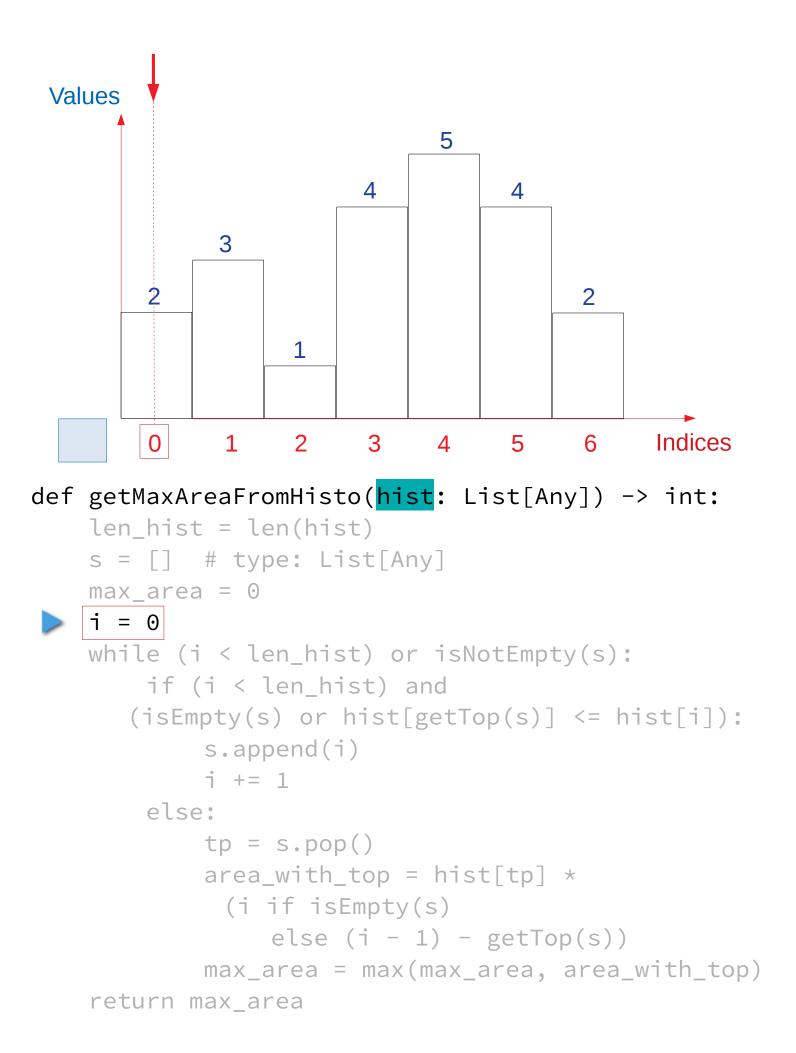
hist

```
Values
```

```
3
2
1
2
0 1 2 3 4 5 6 Indices
```

```
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    i = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
            s.append(i)
            i += 1
        else:
            tp = s.pop()
            area_with_top = hist[tp] *
              (i if isEmpty(s)
                 else (i - 1) - getTop(s))
            max_area = max(max_area, area_with_top)
    return max_area
```

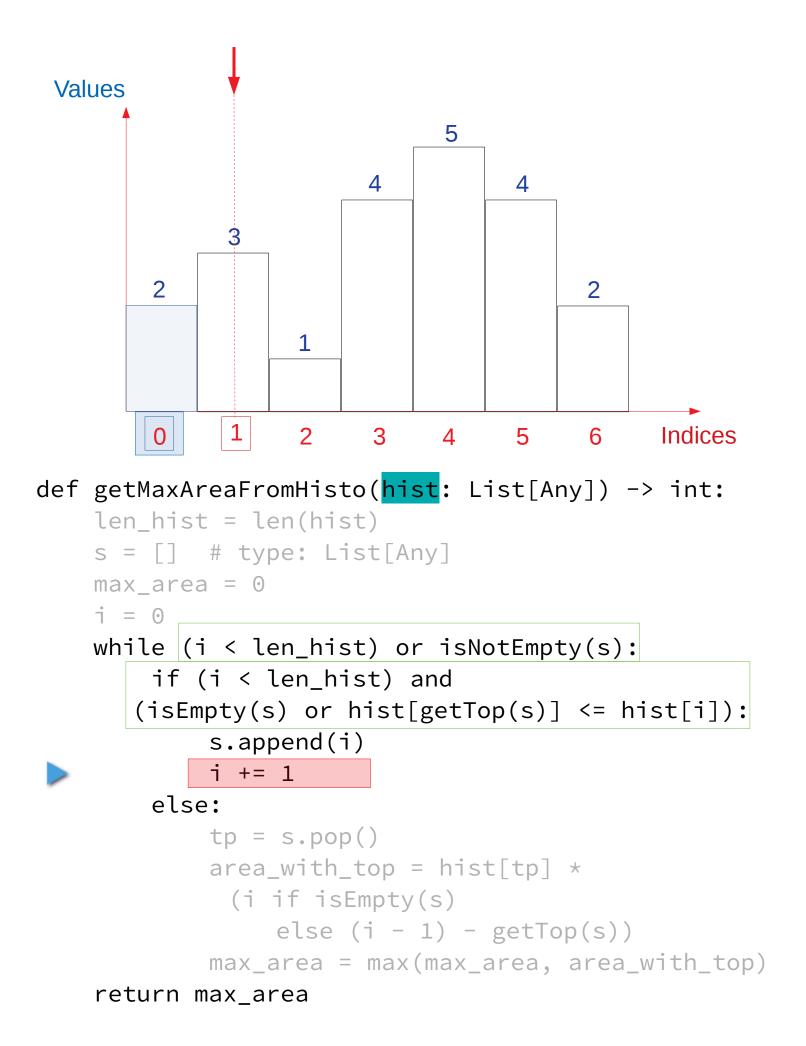
```
hist
 Values
                        len_hist
                               5
                         4
                                    4
              3
        2
                                          2
                    1
                                               Indices
                   2
                         3
              1
                              4
                                    5
                                          6
        0
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    i = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and</pre>
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s))
             max_area = max(max_area, area_with_top)
    return max_area
```



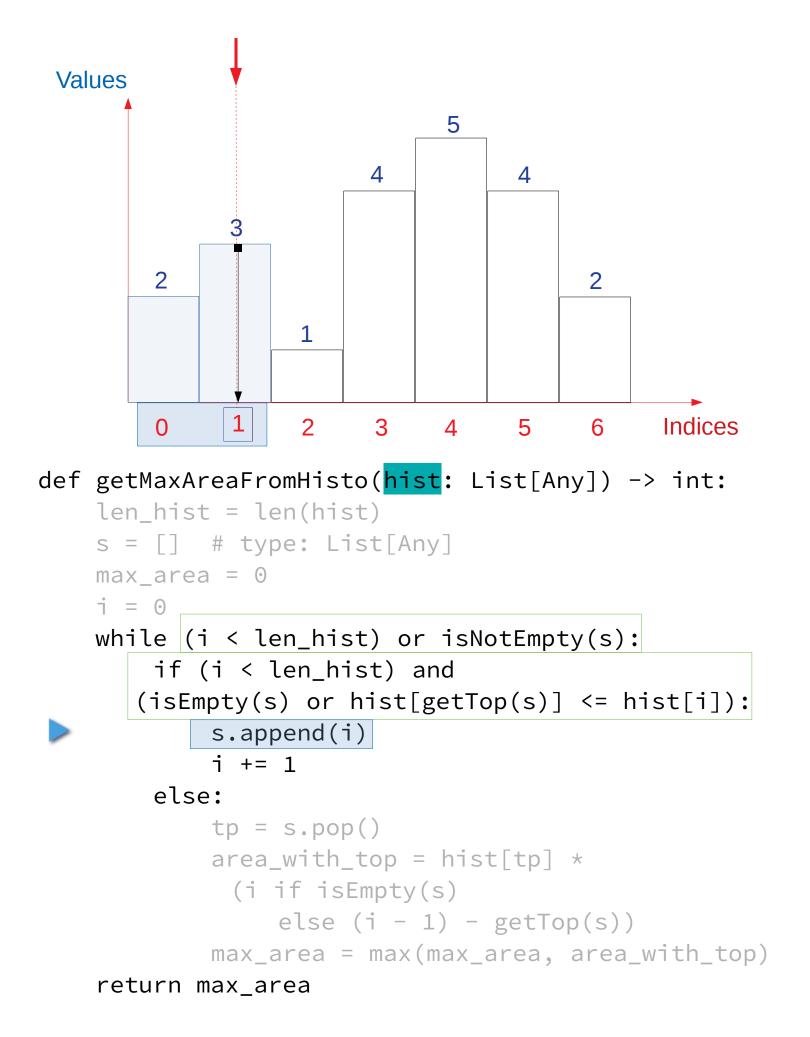
```
Values
                              5
                         4
                                    4
                                         2
                   1
                                               Indices
                   2
                         3
              1
                              4
                                    5
                                         6
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    j = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
             s.append(i)
            i += 1
        else:
            tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s))
            max_area = max(max_area, area_with_top)
    return max_area
```

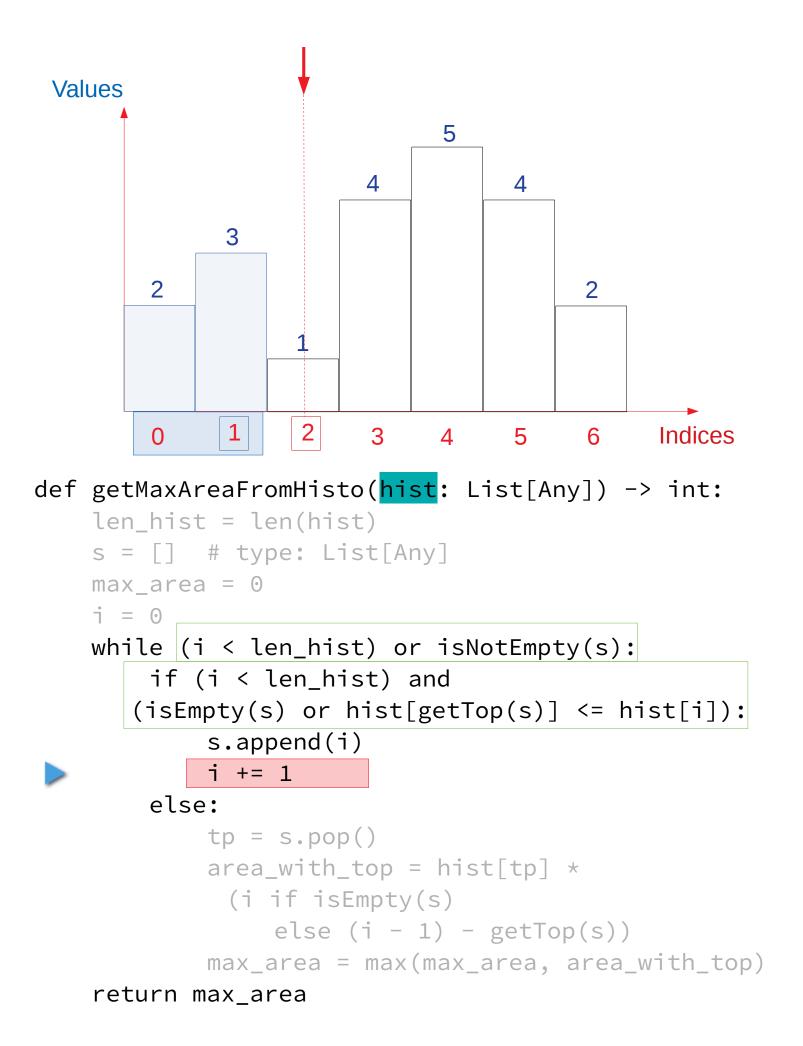
```
Values
                               5
                         4
                                     4
                                          2
                    1
                                                Indices
                    2
                          3
              1
                               4
                                     5
                                          6
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    j = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
         if (i < len_hist) and</pre>
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):</pre>
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s))
             max_area = max(max_area, area_with_top)
    return max_area
```

```
Values
                              5
                         4
                                    4
                                         2
                   1
                                               Indices
                         3
        0
              1
                   2
                              4
                                    5
                                         6
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    j = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                 else (i - 1) - getTop(s))
             max_area = max(max_area, area_with_top)
    return max_area
```



