

# 実行するだけ

とりあえず、動かしてみたい人のため説明です。

GitHubから、とりあえずまとめてダウンロードに、適当なところに展開してください。

ここでは、C:\Users\fuji\Desktop\ナンプレ に展開した場合の実行を示しています。

現在、\Users\fuji\Desktop\ナンプレ に居て、ファイル構成が以下のようになっている  
とします。

```
C:\Users\fuji\Desktop\ナンプレ>tree /F
```

フォルダー パスの一覧

```
C:.
```

```
| NP.jar
|
├─data
|   17P.txt
|   18P.txt
|   HeartP.txt
|   HeartQ.txt
|   Pattern500.txt
|   Problem500.txt
|
└─src
    contributing.md
    Generator.java
    NP.java
    Solution.java
    Solver.java
```

```
C:\Users\fuji\Desktop\ナンプレ>java -jar NP.jar
```

```
===== arguments input error =====
```

```
java -jar NP.jar -s problem_file [answer_file]
```

```
java -jar NP.jar -g pattern_file [problem_file]
```

パラメータなしでNP.jarを起動すると、上記のように入力形式を表示します。

パラメータにエラーがある場合も、上記の表示になります。

データを、GitHubの配置と同じで、dataフォルダに置いた状態になっているとします。

# 問題を解く

パラメータ    -s は問題を解く(solve)指示です。  
                  -g は問題を生成する(generate)指示です。  
-s に引き続いて、問題ファイルを与えると、問題を解きます。

## 正常な問題を解く

```
C:\Users\fuji\Desktop\ナンプレ>java -jar NP.jar -s data\HeartQ.txt
```

```
Heart    H 20
```

```
- - - - -
```

```
- 1 6 - - - 5 9 -
```

```
4 - - 3 - 7 - - 2
```

```
7 - - - 5 - - - 1
```

```
8 - - - - - - - 9
```

```
- 6 - - - - - 7 -
```

```
- - 5 - - - 6 - -
```

```
- - - 2 - 3 - - -
```

```
- - - - 7 - - - -
```

```
0
```

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

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

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

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

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

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

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

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

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

```
Total 1      Success 1
```

```
total time : 759200 nano sec,      average : 759 micro sec
```

最後に

与えられた問題数

きちんと解けた問題数

全体の計算時間（ナノ秒単位）

1問あたりの平均計算時間（マイクロ秒）

を表示します。

問題と答えの間にある数字（この場合は0）は、解けきれなかったマス数を示します。

きちんと解ければ、0になります。

正の数になった場合は、解けきれなかったマス（空白ます「-」）の数を示し、多重解の問題になってしまったことが分かります。

### 解ききれない問題（多重解の問題）を解く

次の例は、正しい問題の数字を一箇所だけ変更したものです。

すると、残りマス数が23になり、最後の成功数が0になっています。

```
C:\Users\fuji\Desktop\ナンプレ>java -jar NP.jar -s data\HeartQ1.txt
```

```
Heart    H 20
```

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

23

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

```
Total 1    Success 0
```

```
total time : 666900 nano sec,    average : 666 micro sec
```

## エラーの場合

さらに、問題の同じマスの変更を試みます。

```
C:\Users\fuji\Desktop\ナンプレ>type data\HeartQ2.txt
```

```
Heart   H 20
```

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

これで実行すると、以下のようになりました。

現バージョンでは、エラーの場合は、問題を表示しません。ちょっと不親切ですね。

```
C:\Users\fuji\Desktop\ナンプレ>java -jar NP.jar -s data\HeartQ2.txt
```

```
Heart   H 20
```

```
ERROR
```

```
Total 1    Success 0
```

```
total time : 175200 nano sec,    average : 175 micro sec
```

# 問題を作る

-gオプション付きでパターンファイルを与えると、パターンを表示後、\*を次々と表示します。

\*1つが、試行錯誤(TRY)1回を示します。

```
C:\Users\fuji\Desktop\ナンプレ>java -jar NP.jar -g data\HeartP.txt
```

```
No.1    H 20
```

```
- - - - -
- X X - - - X X -
X - - X - X - - X
X - - - X - - - X
X - - - - - - - X
- X - - - - - X -
- - X - - - X - -
- - - X - X - - -
- - - - X - - - -
```

```
*****
```

```
*****SUCCESS  TRY 91
```

```
- - - - -
- 7 5 - - - 9 2 -
6 - - 4 - 9 - - 3
1 - - - 5 - - - 4
3 - - - - - - 6
- 4 - - - - - 3 -
- - 2 - - - 5 - -
- - - 3 - 1 - - -
- - - - 4 - - - -
```

```
total 1  failure 0
```

```
total time : 3902 mili sec, average : 3902 mili sec
```

この例では、91回の再トライで成功しました。

最も少ない場合は、\*1つで成功し、 TRY 0 と表示されます。

次の例は、

```
C:\Users\fuji\Desktop\ナンプレ>java -jar NP.jar -g data\ImpossibleP.txt
```

```
No.1    H 17
```

```
- - - - - X X
- - - - - 
- - X X X - - - 
- - X - - X - - 
- - X - - - X - 
- - - X - X - X - 
- - - - X - X - 
X - - - - X - - 
X - - - - - - -
```

```
*****
*****
*****
*****
*****
*****FAILURE
*****
```

```
total 1  failure 1
```

```
total time : 15727 mili sec,      average : 15727 mili sec
```

