

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。

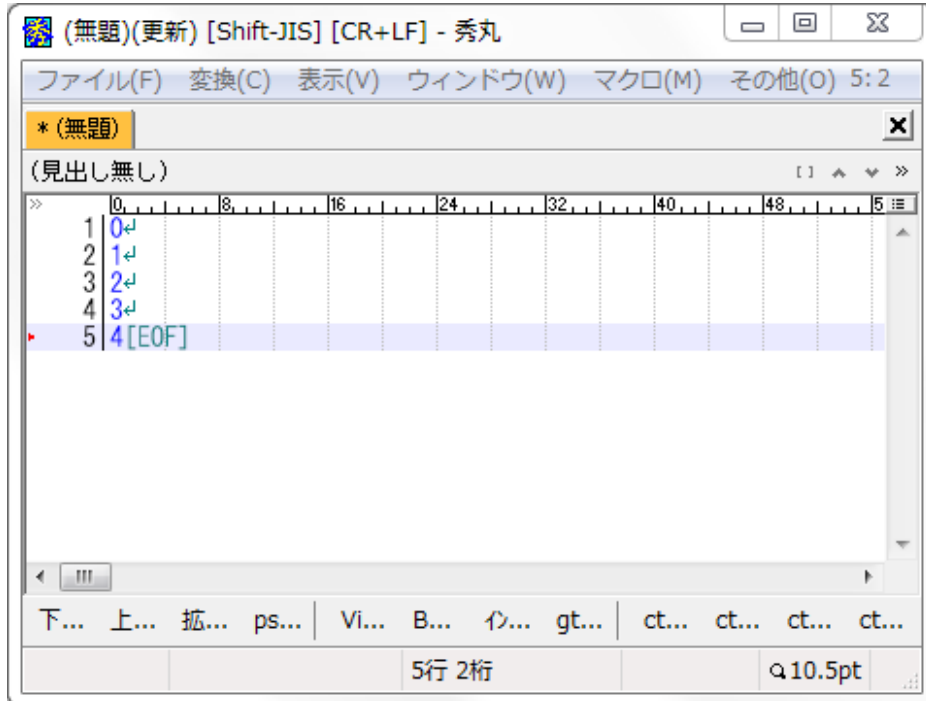
VisualStudio 附属の STL (Dinkumware 社製) の赤黒木をダンプ表示します。

赤黒木を自作されている方の助けになれば幸いです。

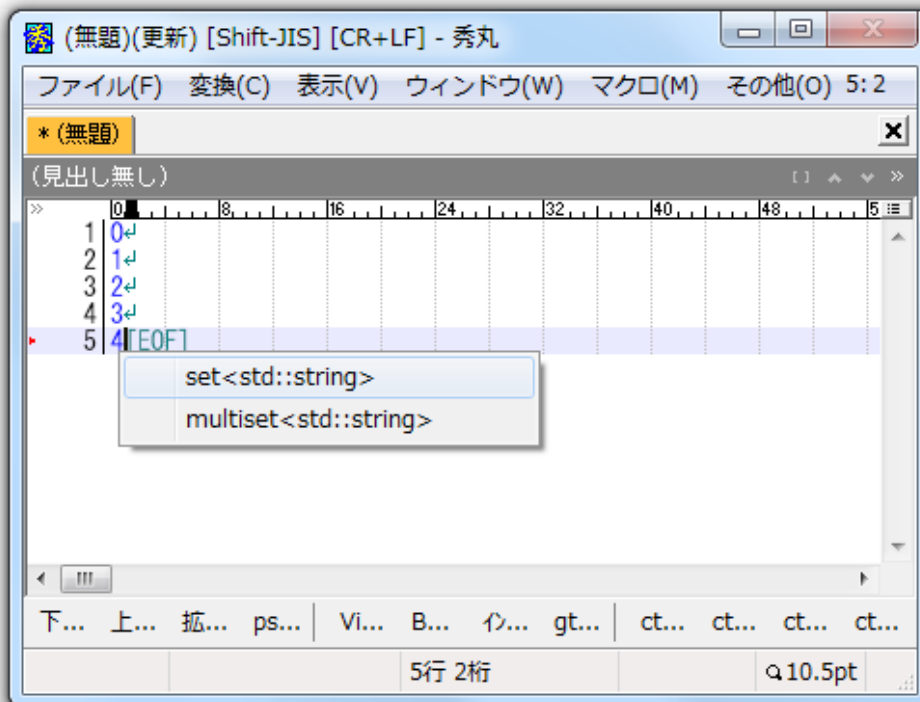
※C++の仕様書には *set/map* の内部実装が赤黒木であるとは書いてなかったはずですが、多くの STL が内部実装に赤黒木を使用しているので、ここでは *set/map* の内部実装は赤黒木であるものとします。

