

# 赤黒木(rb\_tree/red black tree)を表示する秀丸マクロ

## 概要

昔々、オレオレ STL を開発していた頃に、自作の赤黒木をデバッグするために使用した秀丸マクロです。

VisualStudio 附属の STL (Dinkumware 社製) の赤黒木をダンプ表示します。

赤黒木を自作されている方の助けになれば幸いです。

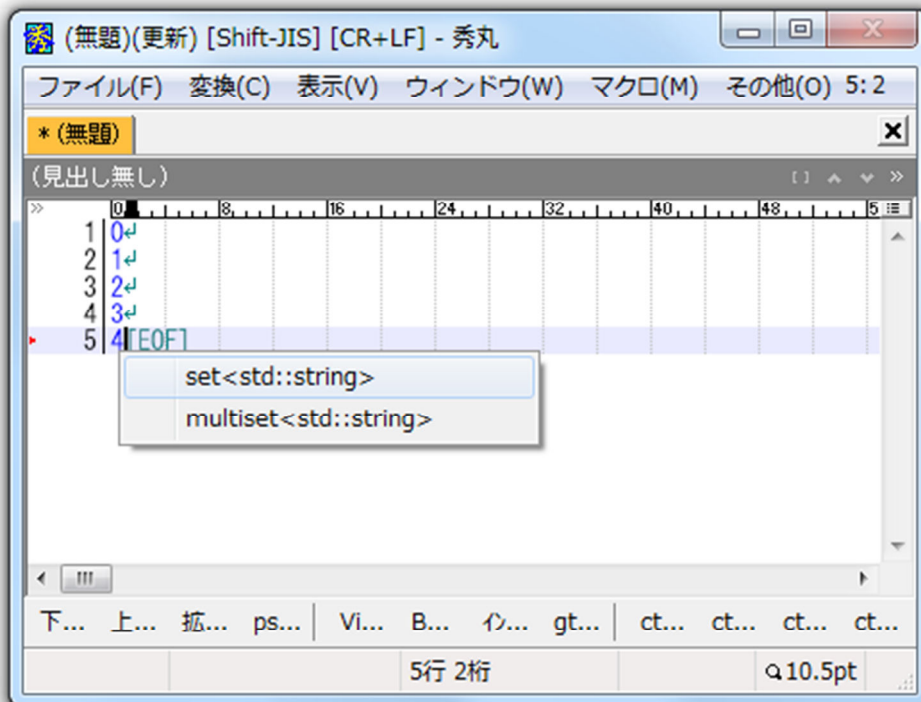
※C++の仕様書には *set/map* の内部実装が赤黒木であるとは書いてなかったはずですが、多くの STL が内部実装に赤黒木を使用しているので、ここでは *set/map* の内部実装は赤黒木であるものとします。

