Binary Search

Idea: halve the space over which we search

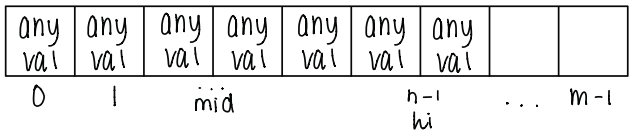
- at each halving, eliminate ½ of remaining possibilities
- max # of steps required is O(log n) ← halving is base 2

Precondition: Must be presorted

Demonstration:

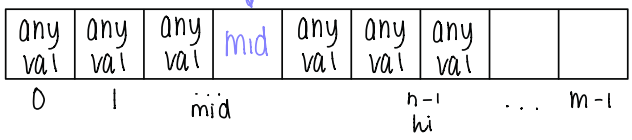
3 indices → lo, mid, hi

1



2

if (x == a[mid]) found



if (x < a[mid]) search below mid

if (x > a[mid]) search above mid

Code analysis:

```
bool binarySearch(int a[], int n, int x)   // n: # of elts; x: looking for
{
    int lo, mid, hi;

    lo = 0;  // index vals
    hi = n-1;
    while (lo <= hi)
    {
       mid = (lo+hi)/2;

       if (a[mid] == x) // found
            return true;

       if(a[mid] < x) // search in lower half
             lo = mid+1;
       else                  // search in upper half
            hi = mid-1;
    }
    return false;
}
```