

### 3.4. COMPUTING FACTORIAL OF A NUMBER

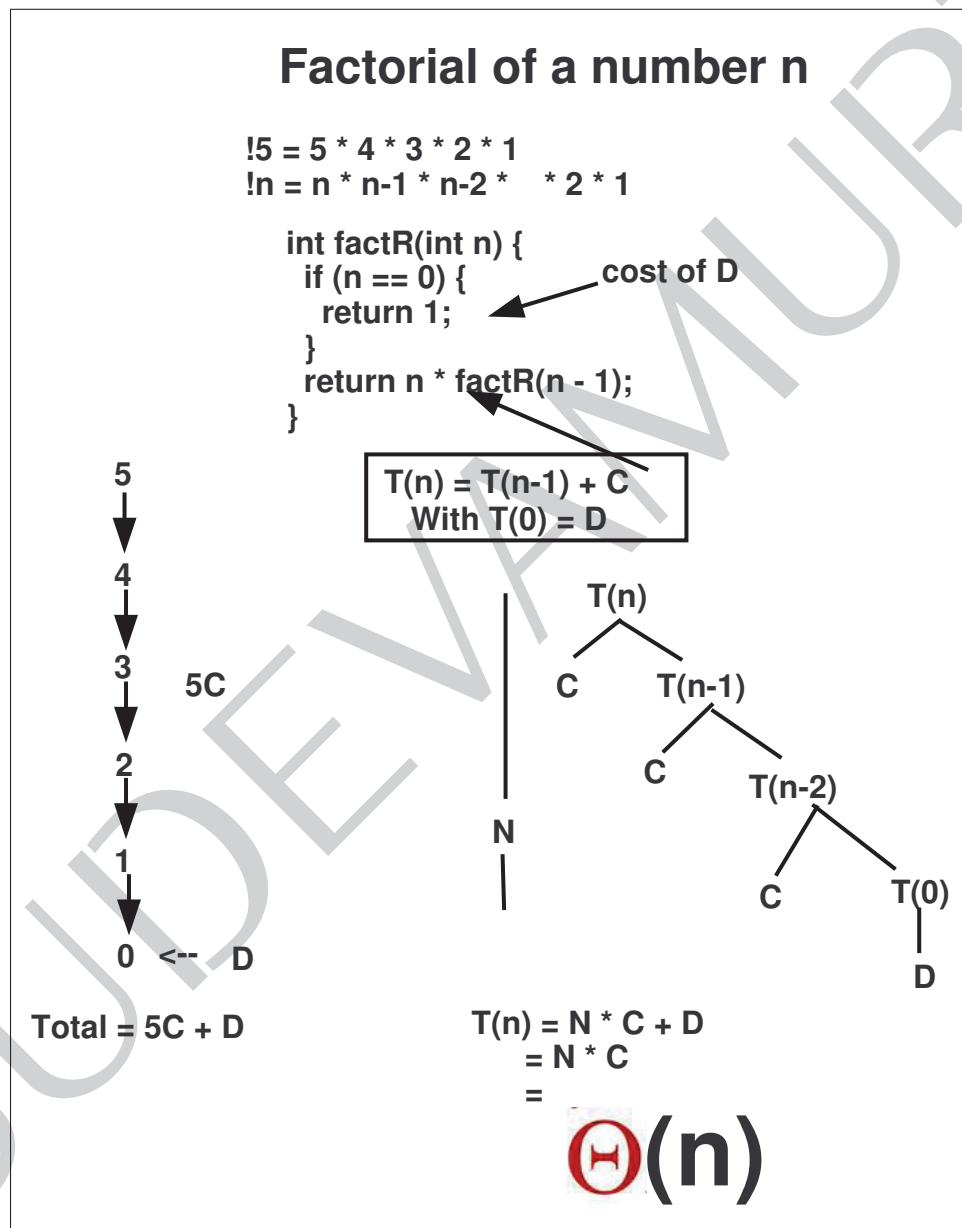


Figure 3.3: Computing  $T(n)$

## 3.4.3 Computing factorial of a number using tail recursion

**Factorial of a number n using Tail recursion**

$$!5 = 5 * 4 * 3 * 2 * 1$$

$$!n = n * n-1 * n-2 * \dots * 2 * 1$$

```
int factTR(int n, int ans) {
    if (n == 0) {
        return ans;
    }
    return factTR(n - 1, ans*n);
}

int factWithTailR(int n) {
    return factTR(n, 1);
}

int r = factWithTailR(5);
```

```
int factTRS(int n) {
    IntStack s = new IntStack();
    int ans = 1;
    s.push(ans);
    while (n > 0) {
        ans = s.pop() * n--;
        s.push(ans);
    }
    return ans;
}

int ans = factTRS(50000);
```

**STACK SIZE OF 1**

**NO Stack overflow**

Figure 3.4: Computing factorial using tail recursion

## 3.5 Computing sum of a number

## 3.5.1 Computing sum of a number using recursion