## 動作イメージ（その1）

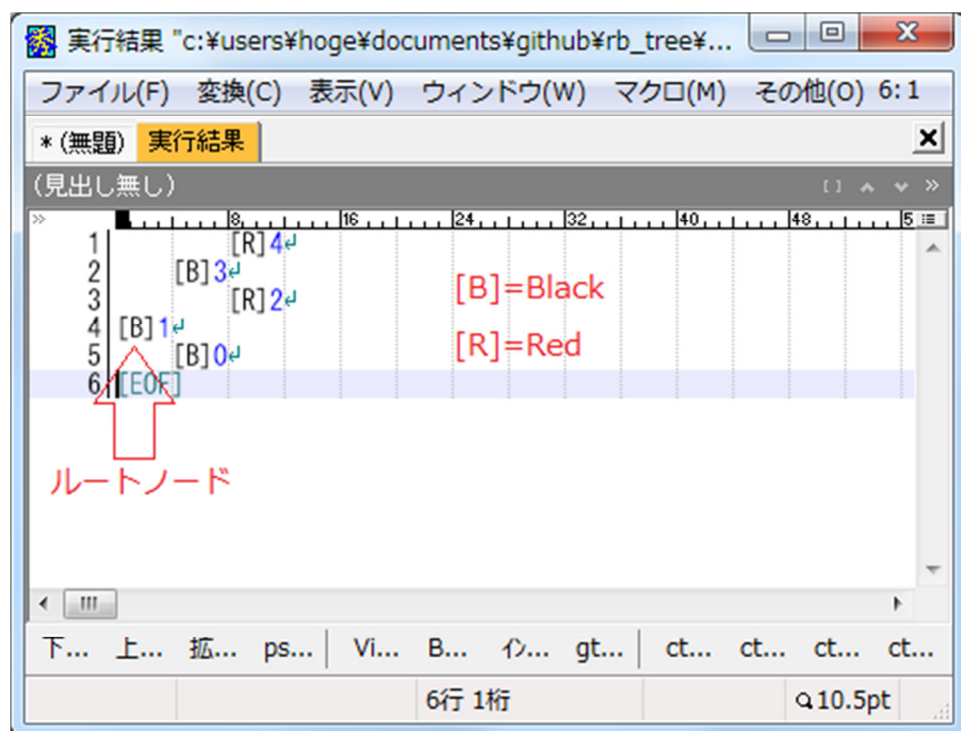
赤黒木に登録する文字を並べます。



マクロを起動するとメニューが開きます。ここでは `set<std::string>` を選択します。

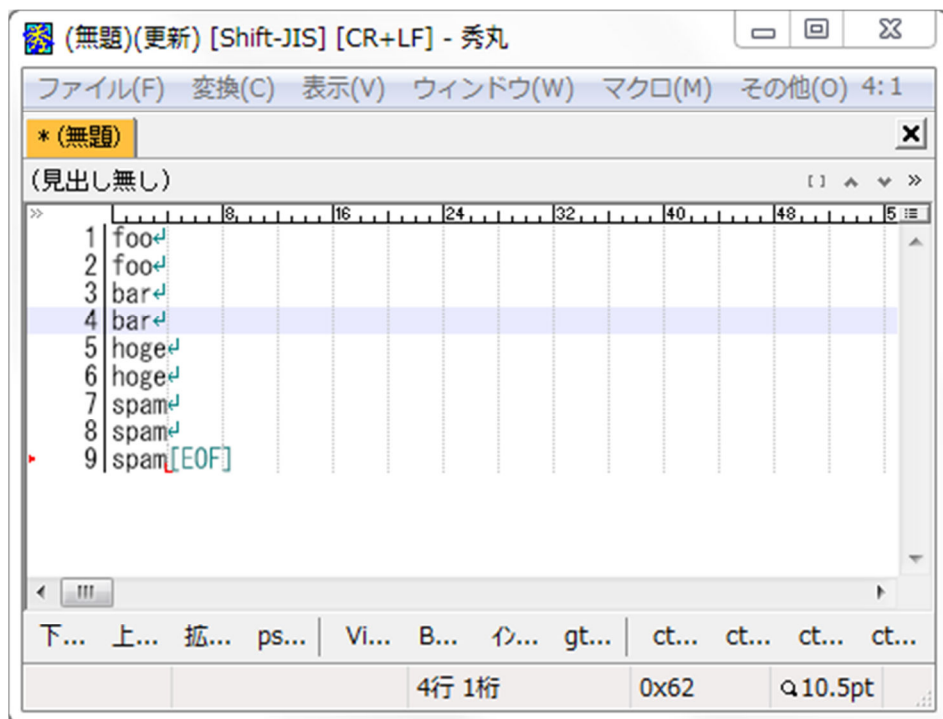


新しいタブに赤黒木の状態がダンプ表示されます。

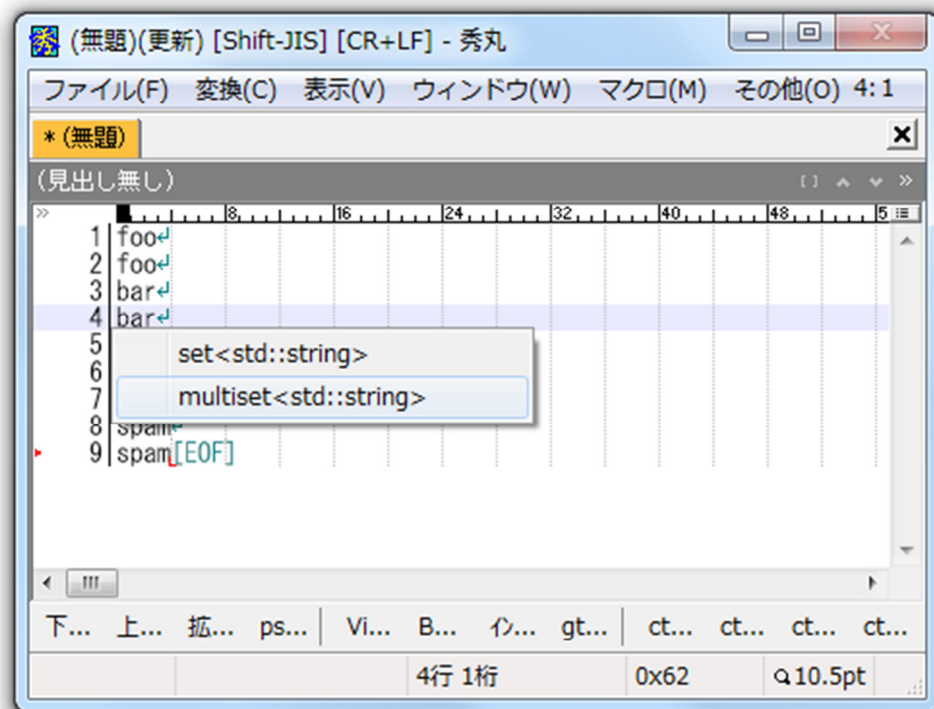


## 動作イメージ（その2）

赤黒木に登録する文字を並べます、ここでは重複する文字があります。



メニューから `multiset<std::string>` を選択します。



赤黒木に重複を許して登録されています。

実行結果 "c:\users\hoge\documents\github\rb\_tree¥..."

ファイル(F) 変換(C) 表示(V) ウィンドウ(W) マクロ(M) その他(O) 10: 1

\* (無題) 実行結果

(見出し無し)

	8	16	24	32	40	48	56
1		[R] spam					
2		[B] spam					
3		[R] spam					
4		[B] hoge					
5	[B] hoge						
6		[B] foo					
7		[R] foo					
8			[R] bar				
9			[B] bar				
10							[EOF]

