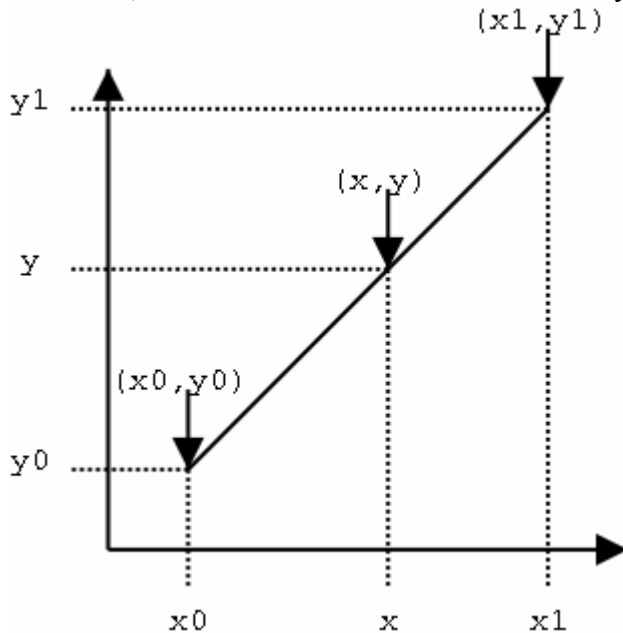


**Interpolation Search:**

The array must be sorted as well.

This time, we estimate the index mathematically.



Let  $x_0 = \text{low}$  ,  $x_1 = \text{high}$  ,  $y_0 = a[\text{low}]$  ,  $y_1 = a[\text{high}]$

The value we are looking for is  $y$  whose index is  $x$ .

We estimate  $x$  by assuming that those 3 points lay at the same straight line.

So,  $\frac{y_1 - y_0}{x_1 - x_0} = \frac{y - y_0}{x - x_0}$  , rearranging the terms, making  $x$  as the subject.

It becomes,  $x = x_0 + \frac{(y - y_0)(x_1 - x_0)}{y_1 - y_0}$  .

We estimate  $x$  with this formula.

Check  $a[x]$ , if  $a[x] > y$  , set  $x_0 = x + 1$  and  $y_0 = a[x + 1]$

Else if  $a[x] < y$  , set  $x_1 = x - 1$  and  $y_1 = a[x - 1]$

Repeatedly use this formula, like binary search.

**Code:**

```
function in_search(var a:arrayA;t,low,high:integer);
begin
  low:=1;
  high:=n;
  while (a[high]>=t) and (key>r[low]) do
  begin
    x:=low+trunc((t-a[low])*(high-low)/(a[high]-a[low]));
    if t>r[x] then low:=x+1
    else if t<r[x] then high:=x-1
    else low:=x;
  end;
  in_search:=a[low]=t;
end;
```