグローバル変数boardに代入する。disp関数は、boardを画面に出力する。checkava関数は、盤面のx行y列目で、整数nがそのマスの候補となる場合は、1を返し、ならない場合は0を返す。solve関数の77~85行目では、盤面が空いているマスを探し、空いているマスの座標をxとyに代入する。86~90行目では、x=-1かつy=-1である時、つまり盤面のマスが全て埋まったとき、その盤面を出力する。91~97行目では、avaが盤面のx行y列目に入りうる数であったら、boardのx行y列にavaを代入し、その盤面を引数としたsolve関数を呼び出す。boardのx行y列目をavaにすることで、その他のマスがうまく埋まらなかったら、boardのx行y列を再び0とし、他のavaの値を代入し同じ動作を繰り返す。

補足のため、solve関数の流れを書くと以下のようになる。

1. 盤面上で空いているマスを見つける。
2. 空いているマスがなかったら、盤面を出力しmain関数へ戻る。
3. 空いているマスに入りうる数を探す。
   1. 見つかったら、空いているマスにその数を代入
   2. solve関数を呼び出す。
   3. 数を代入したマスをまた空白に戻す。

solve関数からmain関数に戻るのは、盤面が全て埋まったときまたは起こりうる全ての分岐を調べ終えたときである。

感想：

前から数独を解くプログラムを書いてみたかったため、レポート課題としてやれたのは嬉しかった。ある値を空白のマスに代入し、盤面が詰んだら改めて違う値で試すというアルゴリズムを実装するのに苦労した。だが、再帰の仕組みの確認などができたため、期末テストの勉強そっちのけでやった甲斐があった。

参考文献

C/C++リファレンス

<<http://www.cppll.jp/cppreference/>>

C言語関数辞典

　<<http://www.c-tipsref.com/>>

「再帰呼び出し」

　<<http://ylb.jp/2007b/proc/recursion/>>

「第１４回プログラミング演習　講義概要」

　< http://www-it.sci.waseda.ac.jp/CPR2/class607/CPR2-14.pdf>