重複を許して登録されます。

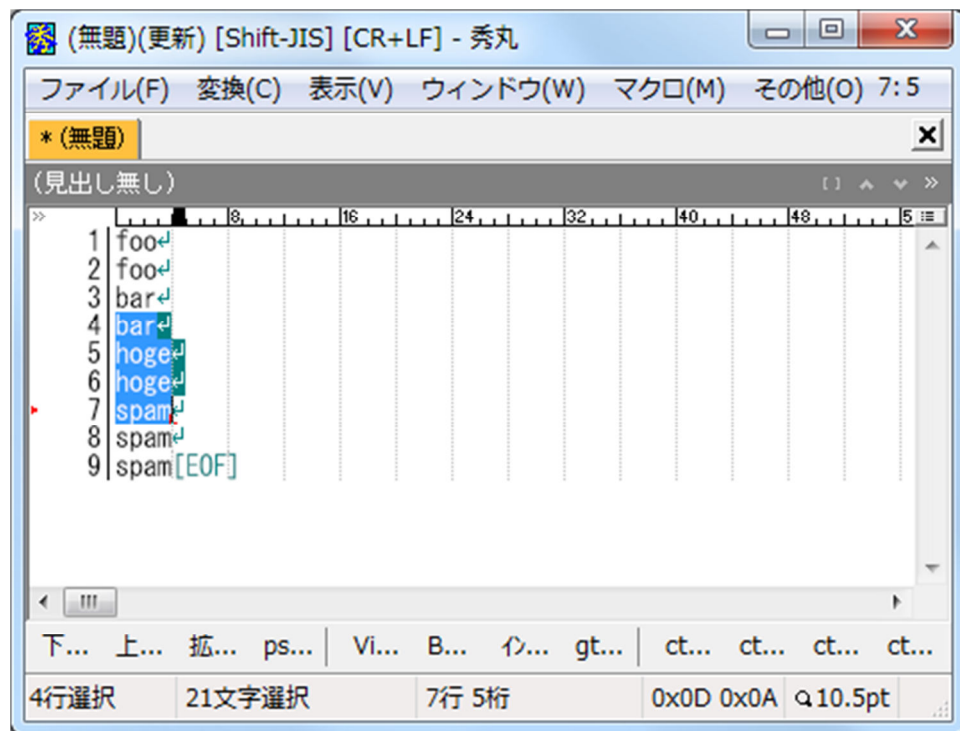
下... 上... 拡... ps... Vi... B... ｲ... gt... ct... ct... ct... ct...

