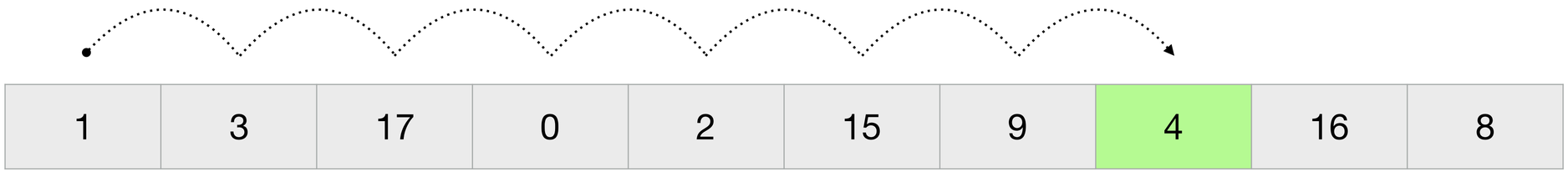
Essential Computing 1

Search

Sequential search for unordered arrays

- Works on unordered lists.
- N elements takes N lookups.

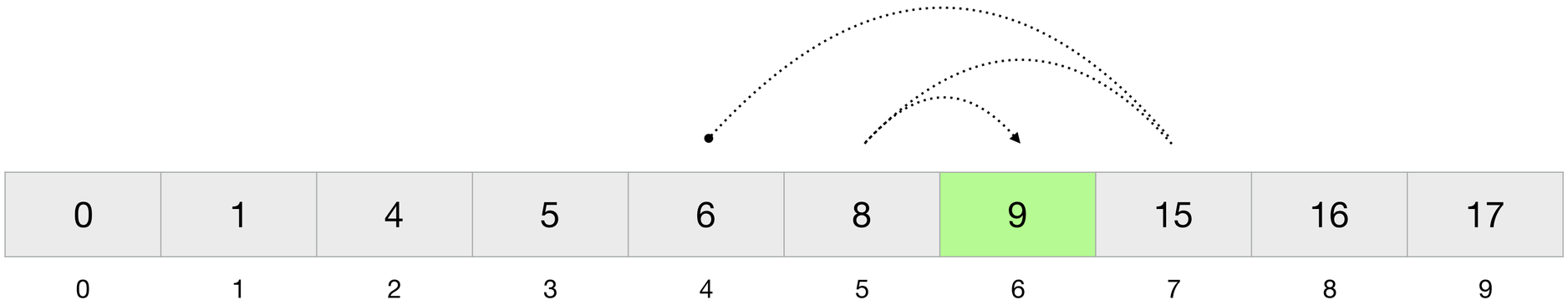


Sequential search for the index of a value in an array of values

```
public static int sequentialIndexOf( int[] values, int value )
{
    for( int i=0; i<values.length; i++ ){
        if( values[i] == value ) return i;
    }
    return -1;
}
```

Binary search: for ordered arrays

- Each step the number of elements to check is halved.
- N elements takes $\log_2 N$ lookups. 100 => 6.64



Binary search for the index of a value in an array of ordered values

```
public static int binaryIndexOf( int[] values, int value )
{
    int iLow = 0;
    int iHigh = values.length-1;
    while( iLow <= iHigh ){
        int iGuess = ( iLow + iHigh ) / 2;
        int testValue = values[iGuess];
        if( testValue == value ) return iGuess;
        if( value > testValue ) iLow = iGuess + 1;
        else iHigh = iGuess - 1;
    }
    return -1;
}
```

Binary search for humans

1. Pick a *target* value to search for in an array of values.
2. Set *low* index and *high* index to the index bounds of the array.
3. Set *guess* index halfway between *low* and *high* and read *value*.
4. If *value* equals *target*, return the *guess* index.
5. If *value* is lower than *target*, set *low* index to *guess* + 1.
6. If *value* is higher than *target*, set *high* index to *guess* - 1.
7. repeat 2.