Lab5

Requirement

- Write program with LC-3 assembly language
- Start your program at x3000
- Use recursion to solve the problem
- Remember to halt your program in the end

Algorithm

Construct a binary tree and perform a prior traversal, then check the value we have got.

Q&A

Q:How to reduce the run time.
A:We can check at every point to decide wehther to deep in or not.

Thoughts

The first time using recursion to code, many things to learn.

codes

```
        .ORIG   x3000

        LD  R1,C_NUM
        AND R0,R0,#0
        STR R0,R1,#0    ;Set the check number to 0

        LD  R6,STA_P    ;Set R6 the stack pointer

        JSR INPUT
        LD  R1,N
        STR R0,R1,#0
NEXT    JSR RE
        JSR POINTER


        LDI R4,N
        LD  R3,POS_D
F1      ADD R4,R4,#-1
        BRn MAIN
        JSR INPUT
        STR R0,R3,#0
        JSR INPUT       ;Input the second number
        STR R0,R3,#1
        ADD R3,R3,#2
        BR  F1          ;Input

MAIN    AND R4,R4,#0    ;R4 is the LEVEL of the tree
        JSR SEARCH

        LDI R1,C_NUM
        BRp ED

        LD  R1,POS_D
        ADD R1,R1,#1
        LD  R2,POS_P
        STR R1,R2,#0

        JSR RE
        JSR SEARCH
```

```
ED      LDI R1,N
        LD  R2,POS_P
LOOP    ADD R1,R1,#-1
        BRn FIN
        LDR R3,R2,#0
        LDR R0,R3,#0
        LDR R4,R3,#0
        LD  R3,A
        ADD R4,R4,R3
        BRn SIMPLE
        LD  R0,ONE
        ADD R0,R0,#1
        TRAP    x21
        AND R0,R0,#0
        ADD R0,R0,R4
SIMPLE  LD  R5,ONE
        ADD R0,R0,R5
        TRAP    x21
        LD  R0,SPACE
        TRAP    x21
        ADD R2,R2,#1
        BR  LOOP        ;Output


FIN     TRAP    x25

SEARCH  ADD R6,R6,#-1
        STR R7,R6,#0    ;Push

        LDI R1,N
        NOT R1,R1
        ADD R1,R1,#1
        ADD R1,R1,R4
        BRz CHECK

        ADD R4,R4,#1
        JSR SEARCH      ;0 NEXT

        LDI R1,C_NUM
        BRp RETURN
```

```
        LD  R1,POS_P
        ADD R1,R1,R4
        LD  R2,POS_D
        ADD R2,R2,R4
        ADD R2,R2,R4
        ADD R2,R2,#1
        STR R2,R1,#0    ;Another number

        ADD R4,R4,#1
        JSR SEARCH      ;1 NEXT

        LDI R1,C_NUM
        BRp RETURN

        LD  R1,POS_P
        ADD R1,R1,R4
        LD  R2,POS_D
        ADD R2,R2,R4
        ADD R2,R2,R4
        STR R2,R1,#0    ;Base number

        BR  RETURN

CHECK   LDI R0,N
F6      ADD R0,R0,#-1
        BRn F7
        LD  R1,POS_P
        ADD R1,R1,R0
        LDR R2,R1,#0
        LDR R1,R2,#0
        ADD R1,R1,#-1
        LD  R2,POS_C
        ADD R1,R1,R2
        AND R2,R2,#0
        ADD R2,R2,#1
        STR R2,R1,#0
        BR  F6

F7      LDI R0,N
F8      ADD R0,R0,#-1
        BRn F9
        LD  R1,POS_C
```

```
        ADD R1,R1,R0
        LDR R2,R1,#0
        BRz RETURN
        BR  F8

F9      AND R0,R0,#0
        ADD R0,R0,#1
        STI R0,C_NUM

RETURN  JSR RE

        ADD R4,R4,#-1
        LDR R7,R6,#0
        ADD R6,R6,#1
        RET



RE      ADD R6,R6,#-1
        STR R7,R6,#0
        LDI R1,N
        LD  R2,POS_C
F2      ADD R1,R1,#-1
        BRn F3
        AND R0,R0,#0
        STR R0,R2,#0
        ADD R2,R2,#1
        BR  F2
F3      LDR R7,R6,#0
        ADD R6,R6,#1
        RET             ;Initialize the check

POINTER LDI R0,N
        LD  R1,POS_D
        LD  R2,POS_P
F4      ADD R0,R0,#-1
        BRn F5
        STR R1,R2,#0
        ADD R1,R1,#2
        ADD R2,R2,#1
        BR  F4
F5      RET             ;Set some pointer
```

```
INPUT   TRAP    x20
        TRAP    x21
        LD  R1,N_MASK
        ADD R1,R1,R0
        TRAP    x20
        TRAP    x21
        LD  R2,N_MASK
        ADD R2,R2,R0
        BRn END0
        ADD R1,R1,R1
        AND R0,R0,#0
        ADD R0,R0,R1
        ADD R1,R1,R1
        ADD R1,R1,R1
        ADD R1,R0,R1
        ADD R1,R1,R2
        TRAP    x20
        TRAP    x21
END0    AND R0,R0,#0
        ADD R0,R0,R1
        RET

ONE     .FILL   #48
A       .FILL   #-10
SPACE   .FILL   #32
N_MASK  .FILL   #-48
N       .FILL   x3400
C_NUM   .FILL   x3401   ;Judge whether a solution has been found or not
POS_C   .FILL   x3300   ;Postion of check
POS_D   .FILL   x3200   ;Postion of data
POS_P   .FILL   x3250   ;Postion of pointer
STA_P   .FILL   x3350   ;Pointer of stack
        .END
```