

数独 (ナンバープレース)

知能システム学
ミニプロジェクト発表資料

0312011012 岩館光
0312011015 梅田龍之輝
0312011032 上山拓也
0312011077 菅原太一

目次

- 数独（ナンバープレース）
- プロジェクト目的
- 開発環境・使用アルゴリズム
- データ構造
- アルゴリズム具体例：解法
- プログラム実行結果

数独（ナンバープレース）

- 数独（ナンバープレース）とは， 3×3 のブロックに区切られた 9×9 の正方形の枠内に1～9までの数字を入れるパズルである
- ルール
 - 空いているマスに1～9いずれかの数字を入れる
 - 縦・横の各列，及び太線で囲まれた 3×3 のブロック内に同じ数字が複数入ってはいけない

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

プロジェクト目的

- 今回のプロジェクトでは、数独における、**数字の配置**の部分に焦点を当てた
- つまり、初期配置は入力させておく必要があるということ
- ゴール
ルール通りにすべての空欄に数字が埋まる
- 対象とする問題
初期状態で与えられている数字から解を導く

開発環境・使用アルゴリズム

- 開発環境
 - Linux (RedHat Enterprise 5)
 - C言語
- 使用アルゴリズム
 - **深さ優先探索アルゴリズム**を使用
(アルゴリズム詳細は割愛)
 - 配置できるマスを見つけ、実際に配置をするために使用

データ構造

- データ構造

- 問題データの読み込み

```
int readproblem(FILE *stream);
```

- 盤面

```
int board[9][9];
```

- 問題を解く

```
void trypos(int x, int y);
```

- 解の候補

```
int possible[10];
```

- 実行結果の出力

```
void printboard(void);
```

アルゴリズム具体例：解法

- 解法

- a. $(r,c) = (0,0)$ から探索を開始する (r:行 / c:列)
- b. $(r + 1, 0)$, $r < 9$ ならば $(r, c + 1)$, $c < 9$ まで探索
- c. マスが空欄ならば, そのマスの解の候補 $(r,c,BOX(3*3))$ に重複していない数字をあげる.
空欄でなければ次のマスに進む
- d. 空きマスに候補の数字を入力
- e. $r = 9$ のとき, ルールどおり数字が埋まっていれば
探索終了
- f. 空きマスに解の候補がなければ1つ前のマスに戻る

実行結果

- 今回説明したプログラムを実装した
 - Sudoku.c #プログラムソース
 - コンパイル `cc -o Sudoku Sudoku.c`
 - 実行 `./Sudoku < Sudoku-20-1.txt`
 - Sudoku-xx_X.txt #与える問題のデータ
 - 例として, Sudoku-xx_1.txt / Sudoku-xx_2.txt / Sudoku-xx_3.txtを添付する
 - この中に, 18~38個のヒントを持つ3つの問題が入っている
 - 実行時は, Sudoku-18_1.txtなどとして, ファイル内に1つのヒントのみが入っている状態にする必要がある

実行結果

- 実行にかかる時間

- 初期状態で埋まっているマス

- 30個以上 : 0.01秒以内
- 24~29個 : 0.01秒 ~ 0.4秒
- 20~23個 : 0.1秒 ~ 7秒
- 19個 : 1秒 ~ 22秒
- 18個 : 23秒 ~ 1分20秒