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。

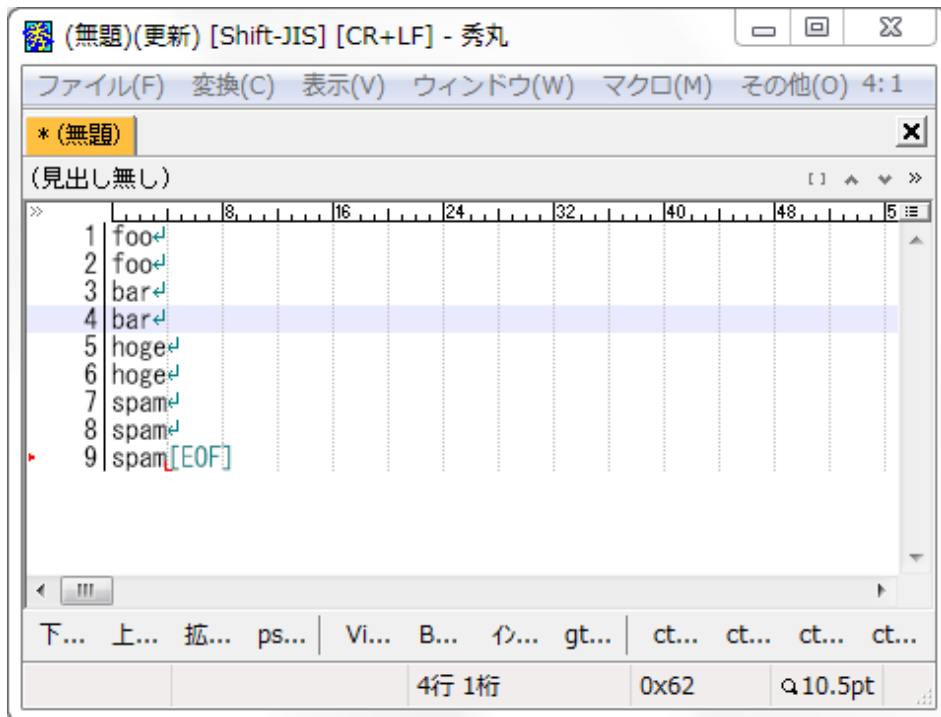
```
1 [R] 4
2 [B] 3
3 [R] 2
4 [B] 1
5 [B] 0
6 [EOF]
```

