

5COP093 - Lista de Exercícios 29

1. Desenhe a AST para o código a seguir e depois traduza o mesmo para *assembly* de MIPS. Durante a tradução para *assembly*, você pode utilizar registradores virtuais sempre que necessário.

```
int a[MAX];

void quicksort(int m, int n)
{
    int i,j;
    int v,x;
    if (n <= m) return;
    i = m-1; j = n; v = a[n];
    while(1)
    {
        do i=i+1; while (a[i] < v);
        do j=j-1; while (a[j] > v);
        if (i >= j) break;
        x = a[i]; a[i] = a[j]; a[j] = x;
    }
    x = a[i]; a[i] = a[n]; a[n] = x;
    quicksort(m,j);
    quicksort(i+1,n);
}

int main()
{
    int i;
    int j;

    for(j = 0; j < 100; j++)
    {
        for(i = 0; i < MAX; i++)
        {
            a[i] = MAX - i;
        }
        quicksort(0,MAX-1);
    }
}
```

```
    for(i = 0; i < MAX; i++)
    {
        printf("%d ",a[i]);
    }
    printf("\n");

    return 0;
}
```

2. Um algoritmo de ordenação baseado no bubble sort está sendo compilado para uma máquina com processador MIPS. Após a fase de geração de código, o assembly gerado tem o seguinte aspecto:

```
sort:    addi $sp,$sp, -4
        sw $ra, 0($sp)
        add $r2, $a0, $zero
        add $r3, $a1, $zero
        add $r0, $zero, $zero
for1tst: slt $r4, $r0, $r3
        beq $r4, $zero, exit1
        addi $r1, $r0, -1
for2tst: slt $r4, $r1, $zero
        bne $r4, $zero, exit2
        sll $r5, $r1, 2
        add $r5, $r2, $r5
        lw $r6, 0($r5)
        lw $r7, 4($r5)
        slt $r4, $r7, $r6
        beq $r4, $zero, exit2
        add $a0, $r2, $zero
        add $a1, $r1, $zero
        jal swap
        addi $r1, $r1, -1
        j for2tst
exit2:   addi $r0, $r0, 1
        j for1tst
exit1:   lw $ra, 0($sp)
        addi $sp,$sp, 4
        jr $ra
```

Nesta fase de compilação ainda não ocorreu a alocação de registradores, por isso o gerador de código utilizou os registradores virtuais \$r0 até \$r7. Considerando que a alocação deve utilizar somente os registradores \$s0, \$s1, \$t0, \$t1 e \$t2; construa o CFG do código apresentado e realize a alocação de registradores. Se for necessário fazer spill, considere os seguintes valores de custo:

\$r0 e \$r1 = 0,55
\$r2 e \$r3 = 0,50
\$r4 e \$r5 = 0,60
\$r6 e \$r7 = 0,57

Após a alocação, reescreva o código substituindo os registradores virtuais pelos registradores físicos e, caso necessário, com o código de spill e de salvamento/restauração dos registradores.

IMPORTANTE:

- Os registradores `$s0` e `$s1`, caso sejam usados, devem ser salvos na pilha no início do programa e restaurados na saída do mesmo. Lembre-se de alocar o devido espaço na pilha.
- Se os registradores `$t0`, `$t1` e `$t2` estiverem vivos durante uma chamada de função, eles devem ser salvos na pilha antes da chamada e restaurados após a chamada. Lembre-se de alocar o devido espaço na pilha.
- Se algum registrador virtual tiver que ir para a memória em decorrência de spill, ele será endereçado pelo registrador `$fp` e o seu respectivo deslocamento, como `0($fp)`, `4($fp)`, `8($fp)`, ... Lembrando que cada registrador virtual irá ocupar um deslocamento diferente.