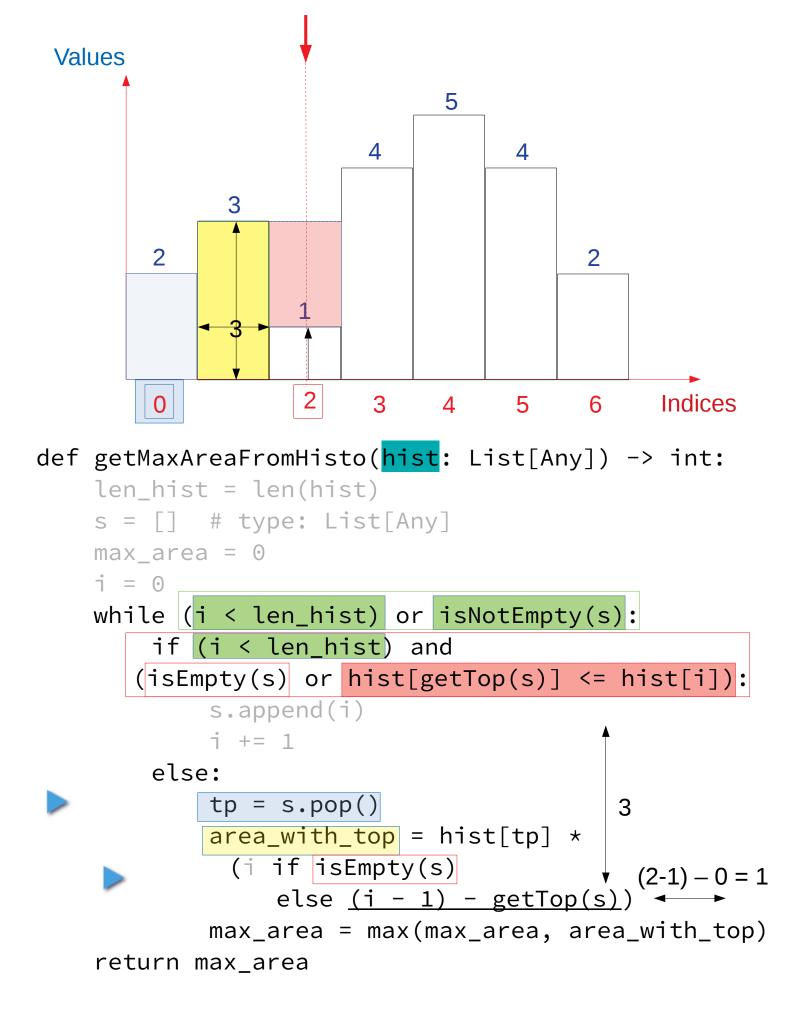
```
Values
                               5
                         4
                                    4
        2
                                          2
                    1
              1
                                                Indices
                    2
                         3
        0
                               4
                                    5
                                          6
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    j = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and</pre>
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s))
             max_area = max(max_area, area_with_top)
    return max_area
```

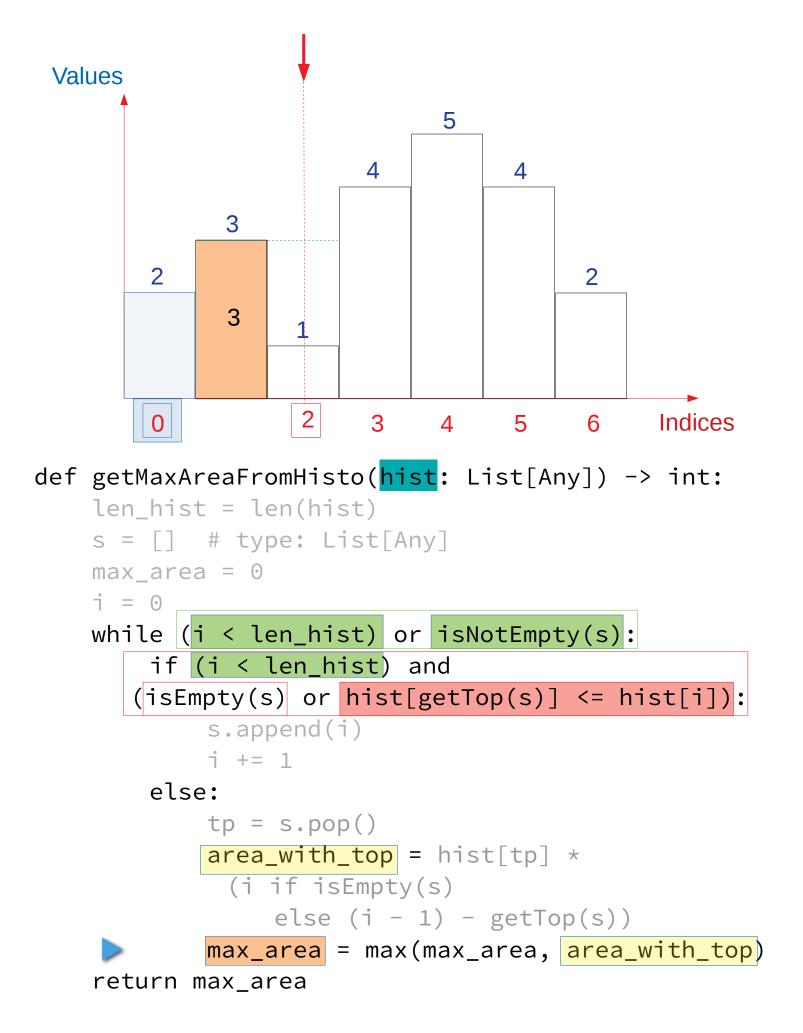




```
Values
                              5
                         4
                                    4
        2
                                         2
                                               Indices
                         3
                              4
                                    5
                                         6
        0
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    i = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
             s.append(i)
             i += 1
        else:
            tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s))
            max_area = max(max_area, area_with_top)
    return max_area
```

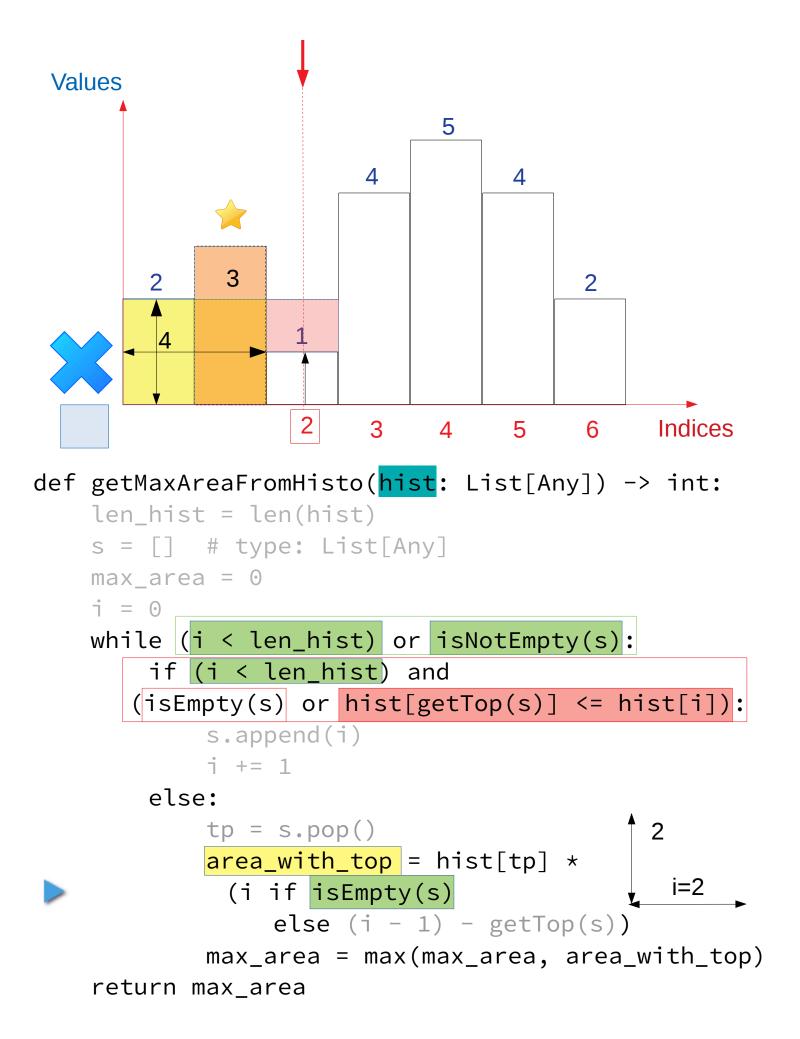
```
Values
                              5
                         4
                                    4
        2
                                         2
                                               Indices
                         3
        0
                              4
                                    5
                                         6
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    j = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
             s.append(i)
            i += 1
        else:
            tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s))
            max_area = max(max_area, area_with_top)
    return max_area
```