[B]=Black
[R]=Red

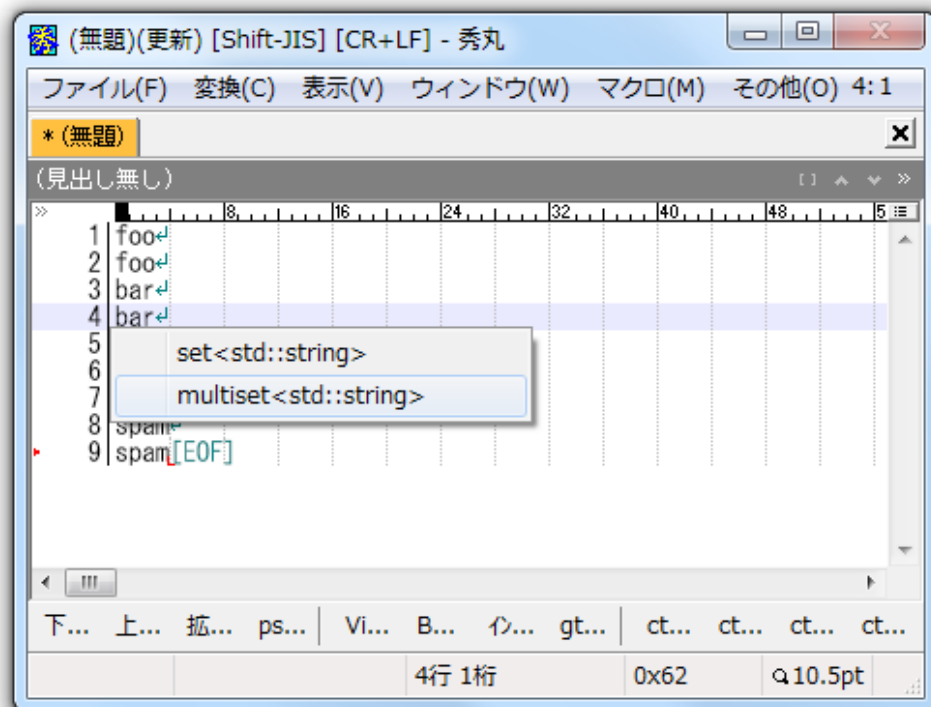
ルートノード

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。

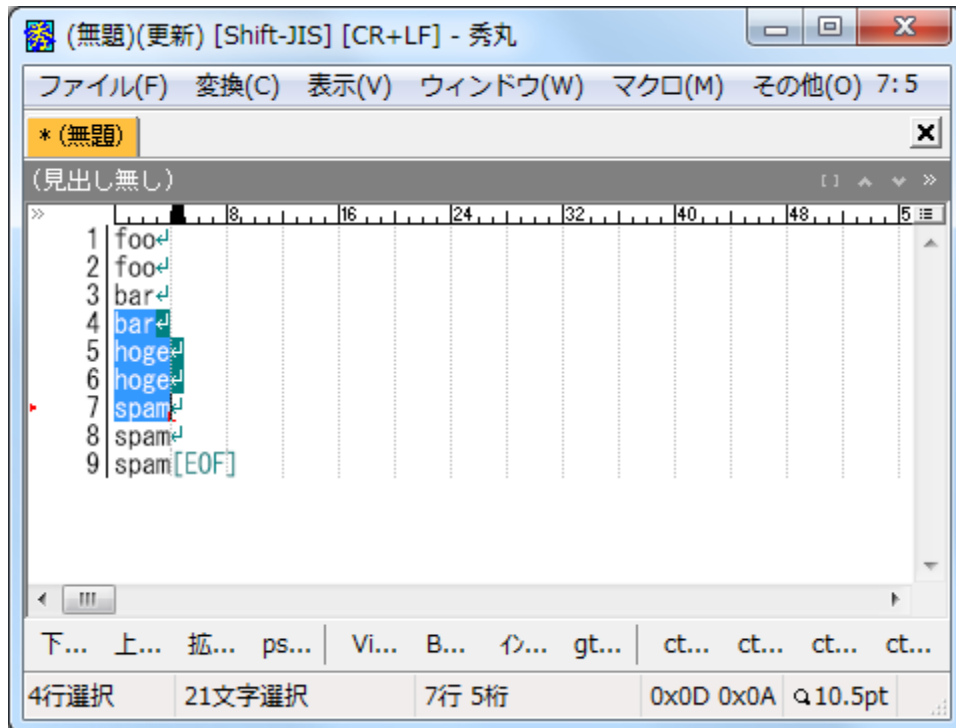
```
1 [R] spam
2 [B] spam
3 [R] spam
4 [B] hoge
5 [B] hoge
6 [B] foo
7 [R] foo
8 [R] bar
9 [B] bar
10 [EOF]
```

