

Basic Block and Flow Control

Quicksort code

```
void quicksort (m, n)
    int m, n;
```

```
{
    int i, j;
    int v, k;
    if (n <= m) return;
```

```
    i = m + 1; j = n; v = a[n] → 1 x n
```

```
    while (1) {
```

```
        do
```

```
            j = j + 1; while (a[j] < v);
```

```
        do
```

```
            i = i - 1; while (a[i] > 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);
    }
```



```

1.  $j = m - 1$ 
2.  $j = n$ 
3.  $t_1 = 4 \times n$ 
4.  $v = a[t_1]$ 
5.  $i = i + 1$ 
6.  $t_2 = 4 \times i$ 
7.  $t_3 = a[t_2]$ 
8. if  $t_3 < v$  goto(5)
9.  $j = j - 1$ 
10.  $t_4 = 4 \times j$ 
11.  $t_5 = a[t_4]$ 
12. if  $t_5 > v$  goto(9)
13. if  $i \geq j$  goto(23)
14.  $t_6 = 4 \times i$ 
15.  $x = a[t_6]$ 
16.  $t_7 = 4 \times i$ 
17.  $t_8 = 4 \times j$ 
18.  $t_9 = a[t_8]$ 
19.  $a[t_7] = t_9$ 
20.  $t_{10} = 4 \times j$ 
21.  $a[t_{10}] = x$ 
22. goto(5)
23.  $t_{11} = 4 \times i$ 
24.  $x = a[t_{11}]$ 
25.  $t_{12} = 4 \times i$ 
26.  $t_{13} = 4 \times n$ 
27.  $t_{14} = a[t_{13}]$ 
28.  $a[t_{12}] = t_{14}$ 
29.  $t_{15} = 4 \times n$ 
30.  $a[t_{15}] = x$ 

```

B1
 B2
 B3
 B4
 B5
 B6