

# Le problème des huit reines

Philippe Langevin

Octobre 2007

## Abstract

Le document est basé sur le programme de TROILO Fabrice qui prend en entrée un entier  $n$  puis calcule le nombre de solutions du problème des  $n$  reines. Le programme précise le nombre de noeuds de l'arbre abstrait correspondant à l'exécution de la fonction récursive `void backtrack(int num, int n, int *pos)`. Le programme a été modifié pour construire une image de cet arbre pour les petites valeurs ce qui donne un exemple de la commande `dot` du projet `graphviz`. Le tout est très sommaire, idéalement l'étudiant devrait compléter la source `reine.tex` pour obtenir un document correctement finalisé.

## Contents

<a href="#">1</a>	<a href="#">reines.c</a>	<a href="#">2</a>
<a href="#">2</a>	<a href="#">Output</a>	<a href="#">3</a>
<a href="#">3</a>	<a href="#">Graphique</a>	<a href="#">4</a>
<a href="#">4</a>	<a href="#">Liens vers les sources</a>	<a href="#">4</a>

## 1 reines.c

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <ctype.h>
#include <string.h>

int cpt=0;
int noeud=0;

int libre(int r, int c, int* pos, int n)
{
    int i, j=0;
    for (i=(r-1); i>=0; i--)
    {
        noeud++; if (pos[i]==c)
        {
            return 0;
        }
    }
    for (i=(r-1), j=(c+1); (j<n) && (i>=0); i--, j++)
    {
        if (pos[i]==j)
        {
            return 0;
        }
    }
    for (i=(r-1), j=(c-1); (j>=0) && (i>=0); i--, j--)
    {
        if (pos[i]==j)
        {
            return 0;
        }
    }
    return 1;
}

void placer (int num, int j, int* pos)
{
    pos[num]=j;
}
```

```

void backtrack(int num, int n, int *pos)
{
    noeud++;
    int j;
    if (num >= n)
    {
        cpt++;
    }
    else
    {
        for (j=0; j<n; j++)
        {
            if (libre (num, j, pos, n))
            {
                pos [num]=j;
                backtrack (num+1, n, pos);
            }
        }
    }
}

int main (int argc, char* argv [])
{
    int n, *pos;
    n=atoi (argv [1]);
    pos=(int*) malloc (sizeof (int)*n);
    backtrack (0, n, pos);
    free (pos);
    printf ("\nNb solutions :%d\nNb noeuds :%d\n", cpt, noeud);
    return 0;
}

```

## 2 Output

Une exécution est illustré par la figure 1.

```
./reines.exe 8
```

```
Nb solutions :92
Nb noeuds :60745
```

Figure 1: Un exemple d'exécution: ./reines.exe 8

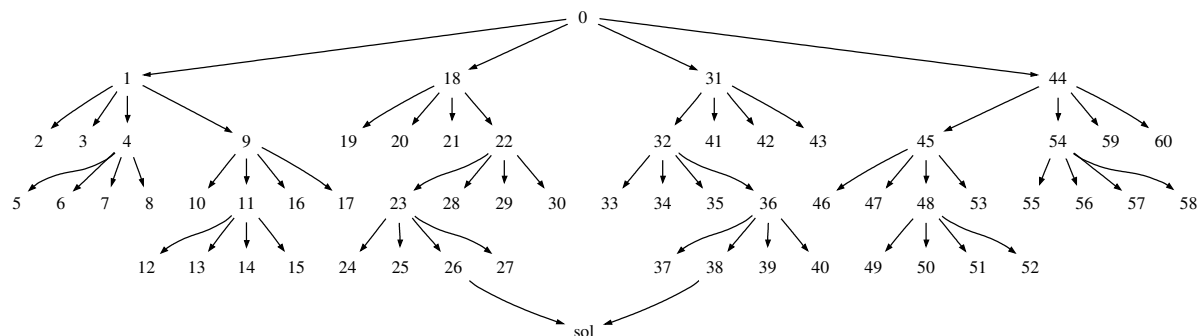


Figure 2: Exemple.

### 3 Graphique

### 4 Liens vers les sources

- [makefile.sh urls.sh](#)
- 
- [reines.c](#)
- [reines.tex urls.tex](#)
- 
- 

### References

- [1] Jean-Pierre Zanotti Travaux-Pratiques d'Algorithmique Module I51, licence.  
<http://zanotti.univ-tln.fr/enseignement/I51/TP3.html>
- [2] Le package “listings” <ftp://ftp.inria.fr/pub/TeX/CTAN/macros/latex/contrib/listings/>
- [3] Le site “graphviz” <http://www.graphviz.org/>