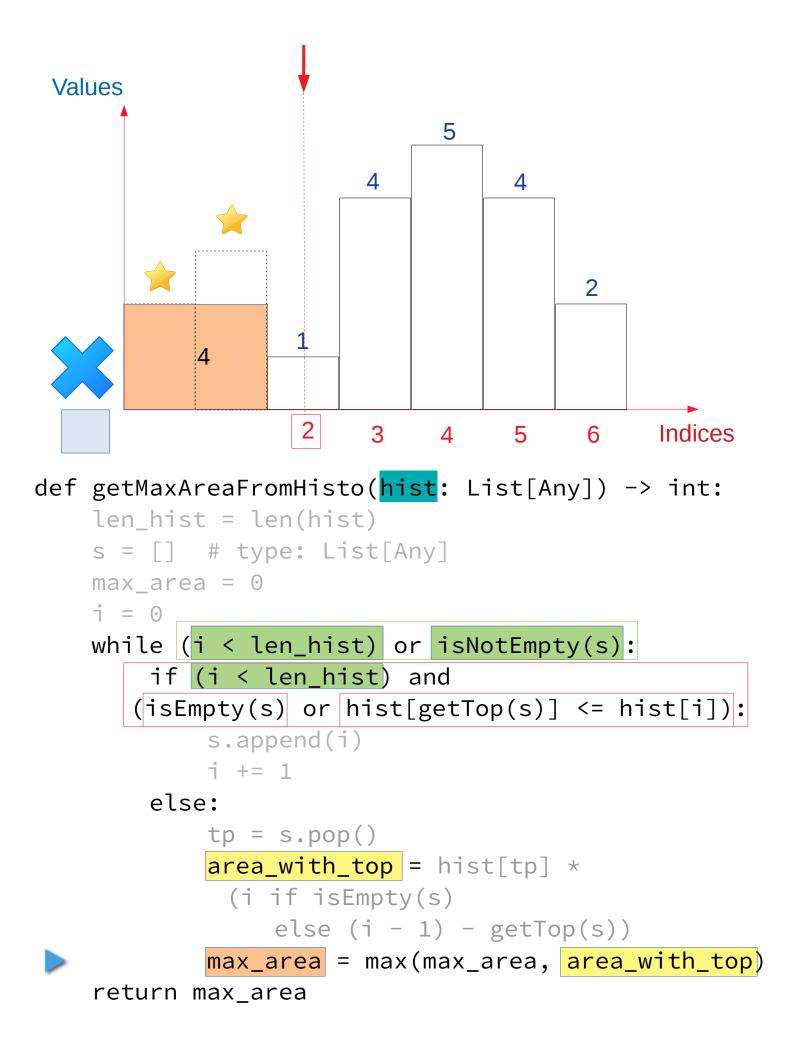




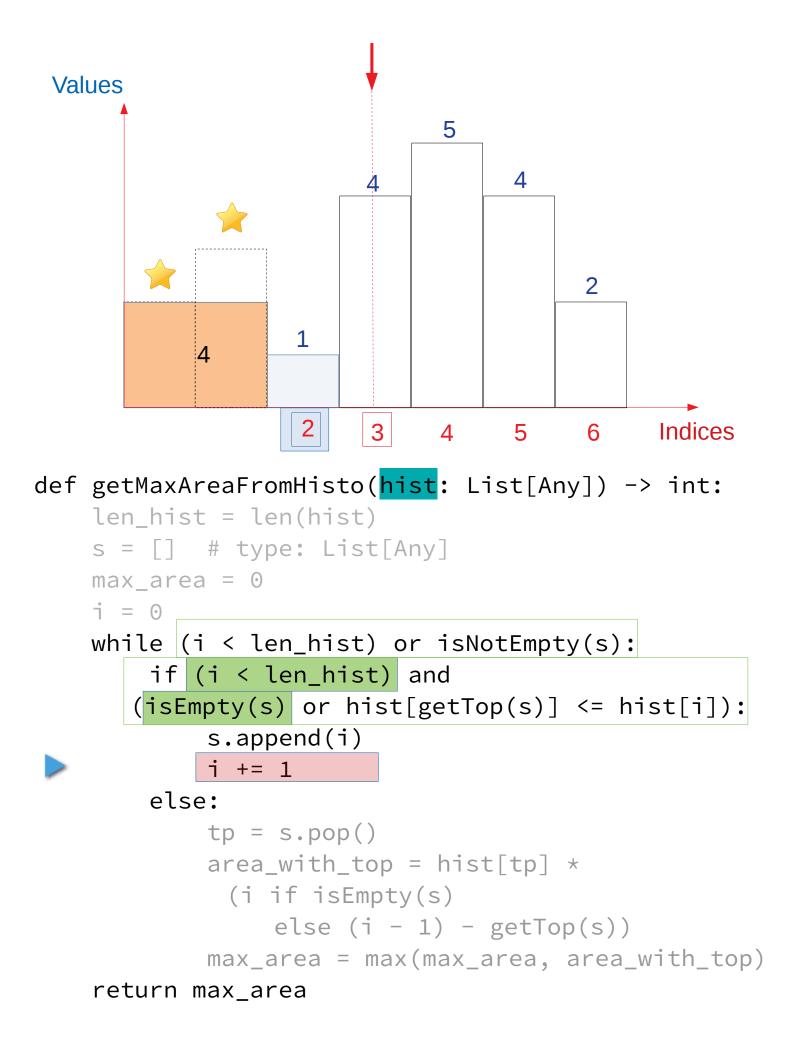
```
Values
                               5
                         4
                                    4
              3
                                         2
                                               Indices
                         3
        0
                              4
                                    5
                                          6
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    i = ⊙
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):</pre>
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s)
             max_area = max(max_area, area_with_top)
    return max_area
```

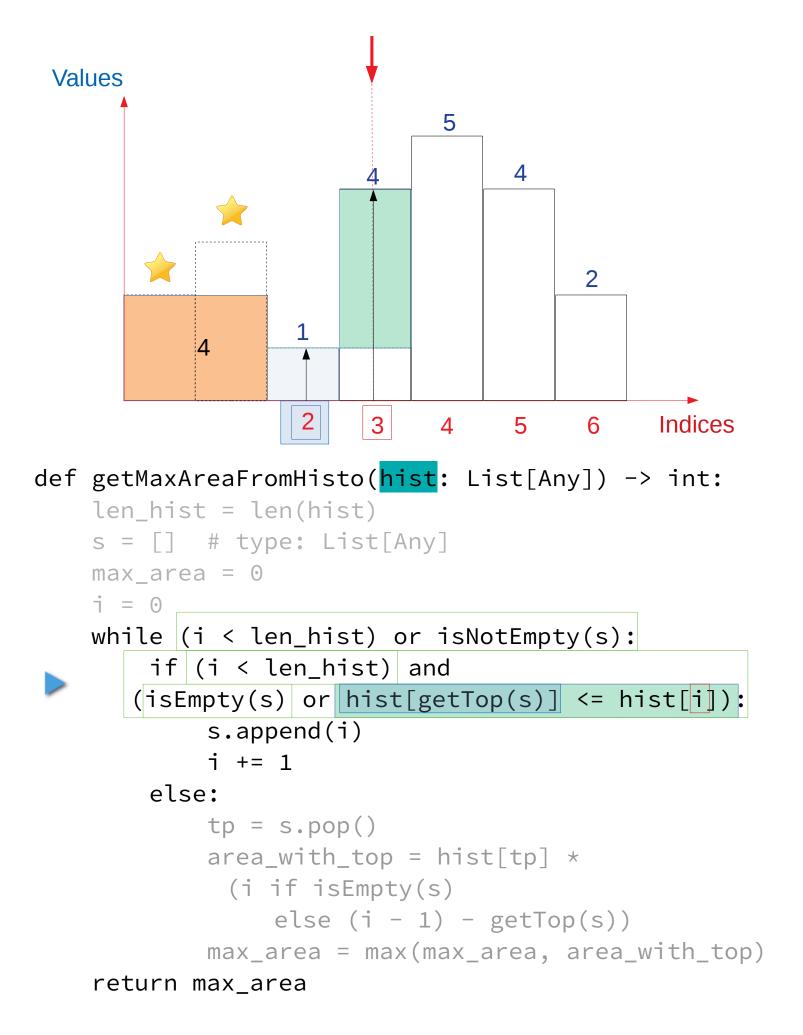
```
Values
                              5
                         4
                                    4
              3
                                         2
                                               Indices
                         3
        0
                              4
                                    5
                                         6
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    i = ⊙
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (|isEmpty(s)| or |hist[getTop(s)] <= hist[i]):
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s)
             max_area = max(max_area, area_with_top)
    return max_area
```

