

CS136 EXERCISE SOLUTIONS: MODULE 7

HOW DO WE DETERMINE RUNTIME OF RECURSIVE CODE?

```
int sum_first(int n) {           // T(n) =
    if (n == 0) return 0;        //      0(1)
    return n + sum_first(n - 1); //      + 0(1) + T(n - 1)
}
// Simplified: T(n) = 0(1) + T(n - 1)
// This is T(n) = 0(1) + T(n - k_1) where k_1 = 1.
```

```
int split_max(int arr[], int len) {           // T(n) =
    if (1 == len) {                           //      0(1)
        return arr[0];                        //
    }                                          //
    int lmax = split_max(arr, len / 2);       //      + T(n / 2)
    int rmax = split_max(arr + len / 2, len - len / 2); // + T(n / 2)
    return lmax > rmax ? lmax : rmax;         //      + 0(1)
}

// T(n) = 0(1) + 2 * T(n / 2)
// Remember that the unusual simplification rules are part of the
// definition of big-O. They don't affect T.
// This is T(n) = 0(1) + k_2 * T(n / k_2) with k_2 = 2.
// From the table, T(n) = O(n).
```