

**MAC0239**

Apresentação para o EP1

Cássio Cancio

# Resumo do EP

## 1 Introdução

Mastermind é um jogo para dois jogadores onde um deles assume o papel de criador do código (*codemaker*) e o outro tentará decifrá-lo (*codebreaker*). O código é formado colocando pinos de uma das 6 cores em cada um dos 4 espaços apropriados. Depois de criado o código, o *codebreaker* tem um número pré-determinado de tentativas para adivinhar o código gerado. A cada tentativa, o *codemaker* indicará por meio de pinos o quão próximo da resposta o chute do *codebreaker* está: para cada pino da cor correta na posição certa, um pino de cor preta de feedback é dado, e para cada pino de uma cor certa, mas em uma posição incorreta é fornecido um pino de cor branca.



# Modelagem

[illegible]

Cada casa da tabela é uma variável booleana ( $x_{ij}$ )

# Tarefas

1. No arquivo `codebreaker.h` você deverá alterar a constante (`#define`) `SOLVER_CALL` para que ela contenha o caminho até a versão executável do `minisat` conforme instalado no seu computador.
2. No arquivo `codemaker.c`, você deverá alterar a função `codemaker_feedback` para que ela indique também cores presentes em algum espaço, porém inserida na posição errada. Por exemplo, se o código for `[ 0 1 2 3 ]` e a última tentativa foi `[0 2 4 6]`, o feedback devolvido poderá ser `[1 2 0 0]`, com a seguinte interpretação. O número 1 indica que a primeira posição está com a cor 0 correta; o número 2 na segunda posição indica que a cor presente nessa posição ocorre em outra posição do código; por fim, o número 0 nas duas últimas posições indica que aquelas cores não ocorrem em qualquer lugar do código.
3. No arquivo `codebreaker.c`, você deverá alterar a função `convert_feedback`, de modo a utilizar o feedback mais informativo gerado pela modificação anterior.

# Entrega

## 7 Entrega

Entregue o código com sua versão modificada do exercício-programa e o relatório em PDF em um arquivo zip no campo adequado no e-disciplinas. O nome do arquivo zip deve seguir a seguinte regra: se seu nome for Fulano Ciclano Beltrano, o nome do arquivo deve ser: `ep1_fulano_ciclano_beltrano.zip`.

# Instalação do Minisat



**Mahesh Spiritson**

Works at Rockwell Collins (company) · 7y



Originally Answered: How do I install minisat on a linux machine?

I am assuming you are using a Ubuntu- based system. If not, you need to get zlib-dev package for your system. If you are using ubuntu based system, go ahead with the steps.

1. Download the "[minisat-2.2.0.tar.gz](#)". Go to the downloaded directory and open the Linux Terminal there. [link](#)

2. Run "tar xvf [minisat-2.2.0.tar.gz](#)".

3. Run "cd minisat"

4. Run "export MROOT=\$(pwd)"

5. Run "cd core".

6. Run "sudo apt-get install libghc-zlib-dev".

7. Run "make"

8. Run the application with "./minisat".

The application should run properly now.

4.6K views · View upvotes



5



# Instalação do make

Verificar se o make já está instalado:  
*make -version*

Caso não esteja:  
*sudo apt install make*

# Instalar zlib-dev

Caso o seu Linux não seja baseado no Ubuntu, é bastante provável que ele não possua a *zlib-dev*. Instale com:

```
sudo apt-get install libghc-zlib-dev
```



# Tentar rodar o programa

```
cassio@cassio-Inspiron-7580: ~/Downloads/minisat/core

45 |     printf("conflict literals      : %-12"PRIu64"    (%4.2f %% deleted)\n"
, solver.tot_literals, (solver.max_literals - solver.tot_literals)*100 / (double
)solver.max_literals);
|
In file included from /home/cassio/Downloads/minisat/core/Dimacs.h:27,
                 from /home/cassio/Downloads/minisat/core/Main.cc:29:
/home/cassio/Downloads/minisat/core/SolverTypes.h:50:16: error: friend declarati
on of 'Minisat::Lit mkLit(Minisat::Var, bool)' specifies default arguments and i
sn't a definition [-fpermissive]
50 |     friend Lit mkLit(Var var, bool sign = false);
|
/home/cassio/Downloads/minisat/core/SolverTypes.h:58:14: error: friend declarati
on of 'Minisat::Lit Minisat::mkLit(Minisat::Var, bool)' specifies default argume
nts and isn't the only declaration [-fpermissive]
58 | inline Lit mkLit    (Var var, bool sign) { Lit p; p.x = var + var + (
int)sign; return p; }
|
/home/cassio/Downloads/minisat/core/SolverTypes.h:50:16: note: previous declarat
ion of 'Minisat::Lit Minisat::mkLit(Minisat::Var, bool)'
50 |     friend Lit mkLit(Var var, bool sign = false);
|
make: *** [/home/cassio/Downloads/minisat/mtl/template.mk:72: /home/cassio/Downl
oads/minisat/core/Main.o] Erro 1
cassio@cassio-Inspiron-7580:~/Downloads/minisat/core$
```

# Corrigir linhas

Comentar linha 50 e colocar o *default* false no sign da linha 58

```
49 // Use this as a constructor:
50 friend Lit mkLit(Var var, bool sign = false);
51
52 bool operator == (Lit p) const { return x == p.x; }
53 bool operator != (Lit p) const { return x != p.x; }
54 bool operator < (Lit p) const { return x < p.x; } // '<' makes p, ~p adjacent in the order
55 };
56
57
58 inline Lit mkLit (Var var, bool sign) { Lit p; p.x = var + var + (int)sign; return p; }
59 inline Lit operator ~(Lit p)
```

# Explicação sobre o .cnf

p cnf NÚMERO\_DE\_VARIÁVEIS NÚMERO\_DE\_CLÁUSULAS

$$(p_1 \vee \neg p_5 \vee p_4) \wedge (\neg p_1 \vee p_5 \vee p_3 \vee p_4) \wedge (\neg p_3 \vee \neg p_4)$$

pode ser expressa com o seguinte arquivo `exemplo.cnf`

```
c Esse é um comentário
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

# Corrigir linha

Corrigir o caminho do minisat no header *codebreaker.h*

ar Executar Terminal Ajuda

C mastermind.c

C codebreaker.c

C codebreaker.h X

C codebreaker.h > ...

```
1  #ifndef CODEBREAKER_H
2  #define CODEBREAKER_H
3
4
5  #define MAX_CLAUSES 2000
6  #define SOLVER_CALL "/home/mfinger/SparkleShare/143.107.45.102/cursos/239/2022/EPs/EP1/minisat-master/bin/minisat"
7  #define SOLVER_RESP "/tmp/solver_resp.txt"
8
9  struct CLAUSE {
10     int plits;
```

# Dúvidas e monitoria