重複を許して登録されます。

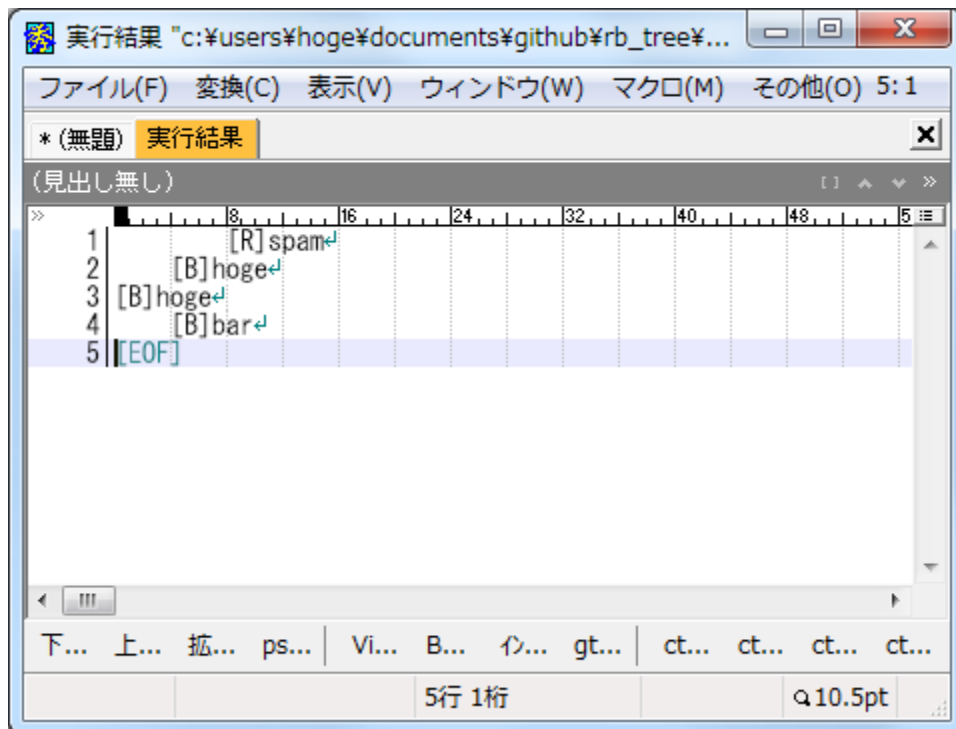
10行 1桁 Q 10.5pt

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

実行結果 "c:\users\hoge\documents\github\rb_tree\macro\rb..." 33: 21

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)

