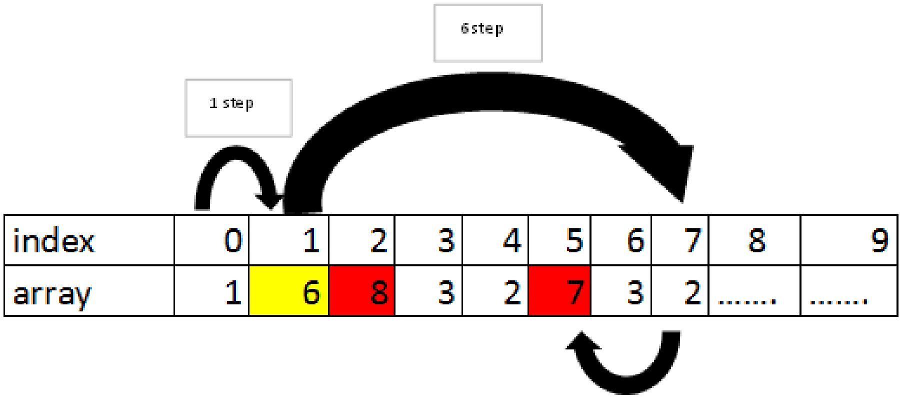
Minimum Jump

First of all, we should look at some cases. If the length of our array is less than 2, we will not jump (return 0;). Or if our any element of our array is equal 0 (except last one), it means that we cannot jump anymore and stuck (return -1;).

As the value of elements of array shows max number of jumps that we can do, we will take a new variable ‘s’ and initialize it with a[0]. Starting from the 2nd element of the array we will make s steps. While going through array s steps, we will remember maximum element of the array in the range (i, i+s). It is clear that we need bigger element but it should be near to the end of our array. For example, we will go s=4 steps through {7, 2, 6, 3}, so we will take as a jump element {6}. So how decide that which element we should take?



index	0	1	2	3	4	5	6	7	8	9
array	1	6	8	3	2	7	3	2	…….	…….

As shown in the picture above we can jump 6 steps. We go through these 6 elements and memorize bigger number among a[i]+i. Here 7+5 > 8 +2. Now our ‘maxx’=7+5=12 and we are in the position j=7 (a[7]). From the picture, it is clear that we already went 2 steps out of 7 steps and there are left 5 steps. So how we can get it? Maxx-j will give us the number of steps that left.