

## MIPS BINARY SEARCH TREE IMPLEMENTATION REPORT

Firstly I want to mention that, it may be my bad but I have tried to use specific order for argument registers. I have used \$a0 for number values. I have used \$a1 to store list address to build the tree. I have used \$a2 for tree root pointer variable. I have used \$a3 for queue operations. And I have modified test file you have uploaded to fit my needs.

I have a mips\_bst.asm file and a test.asm file. In mips\_bst.asm file there is whole program with menu. In test file there is also necessary functions to test the program.

Also I want to mention that I wanted to store the temporary registers to stack pointer as well.

I have used two queue solution to print the tree using level order.

In my mips\_bst.asm program, firstly you will see a menu. Build function is not included in the menu. Build function is executed before menu appears. You can change the list on demand in the mips\_bst.asm file. Register and data base should be decimal for this program.

-9999 is used for ending the build in the list. And -10000 is used in queue to print X.

In the menu there are three options. Insert, find and print. Also you can exit the program on demand. If the value already exist in the tree, insert function does not add the value. Insert function allocates memory and tries to find the correct position to insert the value. There are allocate memory for tree and allocate memory for queue functions. In find function it looks for the value starting from the root. If it finds it prints the address, if it could not find it prints not found message.

In print level order function I have used two queues. Every queue represent one level of the tree. While popping from one queue it appends the other queue, that the children of that level. And passes the other queue to print the lower level when the current queue is empty.

```
Activities QtSpim Ara 6 17:37 Console
For print: 3
1
What is the value to be inserted
35
Menu
Please enter a value
To exit 0
For insert: 1
For find: 2
For print: 3
3
50
25 75
12 33 66 83
X X X 35 X X X X
X X
Menu
Please enter a value
To exit 0
For insert: 1
For find: 2
For print: 3
1
What is the value to be inserted
15
Menu
Please enter a value
To exit 0
For insert: 1
For find: 2
For print: 3
3
50
25 75
12 33 66 83
X 15 X 35 X X X X
X X X X
```

```
Activities QtSpim Ara 6 17:35 Console
Menu
Please enter a value
To exit 0
For insert: 1
For find: 2
For print: 3
3
50
25 75
12 33 66 83
X X X X X X X X
Menu
Please enter a value
To exit 0
For insert: 1
For find: 2
For print: 3
2
What is the value to find
85
The number is NOT FOUND in the tree
Menu
Please enter a value
To exit 0
For insert: 1
For find: 2
For print: 3
2
What is the value to find
12
The number is found at the address 268697648
Menu
Please enter a value
To exit 0
For insert: 1
For find: 2
For print: 3
|
```

Activities QtSpim Ara 6 17:32

QtSpim Console

Text Segment Data Segment Window Help

Data Text

Text

the root

```
[00400054] 3c011001 lui $1, 4097 [buildTest] ; 51: 2 X 6 X 13
[00400058] 34260086 ori $6, $1, 134 [buildTest] ; 52: 2 X X 11 X
[0040005c] 0c100034 jal 0x004000d0 [assertEquals]; 53: 2 X X
[00400060] 3404000b ori $4, $0, 11 ; 56: 1
[00400064] 3c011001 lui $1, 4097 ; 57: 1
[00400068] 8c2600b0 lw $6, 176($1)
[0040006c] 3c051001 lui $5, 4097 [firstList] ; 58: 1
[00400070] 0c100082 jal 0x00400208 [insert_beginning
[00400074] 8c450000 lw $5, 0($2) ; 61: 1
address
[00400078] 3c011001 lui $1, 4097 [insertTest]; 62: 1
[0040007c] 34260092 ori $6, $1, 146 [insertTest]
[00400080] 0c100034 jal 0x004000d0 [assertEquals]; 63: 1
[00400084] 3404000b ori $4, $0, 11 ; 67: 1
[00400088] 00103021 addu $6, $0, $16 ; 68: 1
root at $a2
[0040008c] 0c1000c3 jal 0x0040030c [find] ; 69: 1
[00400090] 8c650000 lw $5, 0($3) ; 71: 1
[00400094] 3c011001 lui $1, 4097 [findTest] ; 72: 1
[00400098] 3426009f ori $6, $1, 159 [findTest]
[0040009c] 0c100034 jal 0x004000d0 [assertEquals]; 73: 1
[004000a0] 3404002c ori $4, $0, 44 ; 79: 1
[004000a4] 00103021 addu $6, $0, $16 ; 80: 1
[004000a8] 0c1000c3 jal 0x0040030c [find] ; 81: 1
[004000ac] 00022021 addu $4, $0, $2 ; 82: 1
[004000b0] 34050001 ori $5, $0, 1 ; 83: 1
[004000b4] 3c011001 lui $1, 4097 [findTest] ; 84: 1
[004000b8] 3426009f ori $6, $1, 159 [findTest]
[004000bc] 0c100034 jal 0x004000d0 [assertEquals]; 85: 1
[004000c0] 00102021 addu $4, $0, $16 ; 88: 1
[004000c4] 0c1000e7 jal 0x0040039c [print_level_orde
[004000c8] 3402000a ori $2, $0, 10 ; 94: 1
[004000cc] 0000000c syscall ; 95: 1
```

1 Build test passed  
2 Insert test passed  
3 Find test passed  
4 Find test passed

8  
3 10  
X 6 X 13  
X X 11 X  
X X  
; 56: 1  
; 57: 1  
; 58: 1  
; 61: 1  
; 62: 1  
; 63: 1  
; 67: 1  
; 68: 1  
; 69: 1  
; 71: 1  
; 72: 1  
; 73: 1  
; 79: 1  
; 80: 1  
; 81: 1  
; 82: 1  
; 83: 1  
; 84: 1  
; 85: 1  
; 88: 1  
; 94: 1  
; 95: 1