10行 1桁

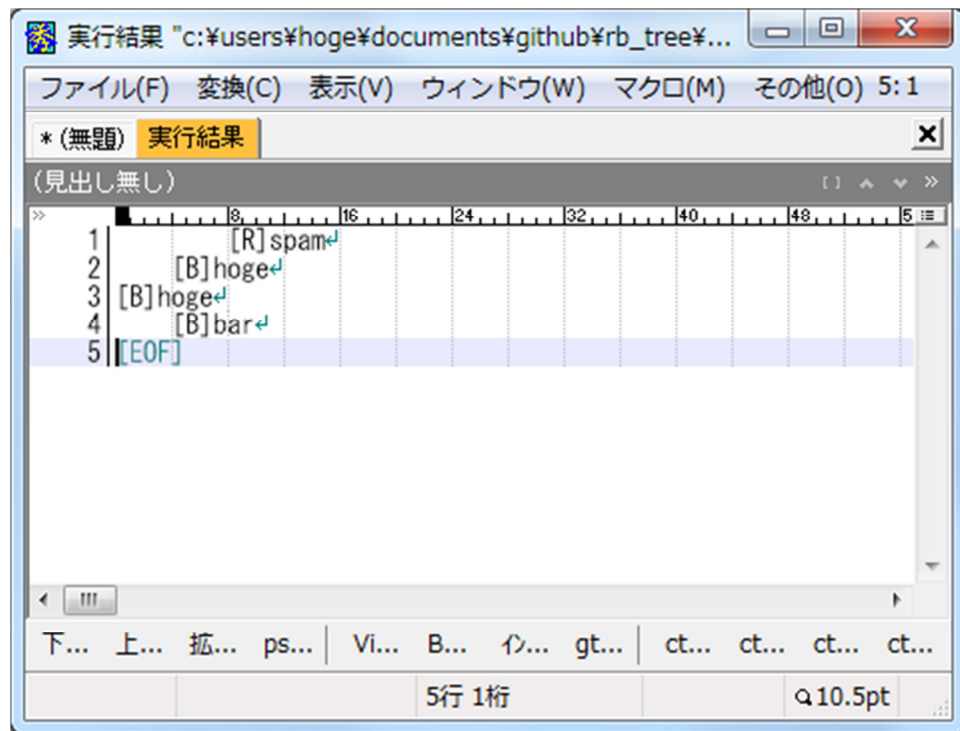
Q 10.5pt

## 選択範囲について

選択範囲があればその範囲のみを対象にします。

