A note on ranges m..n

The notation m..n denotes the list of integers m, m+1, m+2, …, n.

We use this notation in describing a segment of an array:

b[m..n] denotes the list of array elements b[m], b[m+1], …, b[n].

We use this same abbreviation c[m..n] for *any* kind of a list or sequence to represent the part of that list that starts at c[m] and ends at c[n].

The number of values in the range m..n is n+1 – m. The last value of the range is n, and it is followed by n+1. We use this fact to give a mnemonic for remembering the

Formula for the number of values: *the **F**ollower minus the **F**irst*: n+1 – m.

A few examples will make this clear:

2..4 contains 2, 3, 4	That's 4+1 – 2 = 3 values
2..3 contains 2, 3	That's 3+1 – 2 = 2 values
2..2 contains 2	That's 2+1 – 2 = 1values
2..1 contains.	That's 1+1 – 2 = 0 values

The last line is especially noteworthy. Note how it follows the pattern established by the others. It indicates that if the second number (1 in this case) is 1 less than the first number (2), then the range represents the empty set of integers. This may seem weird, but mathematically is it just what we want, and it helps to simplify some assertions used in explaining some algorithms, as you will see later.

Requirement for using of the notation

A requirement of the notation m..n is that m is never bigger than n+1. For example, 2..0 and -3..-6 are not legal ranges. They have no meaning. Whenever you write m..n, it is assumed that m is no bigger than n+1.

Figuring out the formula for the number of values

The formula for the number of values in m..n is *the **F**ollower minus the **F**irst*: n+1 – n. If you forget this formula, you can always figure out by looking at a small example or two. Here are two examples:

The range 5..6 contains 2 values (5 and 6), That's 6+1 – 5: *Follower minus First*.

The range 5..7 contains 3 values (5, 6, and 7), That's 7+1 – 5: *Follower minus First*.

A notational simplification

Suppose b is a Java array. Then b[m..] is an abbreviation for b[m..b.length-1].

Similarly, for a String s, s[m..] is an abbreviation for s[m..s.length()-1].

For *any* list l, l[m..] is the sublist consisting of l[m], l[m+1], … and so on until the end.