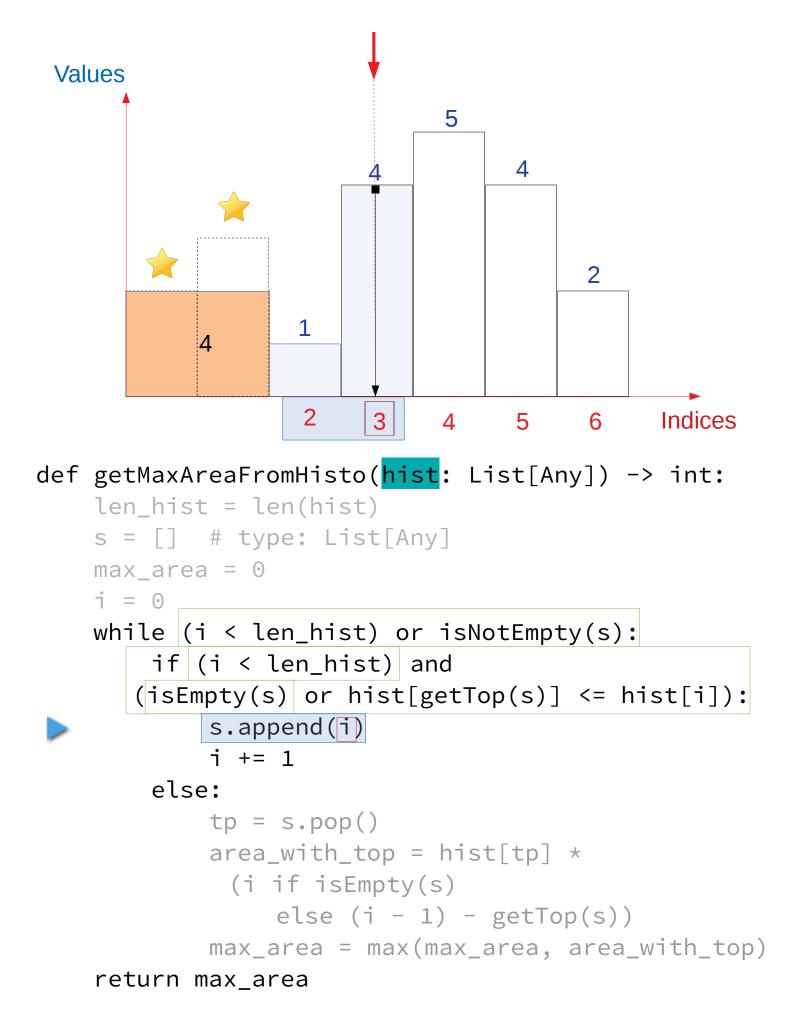


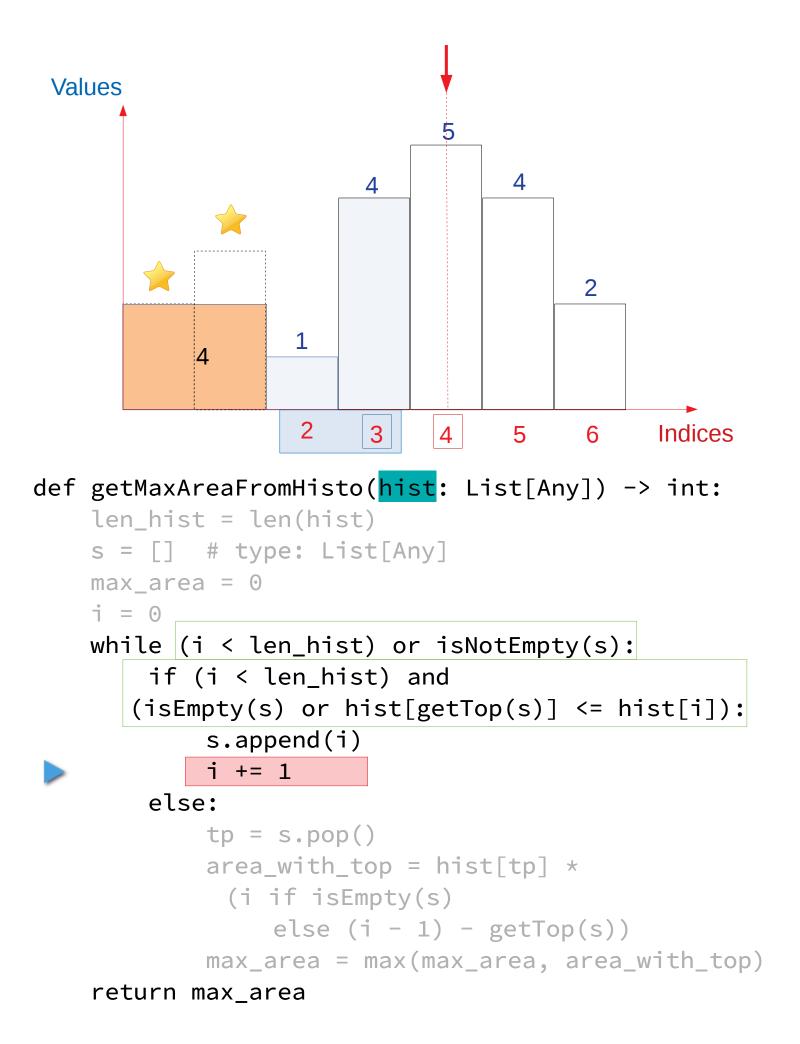


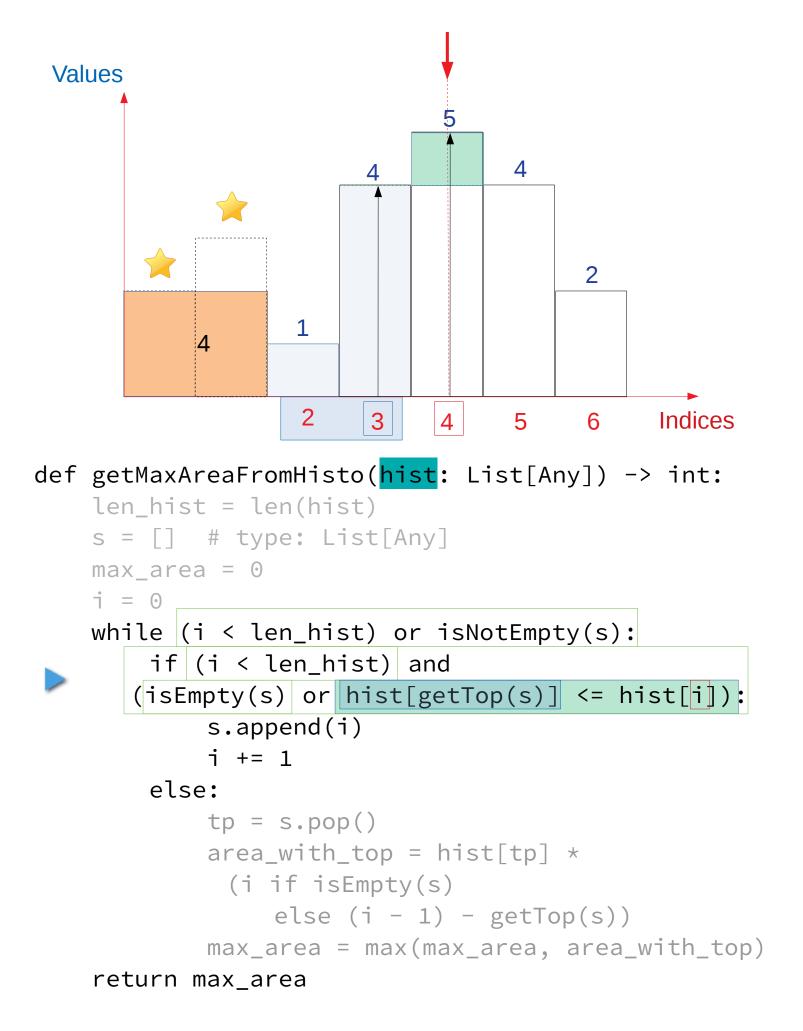
```
Values
                               5
                         4
                                     4
                                          2
                                                Indices
                         3
                                    5
                                          6
                               4
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    j = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and</pre>
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):</pre>
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s))
             max_area = max(max_area, area_with_top)
    return max_area
```