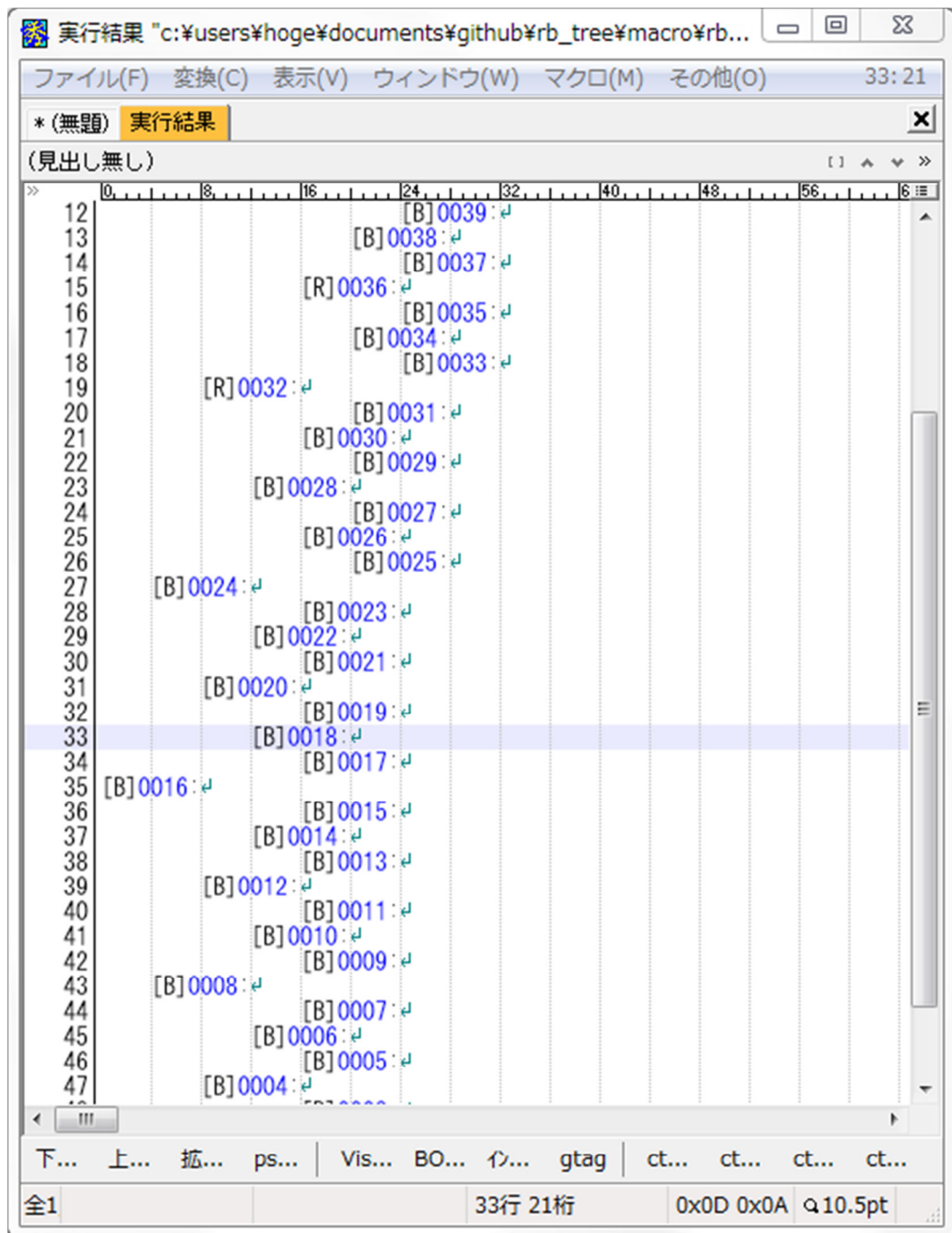


マクロ実行後（選択した4行が登録されています。）

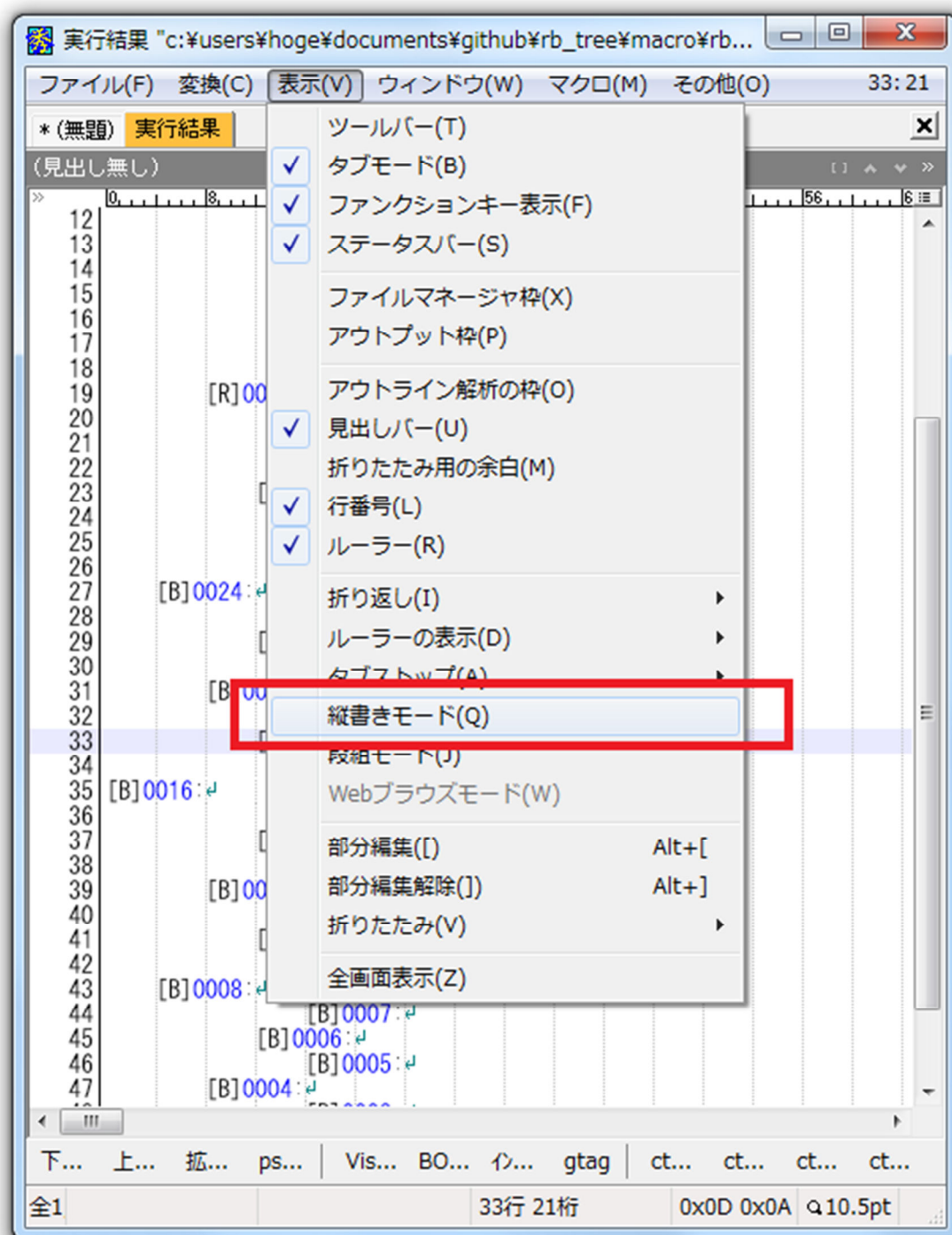


ちょっとした使いこなし