* (無題) 実行結果

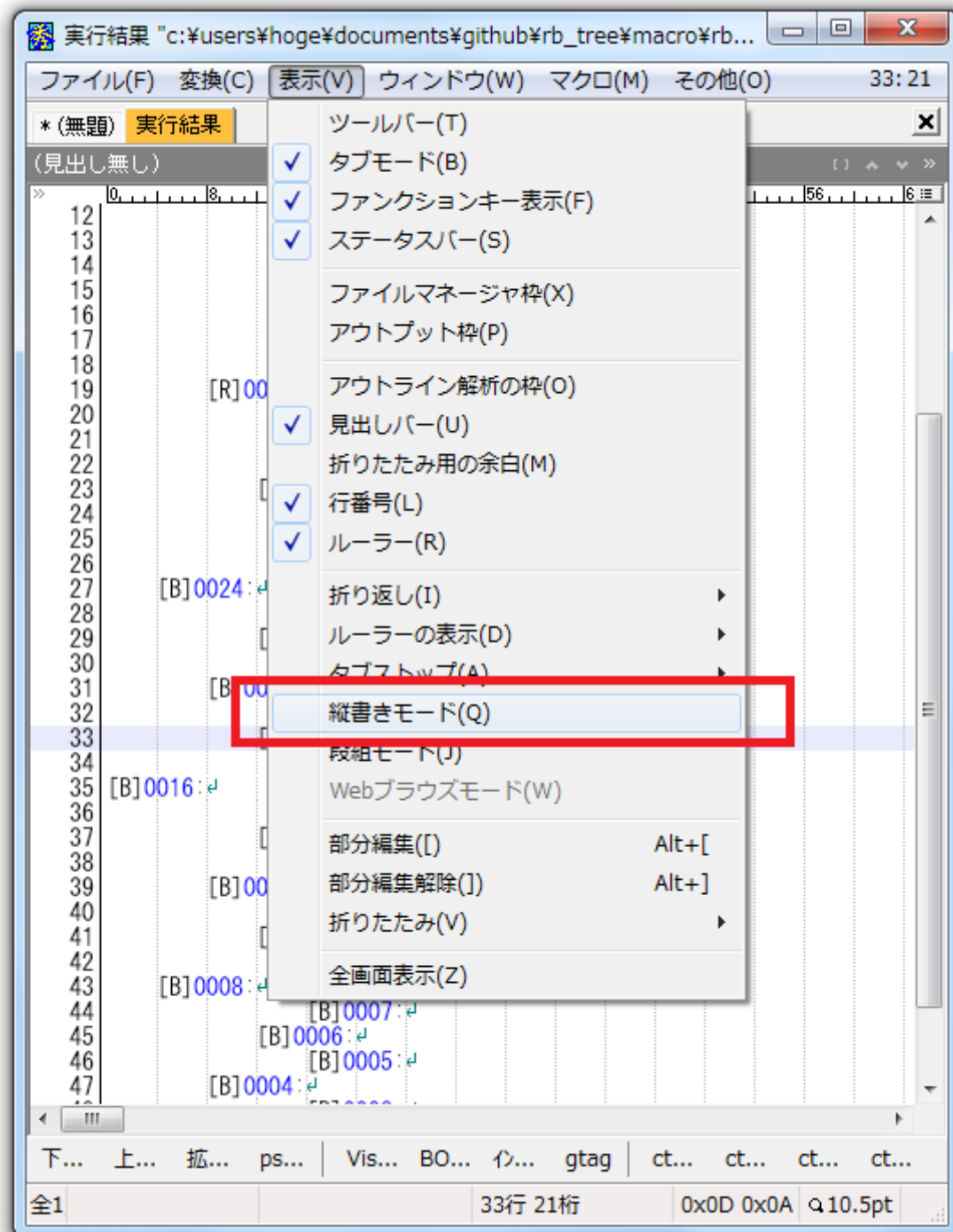
(見出し無し)

Address	Value
12	[B] 0039
13	[B] 0038
14	[B] 0037
15	[R] 0036
16	[B] 0035
17	[B] 0034
18	[B] 0033
19	[R] 0032
20	[B] 0031
21	[B] 0030
22	[B] 0029
23	[B] 0028
24	[B] 0027
25	[B] 0026
26	[B] 0025
27	[B] 0024
28	[B] 0023
29	[B] 0022
30	[B] 0021
31	[B] 0020
32	[B] 0019
33	[B] 0018
34	[B] 0017
35	[B] 0016
36	[B] 0015
37	[B] 0014
38	[B] 0013
39	[B] 0012
40	[B] 0011
41	[B] 0010
42	[B] 0009
43	[B] 0008
44	[B] 0007
45	[B] 0006
46	[B] 0005
47	[B] 0004

