

```
>*                K

*                F                K                P

*      B      F      I      K      M      P      R

*  A  B  C  F  G  I  J  K  L  M  O  P  Q  R  S
   4  3  1  8  6  7  0  2  9  0  8  4  6  3  5
```

Find Q.

start is null.

```
*                               >K

*           F           K           P

*       B       F       I       K       M       P       R

*  A  B  C  F  G  I  J  K  L  M  O  P  Q  R  S
   4  3  1  8  6  7  0  2  9  0  8  4  6  3  5
```

Find Q.

Move forward as much as possible without passing Q.

```
*           K
*           F           >K           P
*           B           F           I           K           M           P           R
*  A  B  C  F  G  I  J  K  L  M  O  P  Q  R  S
   4  3  1  8  6  7  0  2  9  0  8  4  6  3  5
```

Find Q.

Use *value* of Node as start in next lower list.

```
*           K
*           F           K           >P
*           B           F           I           K           M           P           R
*  A  B  C  F  G  I  J  K  L  M  O  P  Q  R  S
   4  3  1  8  6  7  0  2  9  0  8  4  6  3  5
```

Find Q.

Move forward as much as possible without passing Q.

```
*           K
*           F           K           P
*           B           F           I           K           M           >P           R
*  A  B  C  F  G  I  J  K  L  M  O  P  Q  R  S
   4  3  1  8  6  7  0  2  9  0  8  4  6  3  5
```

Find Q.

Use *value* of Node as start in next lower list.

```
*           K
*           F           K           P
*           B           F           I           K           M           >P           R
*  A  B  C  F  G  I  J  K  L  M  O  P  Q  R  S
   4  3  1  8  6  7  0  2  9  0  8  4  6  3  5
```

Find Q.

Move forward as much as possible without passing Q.

```
*           K
*           F           K           P
*           B           F           I           K           M           P           R
*  A  B  C  F  G  I  J  K  L  M  O >P  Q  R  S
   4  3  1  8  6  7  0  2  9  0  8  4  6  3  5
```

Find Q.

Use *value* of Node as start in next lower list.

```
*           K
*           F           K           P
*           B           F           I           K           M           P           R
*  A  B  C  F  G  I  J  K  L  M  O  P >Q  R  S
   4  3  1  8  6  7  0  2  9  0  8  4  6  3  5
```

Find Q.

Move forward as much as possible without passing Q.


```
*           K
*           F           K           P
*           B           F           I           K           M           P           R
*  A  B  C  F  G  I  J  K  L  M  O  P >Q  R  S
   4  3  1  8  6  7  0  2  9  0  8  4  6  3  5
```

Find Q.

Node has key==Q, return it.