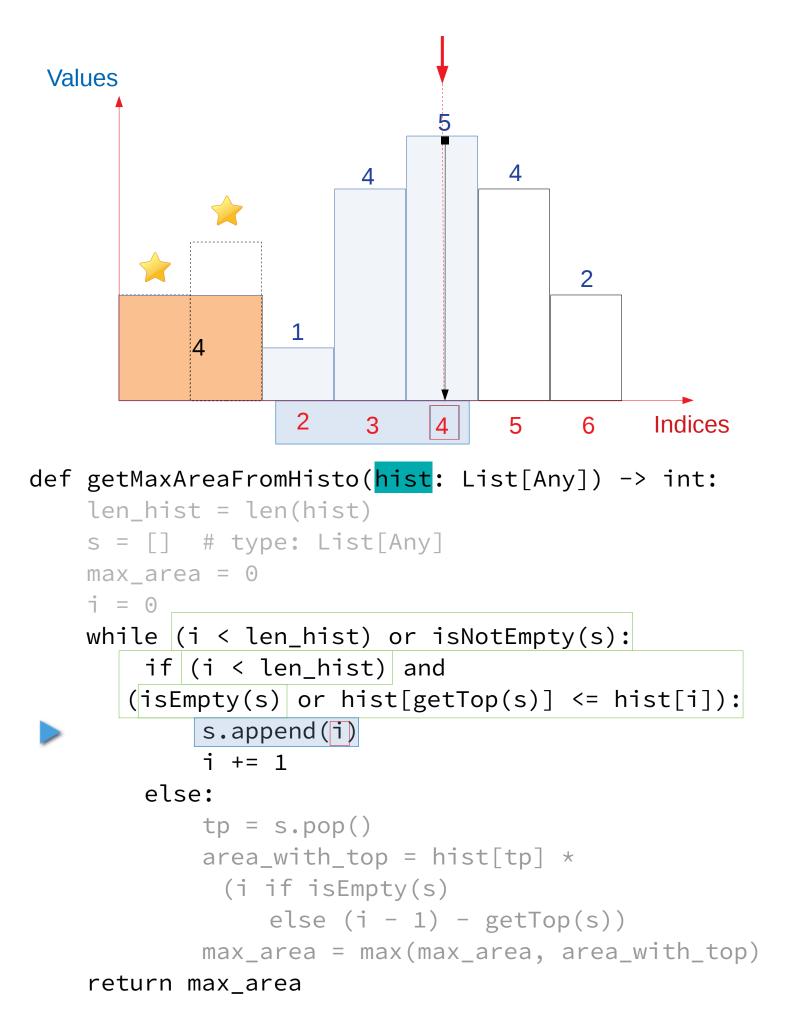


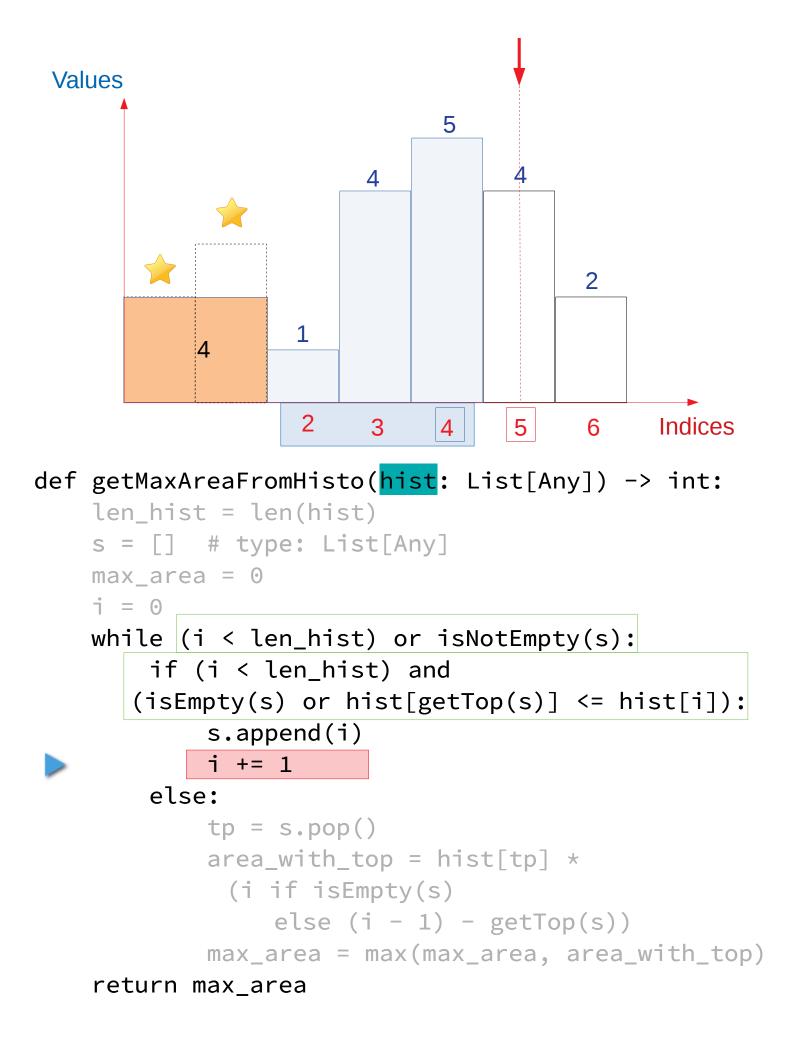






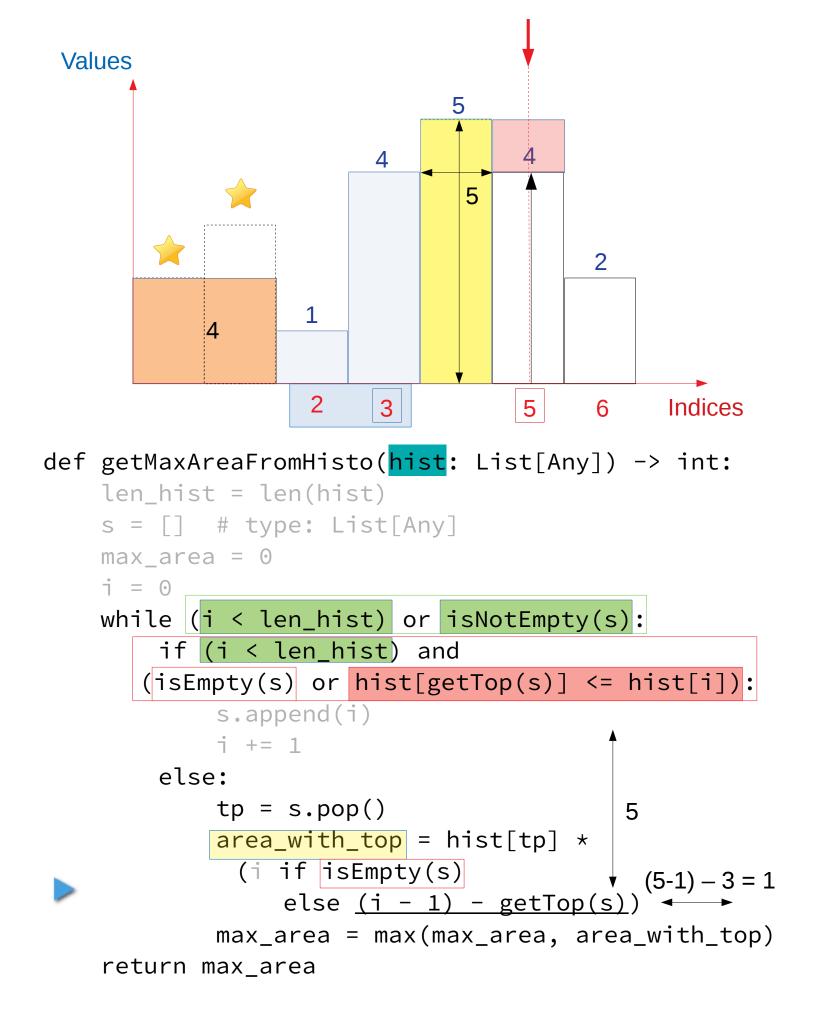


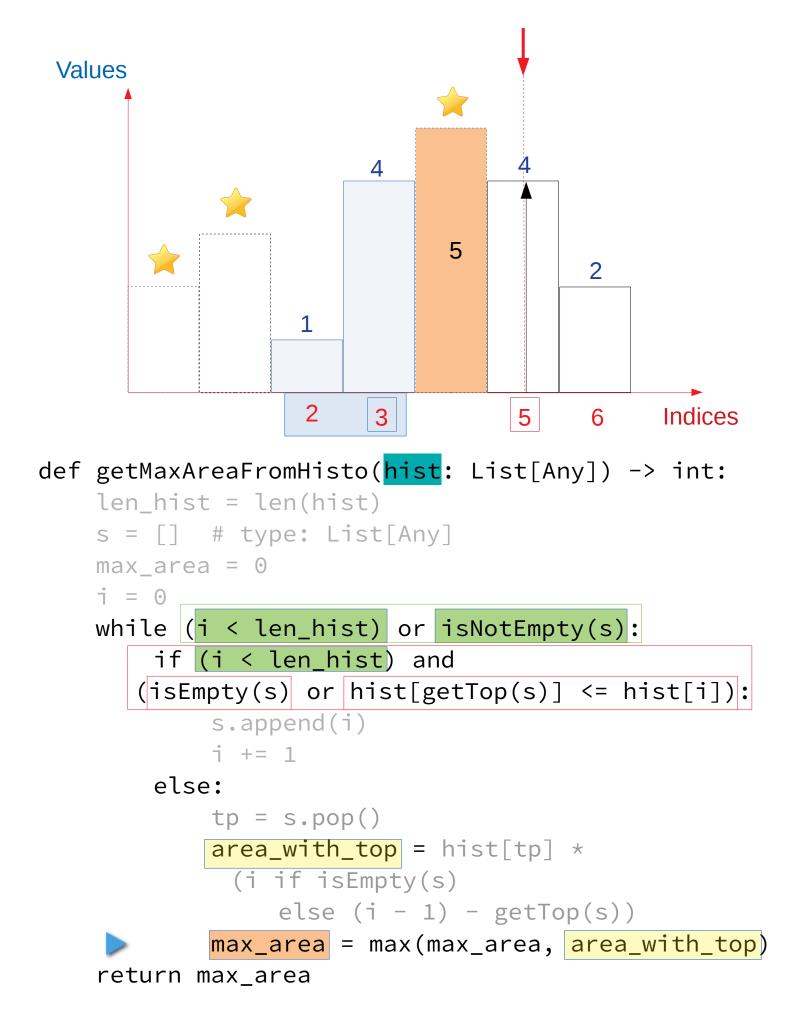


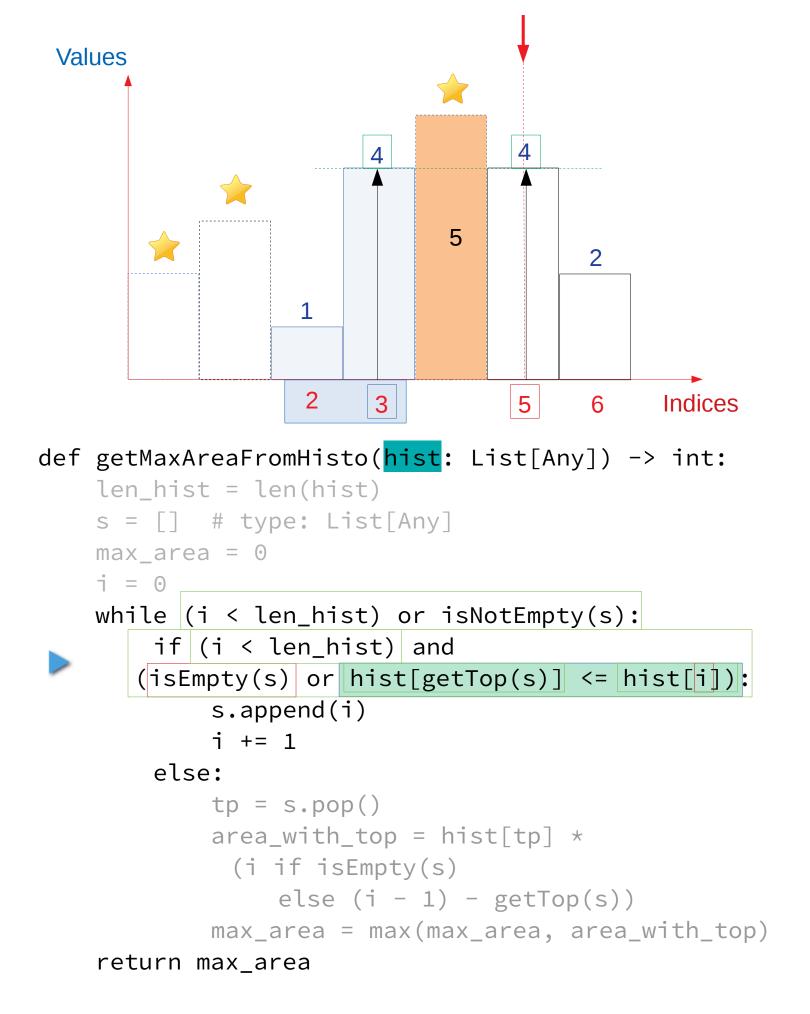


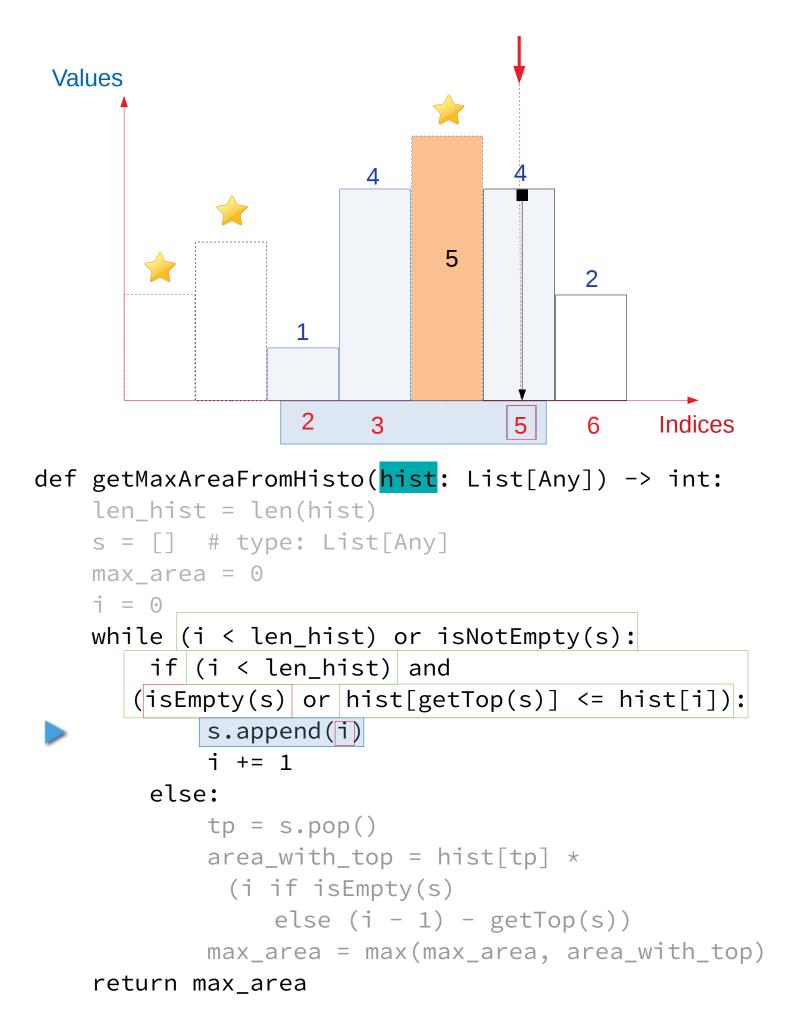
```
Values
                              5
                         4
                                         2
                   1
                                   5
                                              Indices
                         3
                                         6
                              4
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    i = ⊙
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
             s.append(i)
            i += 1
        else:
            tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                 else (i - 1) - getTop(s))
            max_area = max(max_area, area_with_top)
    return max_area
```

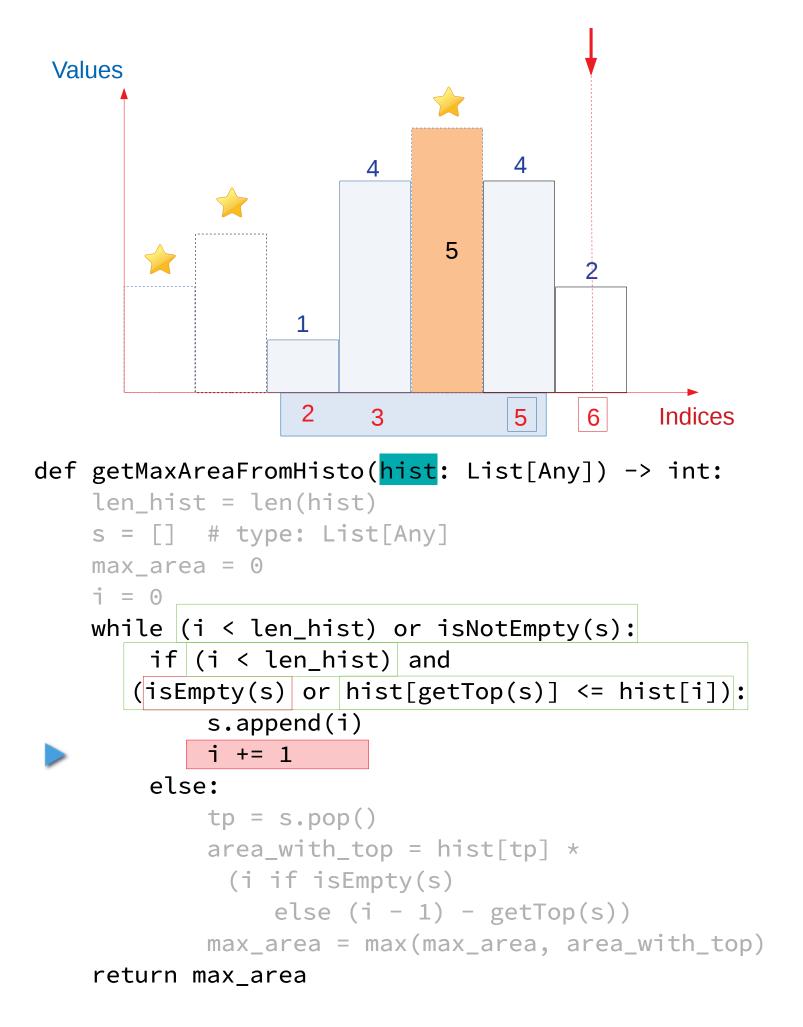
```
Values
                              5
                        4
                                         2
                   1
                                              Indices
                                   5
                         3
                                         6
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    i = ⊙
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
             s.append(i)
            i += 1
        else:
            tp = s.pop()
            area_with_top = hist[tp] *
              (i if isEmpty(s)
                 else (i - 1) - getTop(s))
            max_area = max(max_area, area_with_top)
    return max_area
```

