AP COMPUTER SCIENCE A – RECURSIVE BINARY SEARCH

The two examples which follow concern the sorted list of 18 numbers shown at lower left. The variables `b`, `e` and `m` are the current positions of the beginning, end, and middle of the part of the list that currently is being searched. Note that, initially, for `iteration 1`, `b = 0` and `e = 17`. Also, $m = (b + e)/2$. As we discussed last semester, the binary search algorithm has on order of $\log n$ operations, where n is the length of the list. Here, $\log_2 18 = 4.17$ so that the maximum number of iterations possible is 5.

```
i |      i                i
n | l    t                t
d | i    e                e
e | s    r  Example 1:    r  Example 2:
x | t    a  ---------     a  ---------
--+--    t                t
 0 17    i                i
 1 23    o                o
 2 25    n   b   m   e    n   b   m   e
 3 29    ---+--+--+---    ---+--+--+---
 4 33    1   0   8  17    1   0   8  17
 5 37    2   8  12  17    2   8  12  17
 6 41                     3   8  10  12
 7 42                     4  10  11  12
 8 48                     5  10  10  11
 9 54
10 59
11 65
12 73
13 78
14 82
15 85
16 90
17 97
```

Example #1. Here we are searching for `73`. For `iteration 1`, we check if `list[m]`, or `list[8] = 48`, is equal to `73`. It is not, so since `48 < 73`, `b` is redefined as `m = 8`. For `iteration 2`, `list[12] = 73`, so the number `73` is found at index `12` after `2` iterations.

Example #2. Here we are searching for `60`. For `iteration 1`, we check if `list[8] = 48` is equal to `60`. It is not, so since `48 < 60`, `b` is redefined as `m = 8`. For `iteration 2`, we check if `list[12] = 73` is equal to `60`. It is not, so since `60 < 73`, `e` is redefined as `m = 12`. For `iteration 3`, we check if `list[10] = 59` is equal to `60`. It is not, so since `59 < 60`, `b` is redefined as `m = 10`. For `iteration 4`, we check if `list[11] = 65` is equal to `60`. It is not, so since `60 < 65`, `e` is redefined as `m = 11`. For `iteration 5`, we check if `list[10] = 59` is equal to `60`. It is not. Since `b == m`, the process is stopped (and the number has not been found). In general, the process should be stopped if `( b == m ) || ( m == e )`.