このパターンでは、多数回試してもだめで、あきらめてしまいました。

現在は、400回試してだめだったらあきらめ、FAILUREと表示します。

このパターンは、このプログラムには組み込んでいない手筋を使って解く・作るプログラムでは、以下のような問題を作ることができます。

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

## 問題を連続して解く

ナンプレ500問が1つのファイルになっている Problem500.txt の問題を解いてみます。

```
C:\Users\fuji\Desktop\ナンプレ>java -jar NP.jar -s data\Problem500.txt
```

No.1 H 18

```
- - - - 8 - - - -  
- - - 9 - 7 - - -  
4 2 - - - - 3 - -  
- - 7 - - 9 - - -  
- 8 - - - - - 3 -  
- - - 1 - - 4 - -  
- - 2 - - - - 8 7  
- - - 3 - 4 - - -  
- - - - 5 - - - -  
0
```

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

No.2 H 18

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

◇ ◇ 中略 ◇ ◇

No.500 H 24

```
- - 2 3 - - - - -  
- - 4 - - 5 - - -  
- - - - - 6 - 1 8  
- 2 8 5 - 7 - - 6  
- - - - - - - - -  
7 - - 6 - 1 5 9 -  
9 5 - 7 - - - - -  
- - - 2 - - - - -  
- - - - - 3 7 - -  
0  
8 9 2 3 1 4 6 7 5  
6 1 4 8 7 5 9 2 3  
3 7 5 9 2 6 4 1 8  
1 2 8 5 9 7 3 4 6  
5 6 9 4 3 2 1 8 7  
7 4 3 6 8 1 5 9 2  
9 5 1 7 6 8 2 3 4  
4 3 7 2 5 9 8 6 1  
2 8 6 1 4 3 7 5 9
```

Total 500 Success 500

total time : 55101600 nano sec, average : 110 micro sec

一気に解いて、1問平均で110マイクロ秒の速度で解きました。

ヒント数が18~24個の問題500問の問題集です。

あまりにも高速に解くので、時間計測が正しくなるように、一気に解きながら解を記憶し  
(表示しない)、解いたときの経過時間を記憶しています。

解き終えてから、問題と解を示し、最後に経過時間を表示しています。

これは、解く時間に比べて表示の時間が圧倒的に長いための措置です。



# 問題を連続して作る

問題集の数字を全てXにしたパターンファイルが Pattern500.txt です。

```
C:\Users\fuji\Desktop\ナンプレ>java -jar NP.jar -g data\Pattern500.txt
```

No.1 H 18

```
- - - - X - - - -  
- - - X - X - - -  
X X - - - - X - -  
- - X - - X - - -  
- X - - - - - X -  
- - - X - - X - -  
- - X - - - - X X  
- - - X - X - - -  
- - - - X - - - -
```

\*\*\*\*\*SUCCESS TRY 8

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

No.2 H 18

```
- X - - - - X - -  
- X - X - - - X -  
- - - X - - - - -  
- - - - - X X - -  
- - X - - - X - -  
- - X X - - - - -  
- - - - - X - - -  
- X - - - X - X -  
- - X - - - - X -
```

\*\*\*\*\*SUCCESS TRY 4

◇ ◇ 中略 ◇ ◇

\*SUCCESS TRY 0

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

No.500 H 23

- - X X - - - - -  
- - X - - X - - -  
- - - - - X - X X  
- X X X - X - - X  
- - - - - - - - -  
X - - X - X X X -  
X X - X - - - - -  
- - - X - - - - -  
- - - - - X X - -

\*\*\*\*\*SUCCESS TRY 4

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

total 500 failure 0

total time : 133233 mili sec, average : 266 mili sec

失敗(failure)は0です。500問の作成を133秒で終わりました。  
1問あたり266ミリ秒(0.266秒)で作れていますので、かなり高速に作成できています。

問題作成の難易度（失敗のしやすさ）は、パターンによってかなり違います。  
ヒント数が少なくなることも、作りにくくします。

このような感じで、問題を作成することができます。

実は、問題ファイルを、そのままパターンファイルとして利用することができます。  
同じファイルを、-s で動かすと、指定されたファイルを問題ファイルとして実行します。  
-g で動かすと、マスが - になっていない箇所は全てヒントマスの指定と解釈して解きます。

```
C:\Users\fuji\Desktop\ナンプレ>java -jar NP.jar -g data\HeartQ.txt
```

```
No.1    H 20
```

```
- - - - -
- X X - - - X X -
X - - X - X - - X
X - - - X - - - X
X - - - - - - X
- X - - - - - X -
- - X - - - X - -
- - - X - X - - -
- - - - X - - - -
```

```
*****SUCCESS  TRY 11
```

```
- - - - -
- 3 8 - - - 7 4 -
2 - - 3 - 6 - - 1
6 - - - 4 - - - 5
1 - - - - - - 3
- 2 - - - - - 1 -
- - 7 - - - 8 - -
- - - 1 - 5 - - -
- - - - 6 - - - -
```

```
total 1  failure 0
```

```
total time : 518 mili sec, average : 518 mili sec
```