

	0	1	2	3	4	5	6	7	8	9
A	10	8	15	32	45	78	20	15	23	10

looking for target = t

for $i = 0; i < A.length; i++ \{$

if $A[i] == t \{$

return $i; // t is at index$

$\}$

$\}$ return $-1; // not found$

returns the index of t in A
or -1 if not found.

if t is found at index 0
 $O(1)$

if t is found at end of array
 $O(n)$

if t is not in array
 $O(n)$

A

2	5	10	15	19	29	40	50	75	101
0	1	2	3	4	5	6	7	8	9

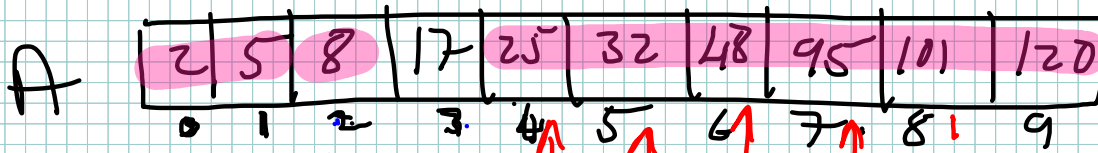
target = π

```

for i = 0; i < A.length; i++ {
  if (A[i] ==  $\pi$ ) {
    return i;
  } else {
    if (A[i] >  $\pi$ ) {
      return -1;
    }
  }
}
return -1;

```

$O(n)$



target = t
= 48

Low = L = 0

High = H = A.length - 1

MIDDLE = M = (L + H) / 2

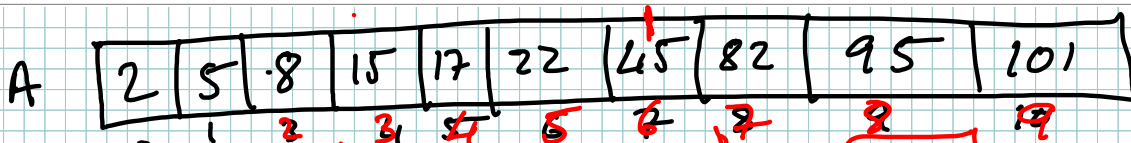
```

while (H >= L) {
  m = (L + H) / 2;
  if (A[m] == t) {
    return m;
  } else if (A[m] > t) {
    H = m - 1;
  } else {
    L = m + 1;
  }
}
return -1;

```

L	H	M
0	9	4
5	6	7
6		5
		6

$O(\log n)$

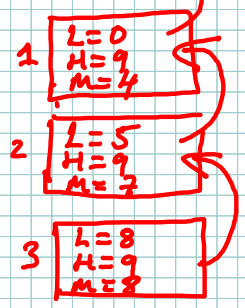


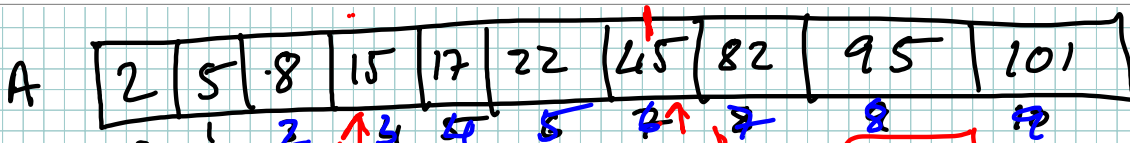
```

int BinarySearch(A, L, H, t) {
    if (H < L) {
        return -1;
    }
    M = (H + L) / 2;
    if (A[M] == t) {
        return M;
    } else if (A[M] > t) {
        return BinarySearch(A, L, M-1, t);
    } else {
        return BinarySearch(A, M+1, H, t);
    }
}
  