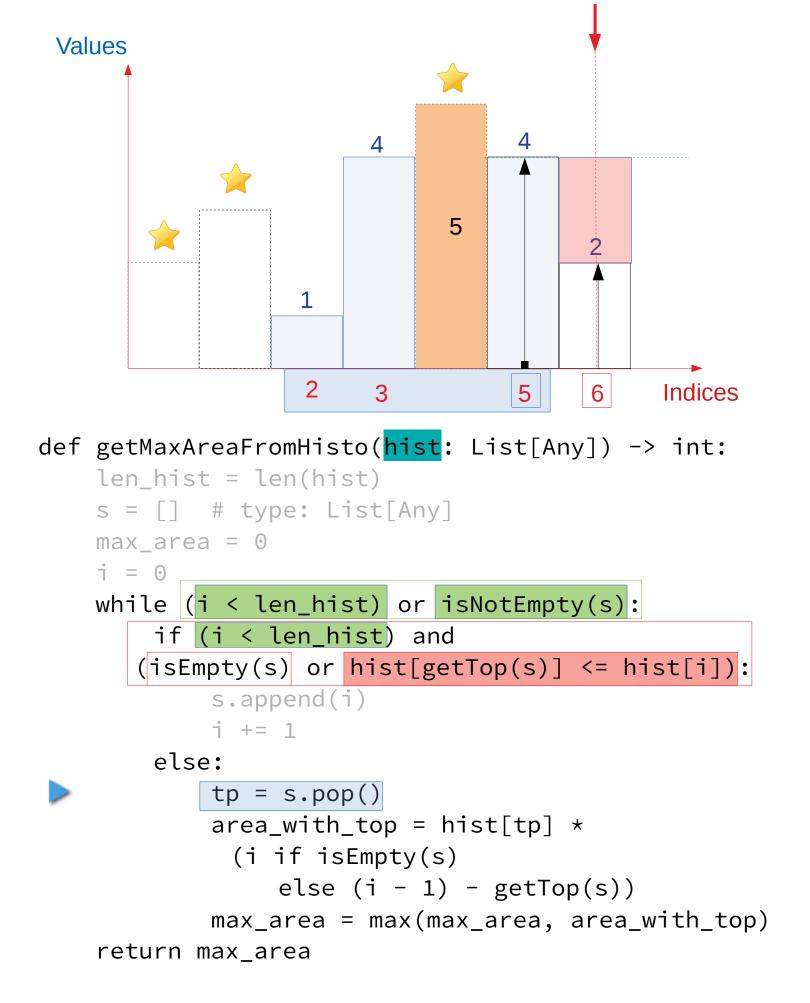


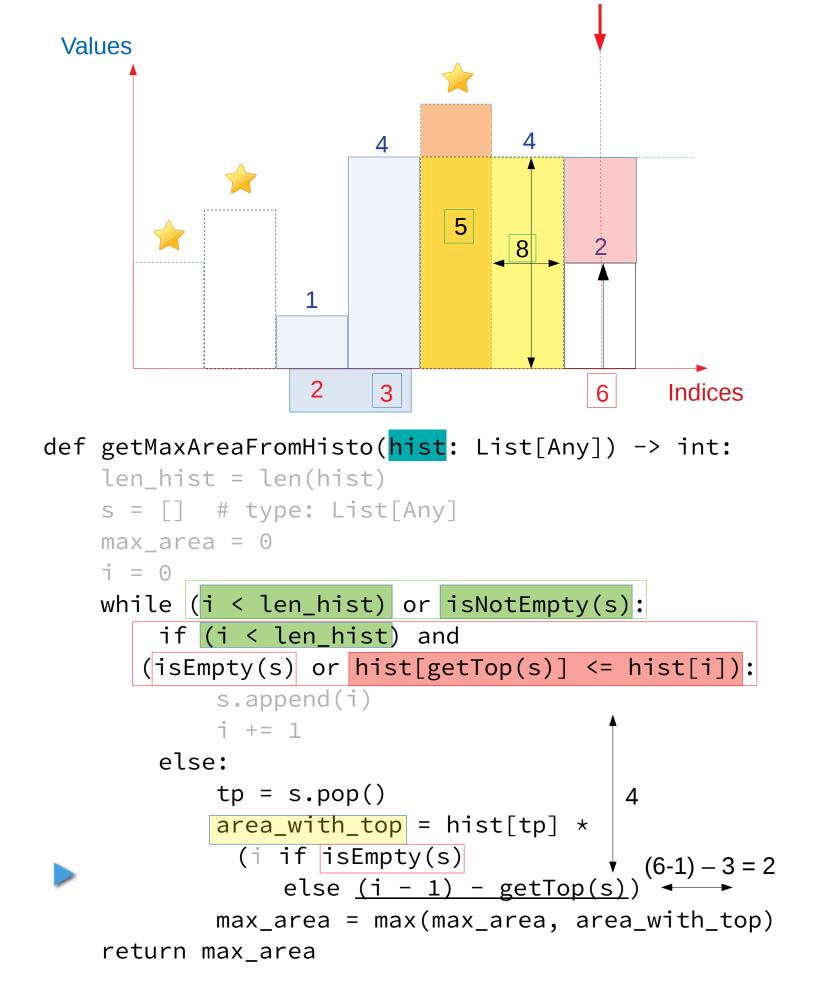


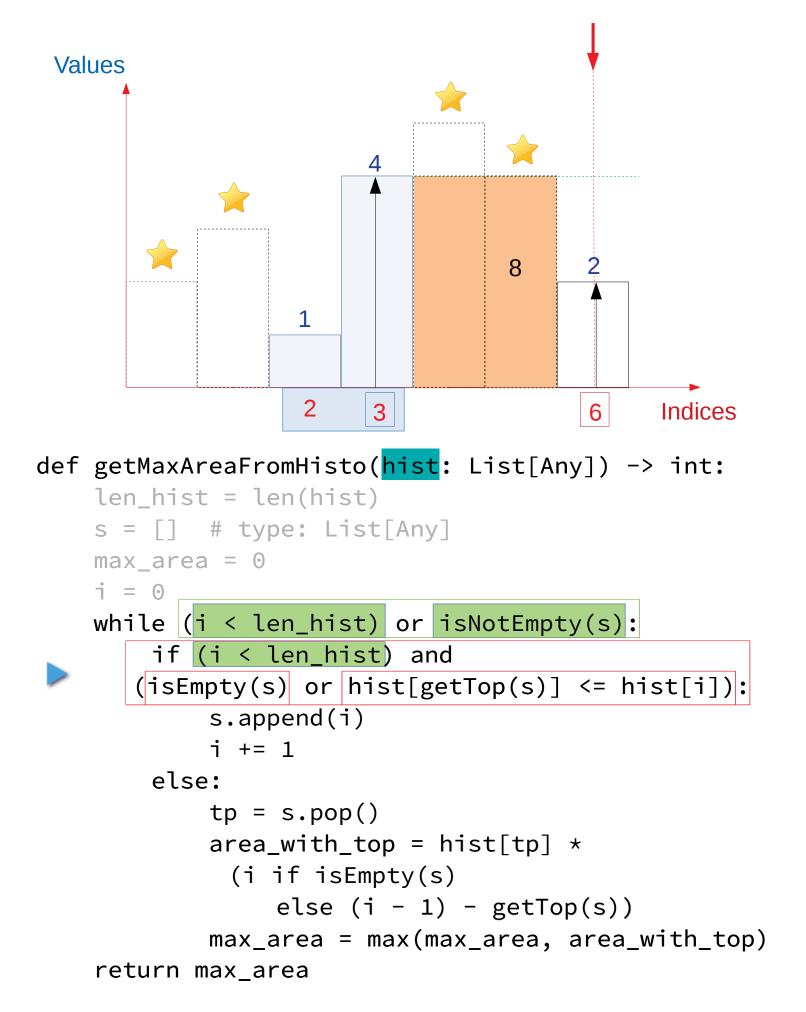


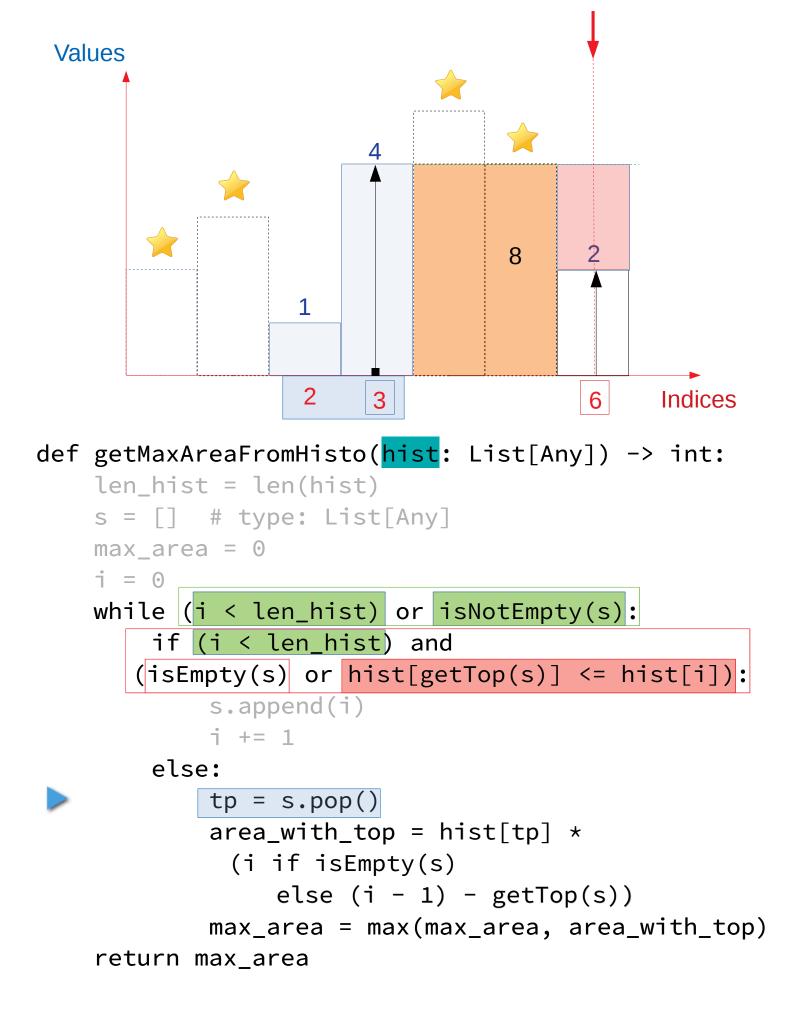


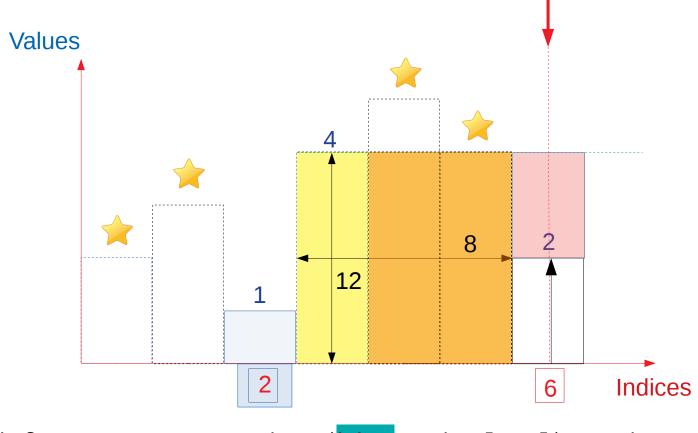




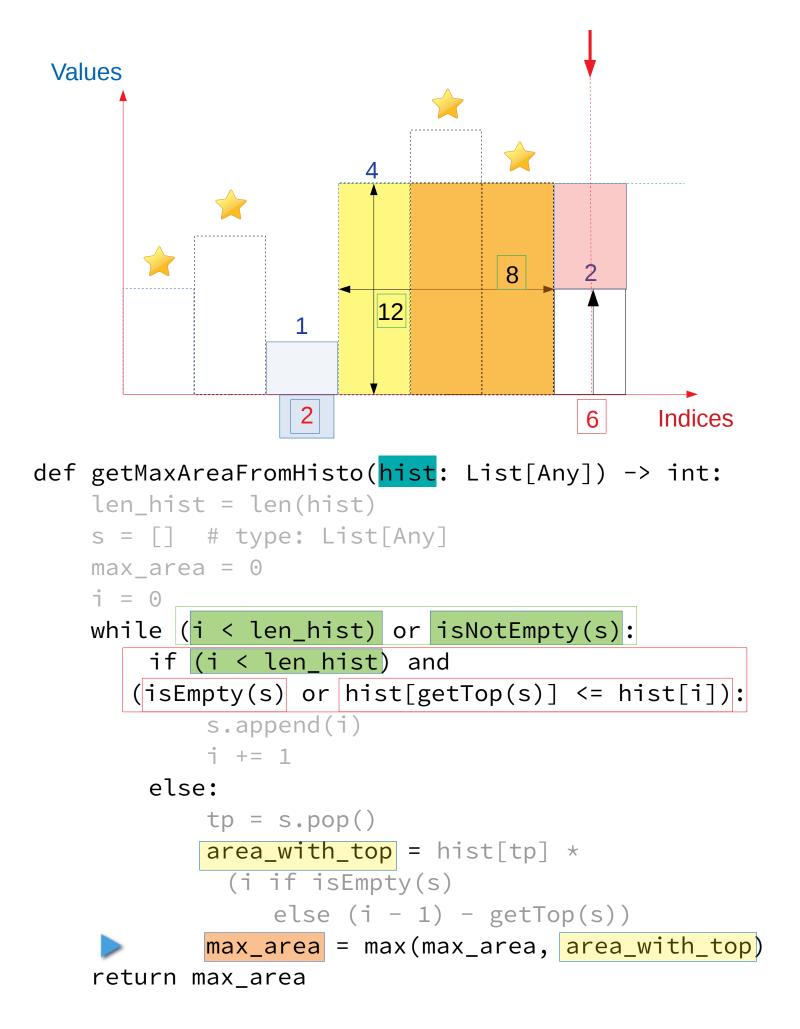


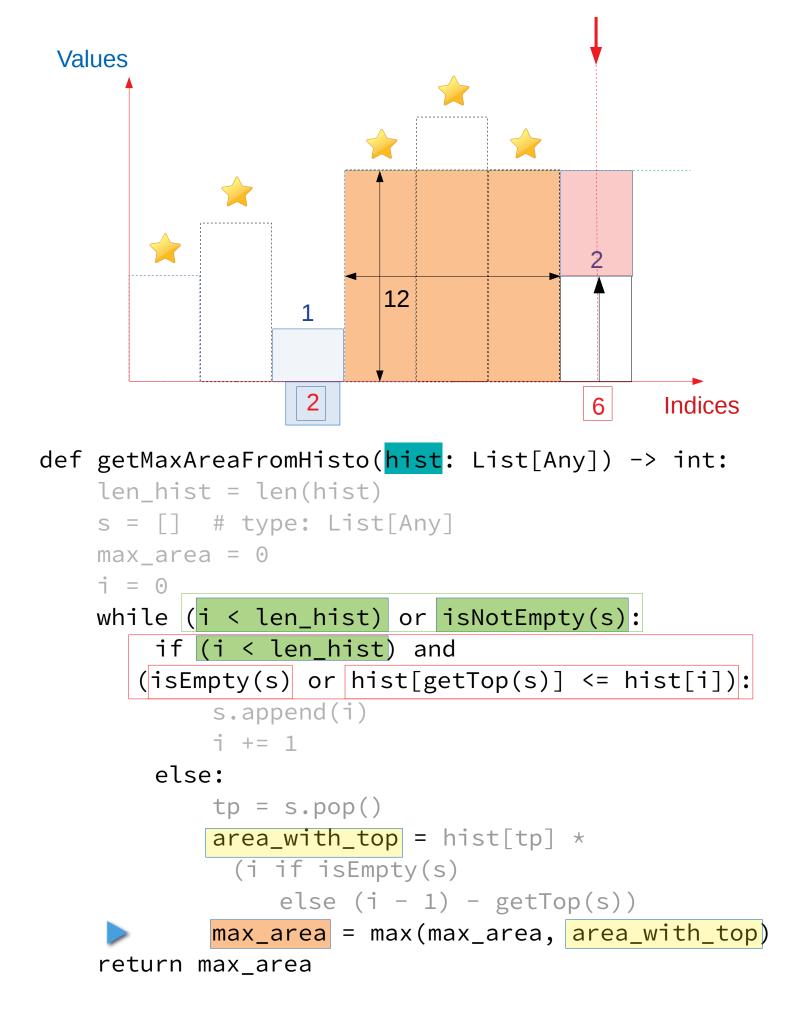


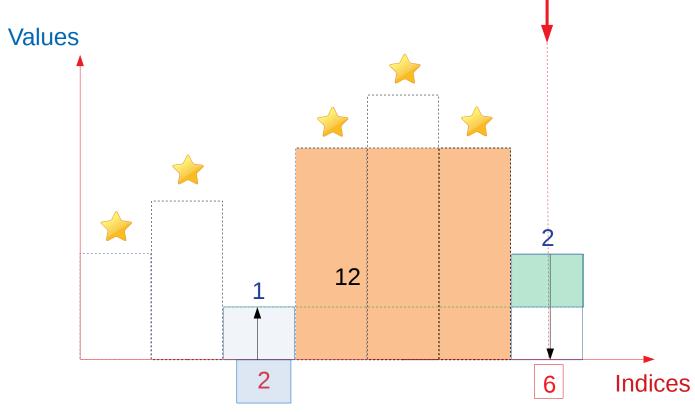




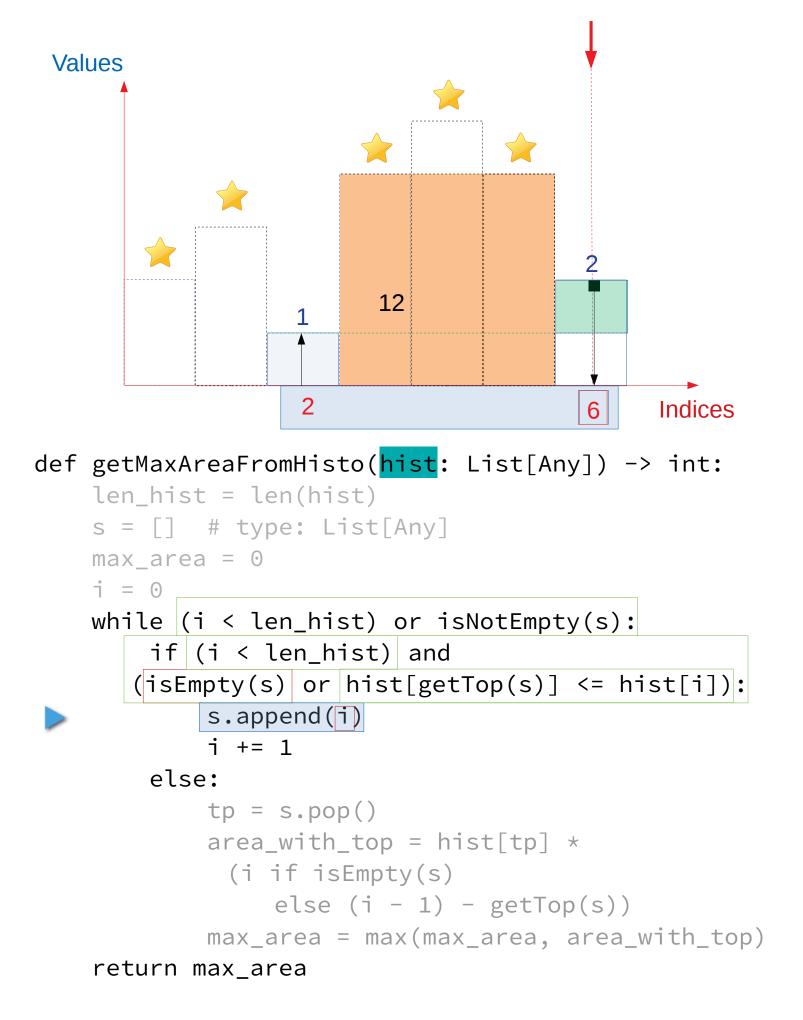
```
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    i = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (|isEmpty(s)| or |hist[getTop(s)] <= hist[i]):
            s.append(i)
            i += 1
        else:
            tp = s.pop()
                                          4
            area_with_top = hist[tp] *
              (i if isEmpty(s)
                 else (i - 1) - getTop(s)
            max_area = max(max_area, area_with_top)
    return max_area
```

