

Binary search

pre: b 0 ? b.length post: b 0 h b.length
<= v ? > v | <= v ? > v

```

h= -1; t= b.length;
while (h != t-1) {
  int e= (h+t)/2;
  // h < e < t
  if (b[e] <= v) h= e;
  else t= e;
}

```

```

h= -1; t= b.length - 1;
while (h != t) {
}

```

To make sure that progress is made, b[e] has to be in ? section

Binary search

pre: b 0 ? b.length post: b 0 h b.length
Case that ? has 1 element, t = h+1
<= v ? > v | <= v ? > v

```

(h+t)/2
= <t = h+1>
(2h + 1) / 2
= <arith: / is int division>
h

```

```

h= -1; t= b.length - 1;
while (h != t) {
  int e= (h+t)/2;
}

```

In this case, e= (h+t)/2; sets e to h! b[e] not in ? section

Binary search

pre: b 0 ? b.length post: b 0 h b.length
<= v ? > v | <= v ? > v

```

h= -1; t= b.length;
while (h != t-1) {
  int e= (h+t)/2;
  // h < e < t
  if (b[e] <= v) h= e;
  else t= e;
}

```

```

h= -1; t= b.length - 1;
while (h != t) {
  int e= (h+t+1)/2;
  // h < e <= t
  if (b[e] <= v) h= e;
  else t= e-1;
}

```

If b[e] > v, change t to point before b[e]