```

L H M
0 9 4
5 7

t=82



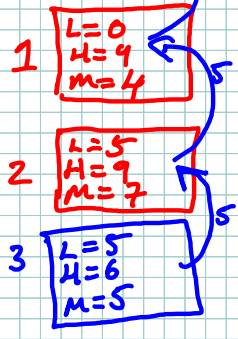


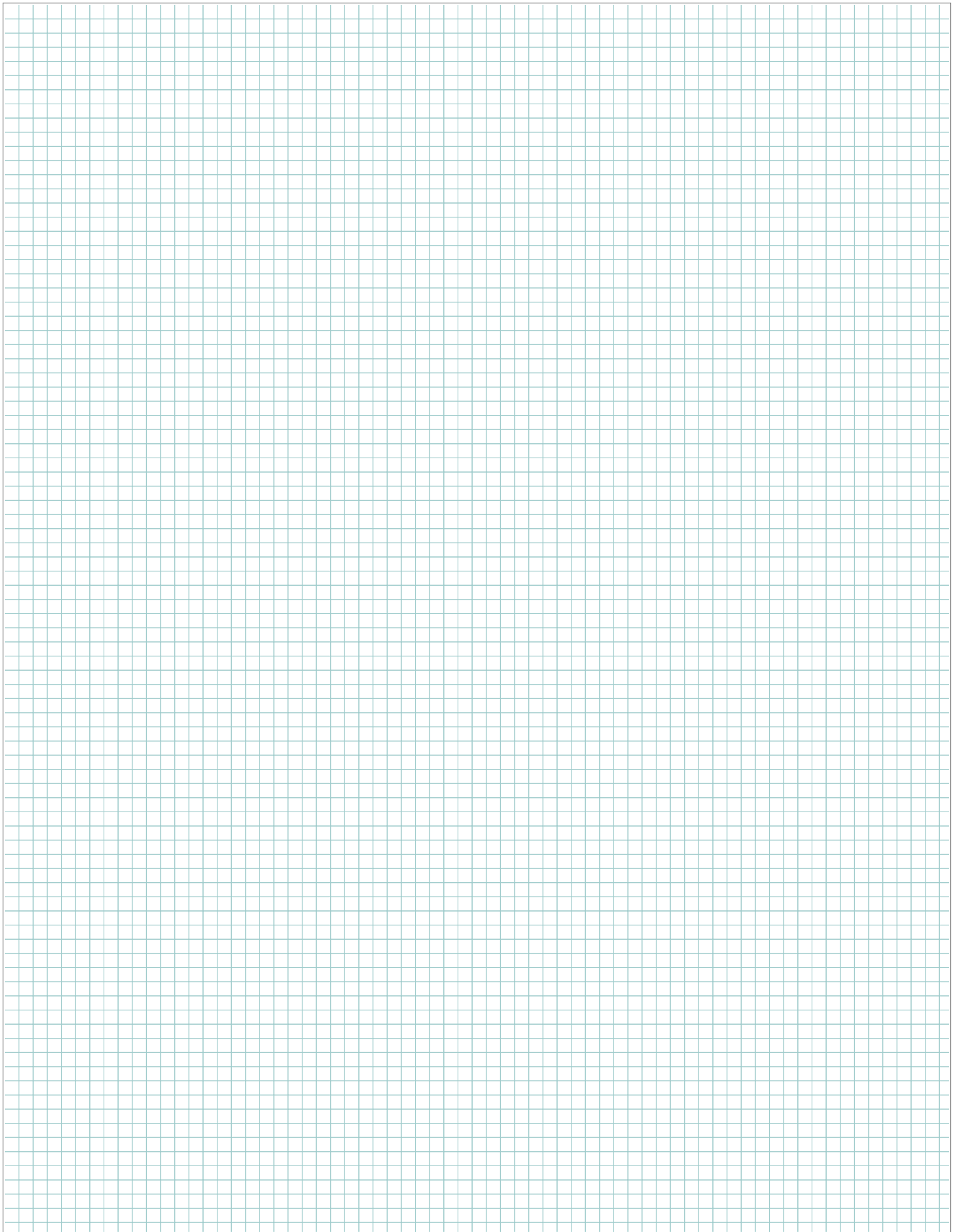
```

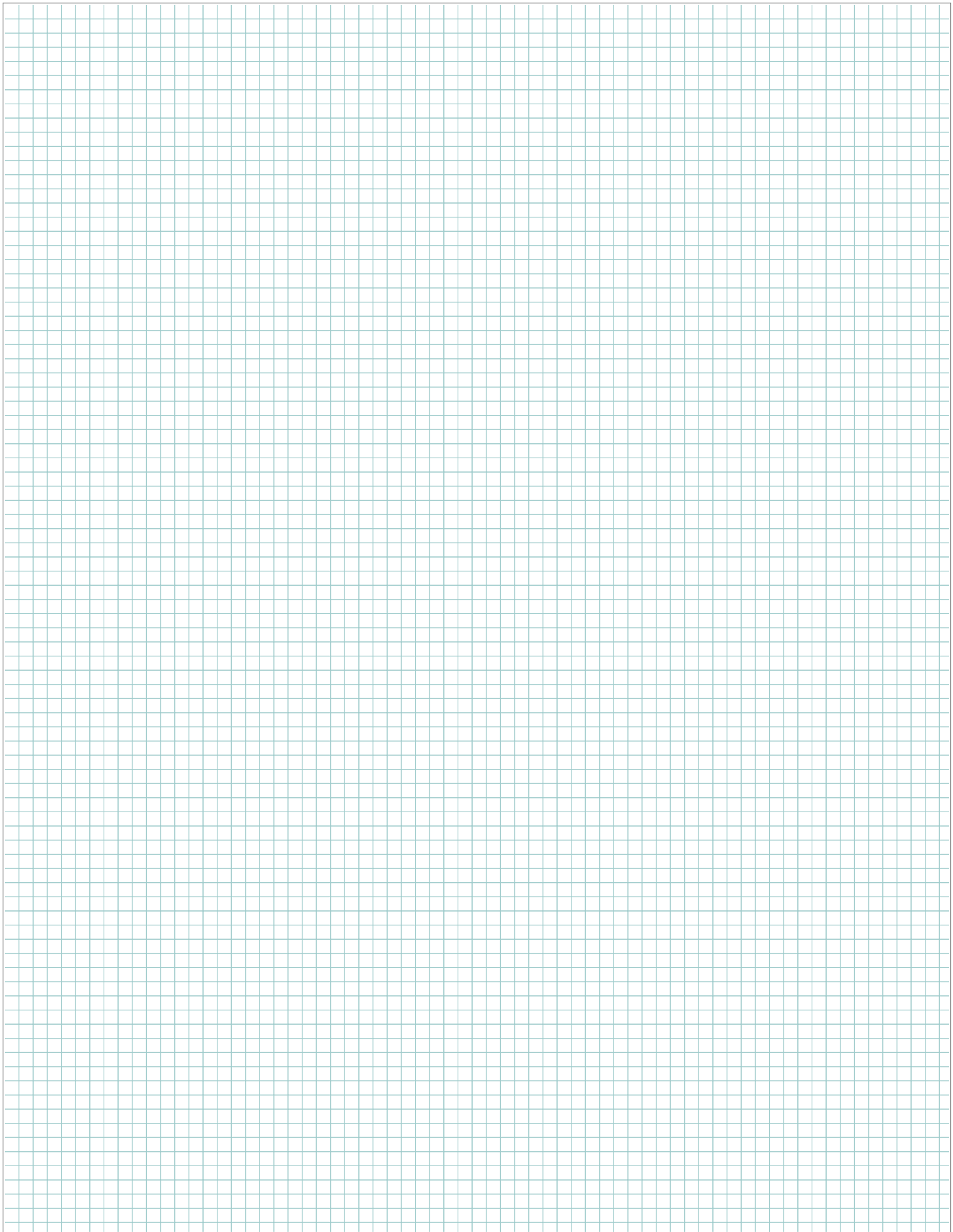
int BinarySearch(A, L, H, t) {
    if (H < L) {
        return -1;
    }
    M = (H + L) / 2;
    if (A[M] == t) {
        return M;
    } else if (A[M] > t) {
        return BinarySearch(A, L, M-1, t);
    } else {
        return BinarySearch(A, M+1, H, t);
    }
}
  
