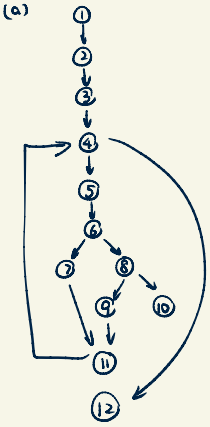
1.

```
int binsearch(int X, int V[], int n){
 1 int low, high, mid;
 2 low = 0;
 3 high = n - 1;
 4 while (low <= high) {
 5    mid = (low + high)/2;
 6    if (X < V[mid])
 7          high = mid - 1;
 8    else if (X > V[mid])
 9          low = mid + 1;
      else
10          return mid;
11 }
12 return -1;
}
```

(a)



(b) Linearly independent paths

- 1, 2, 3, 4, 12
- 1, 2, 3, 4, 5, 6, 8, 10
- 1, 2, 3, 4, 5, 6, 7, 11, 4, 12
- 1, 2, 3, 4, 5, 6, 7, 11, 4, 5, 6, 8, 10
- 1, 2, 3, 4, 5, 6, 8, 9, 11, 4, 12
- 1, 2, 3, 4, 5, 6, 8, 9, 11, 4, 5, 6, 8, 10

(c) Complete test cases

- 1, 2, 3, 4, 12
  input : (0, [0], 0) output : -1
- 1, 2, 3, 4, 5, 6, 8, 10
  input : (1, [0, 1, 2], 3) output : 1
- 1, 2, 3, 4, 5, 6, 7, 11, 4, 12
  input : (0, [1, 2, 3], 3) output : -1
- 1, 2, 3, 4, 5, 6, 7, 11, 4, 5, 6, 8, 10
  input : (0, [0, 1, 2], 3) output : 0
- 1, 2, 3, 4, 5, 6, 8, 9, 11, 4, 12
  input : (3, [0, 1, 2], 3) output : -1
- 1, 2, 3, 4, 5, 6, 8, 9, 11, 4, 5, 6, 8, 10
  input : (2, [0, 1, 2], 3) output : 2

2.

	a	b	c
A	B/e	B/f	C/e
B	A/f	C/f	B/f
C	C/f	A/e	B/e

	1	2	3	4	5	6	7	8	9
Start State	A	A	A	B	B	B	C	C	C
Input	a	b	c	a	c	b	b	c	a
Expected Output	e	f	e	f	f	f	e	e	f
Finish State	B	B	C	A	B	C	A	B	C
Test Coverage Item	1	2	3	4	6	5	8	9	7