

Time Complexity

- The time, $T(P)$, taken by a program, P , is the sum of its compile time and its run (or execution) time. The compile time is similar to the fixed space component since it does not depend on the instance characteristics. We are really concerned only with the program's execution time, T_p .
- Finding T_p is difficult because it requires a detailed knowledge of the compiler's attributes.
- For example, suppose we have a simple program that adds and subtract numbers. Let n denote the instance characteristic.

$T_p(n) = c_a \text{ADD}(n) + c_s \text{SUB}(n) + c_l \text{LDA}(n) + c_{st} \text{STA}(n)$, where c_a, c_s, c_l, c_{st} are constants that refer to the time needed to perform each operation, and $\text{ADD}, \text{SUB}, \text{LDA}, \text{STA}$ are the no. of additions, subtractions, loads and stores that are performed when the program is run with instance characteristic n . Obtaining such a detailed estimate of running time is rarely worth the effort.

Defn: A program step is a syntactically or semantically meaningful program segment whose execution time is independent of the instance characteristics.

Ex 1.9: We want to obtain the step count for the iterative sum function discussed earlier. Worry only about the executable statements.

```
float sum(float list[], int n)
{
    float tempsum = 0;    count++; /* for assignment */
    int i;
    for(i = 0; i < n; i++) {
        count++;          /* for the for loop */
        tempsum += list[i]; count++; /* for assignment */
    }
}
```