```

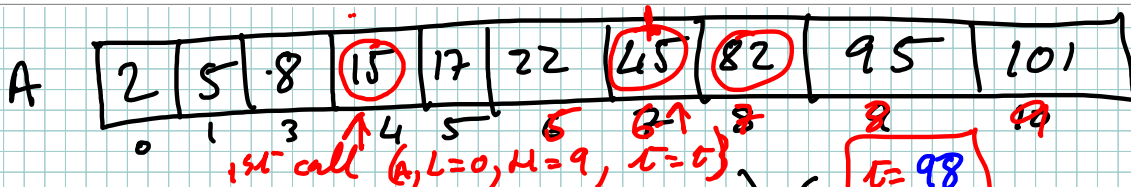
L H M
0 9 4

t=17









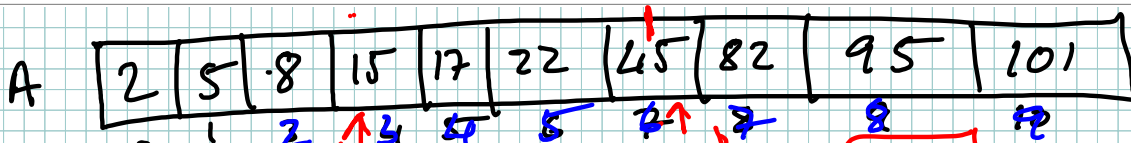
```
int BinarySearch(A, L, H, t) {
    if (H < L) {
        return -1;
    }
    M = (H + L) / 2;
    if (A[M] == t) {
        return M;
    } else if (A[M] > t) {
        return BinarySearch(A, L, M-1, t);
    } else {
        return BinarySearch(A, M+1, H, t);
    }
}
```

L H M
0 9 4

1 L=0
H=9
M=4

2 L=5
H=9
M=7

3 L=8
H=9
M=8



```

int BinarySearch(A, L, H, t) {
    if (H < L) {
        return -1;
    }
    M = (H + L) / 2;
    if (A[M] == t) {
        return M;
    } else if (A[M] > t) {
        return BinarySearch(A, L, M-1, t);
    } else {
        return BinarySearch(A, M+1, H, t);
    }
}
  
```

L H M
0 9 4

t=17