木が上下に長く見にくいときは

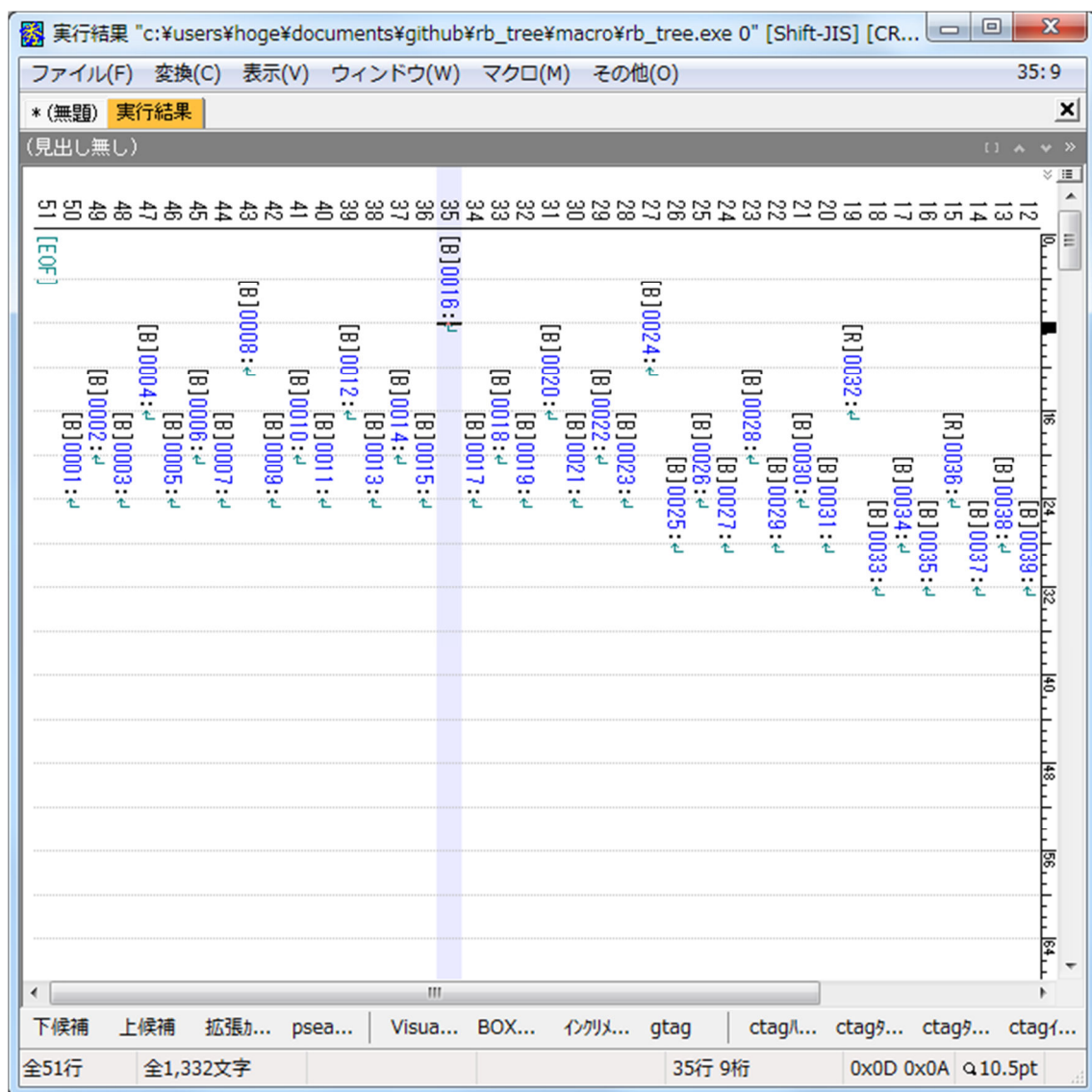


縦書きにすると・・・





多少見やすくなります。



本当に多少ですが・・・