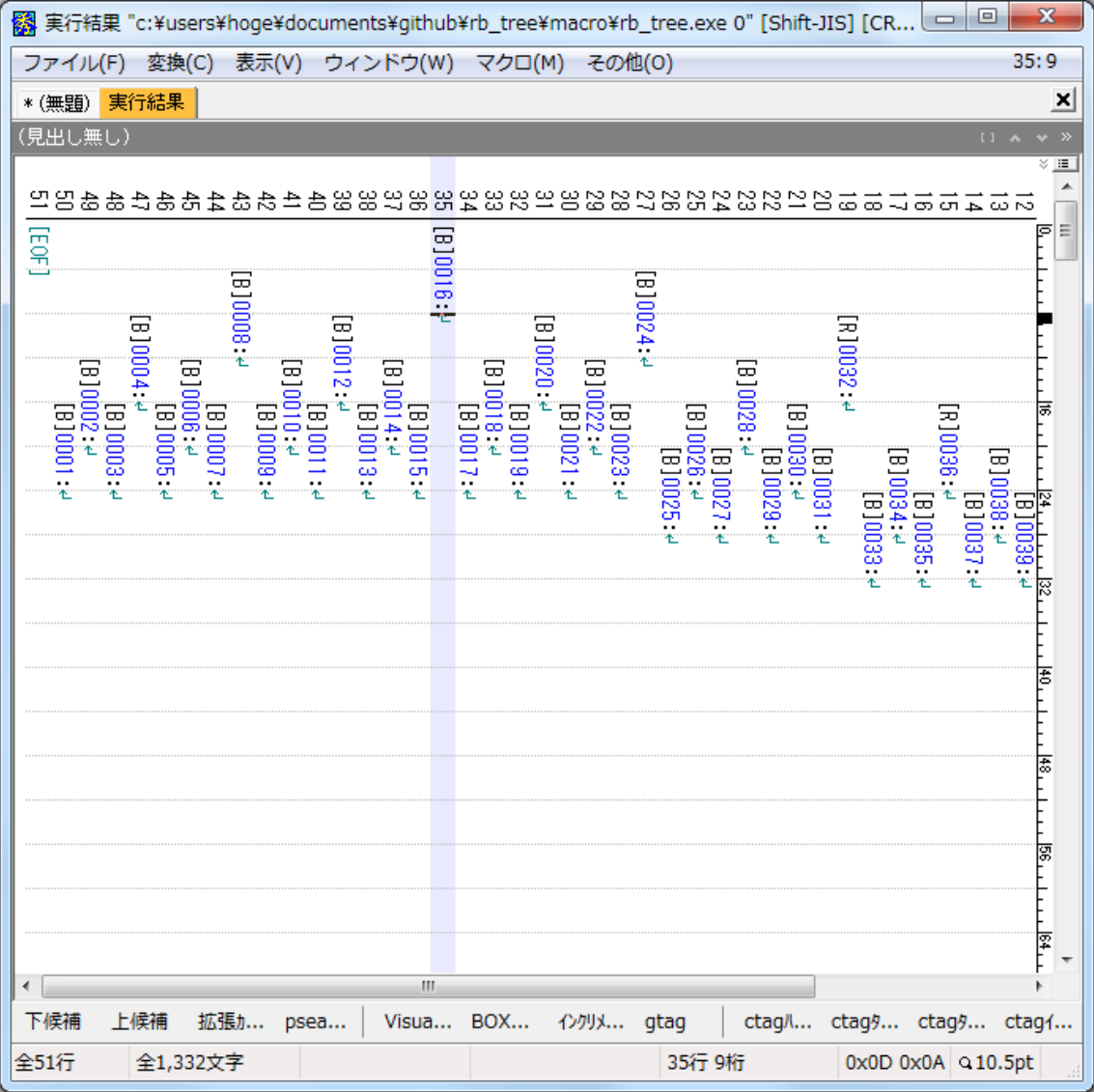
下... 上... 拡... ps... Vis... BO... ｲ... gtag | ct... ct... ct... ct...

全1 33行 21桁 0x0D 0x0A Q 10.5pt

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・・・