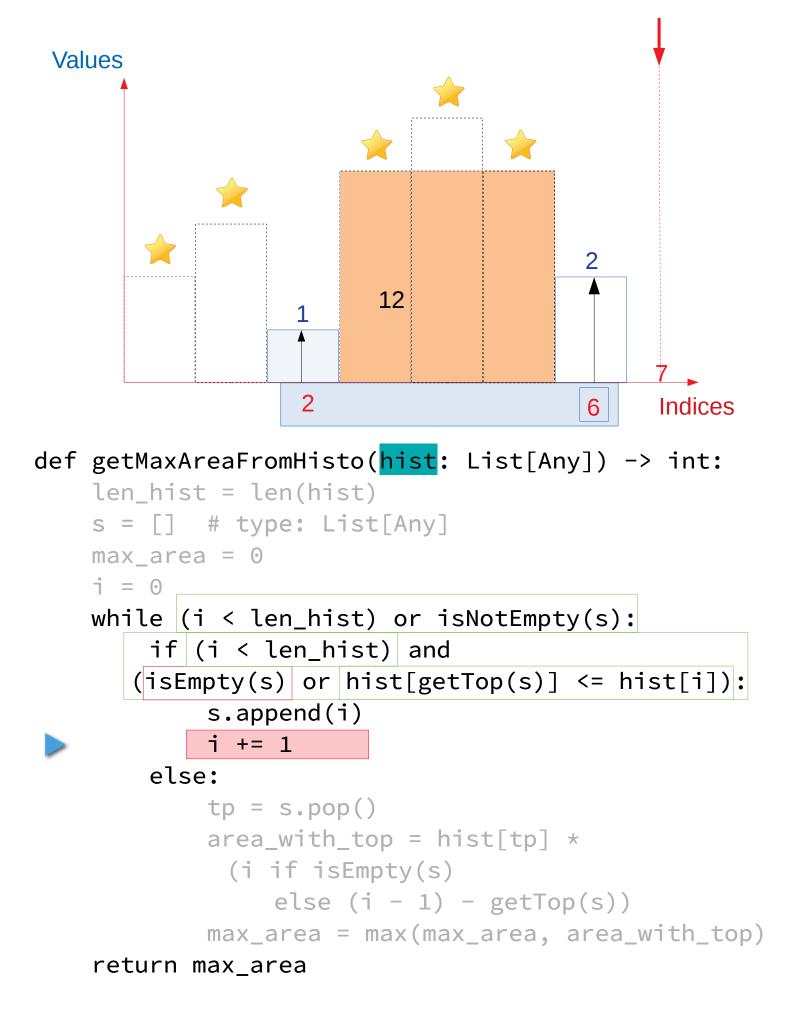


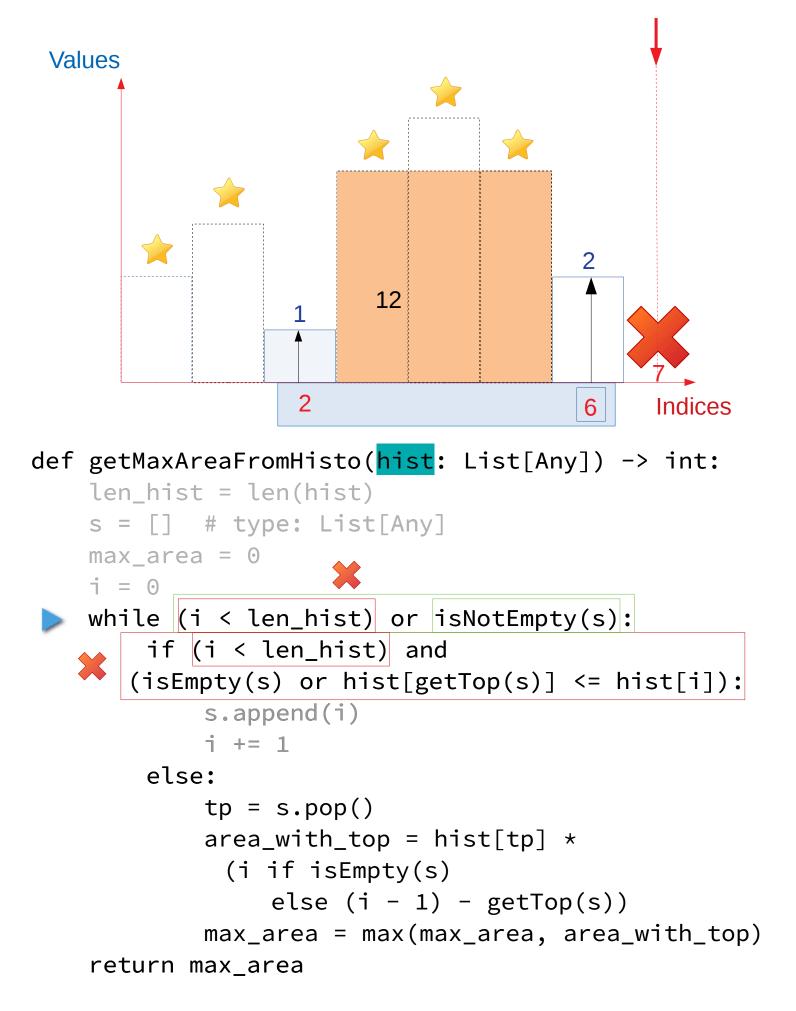


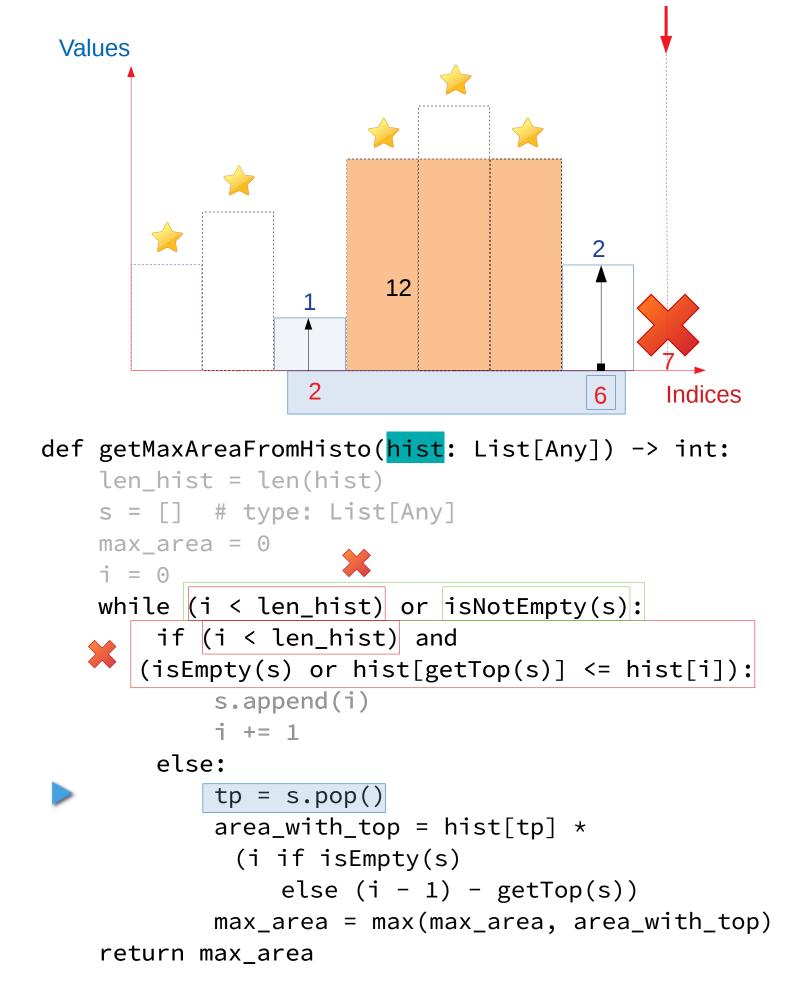


```
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max area = 0
    i = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
            s.append(i)
            i += 1
        else:
            tp = s.pop()
            area_with_top = hist[tp] *
              (i if isEmpty(s)
                 else (i - 1) - getTop(s))
            max_area = max(max_area, area_with_top)
    return max_area
```



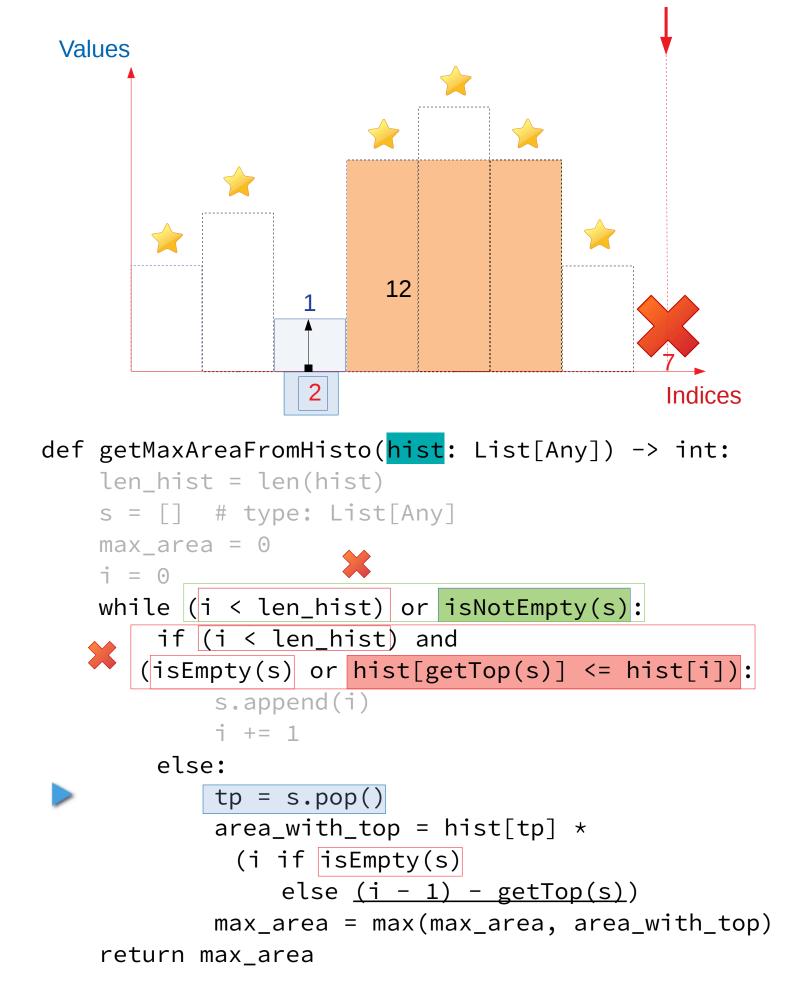






```
Values
                          12
                   1
                                               Indices
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max_area = 0
    j = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):</pre>
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s) (7-1) - 2 = 4
             max_area = max(max_area, area_with_top)
    return max_area
```

```
Values
                         12
                   1
                                        8
                                              Indices
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max_area = 0
    i = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):</pre>
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
              (i if isEmpty(s)
                  else (i - 1) - getTop(s)
            max_area = max(max_area, area_with_top)
    return max_area
```



```
Values
                          12
                    1
                                               Indices
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max_area = 0
    i = 0
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):</pre>
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
               (<u>i</u> if isEmpty(s)
                  else (i - 1) - getTop(s))
             max_area = max(max_area, area_with_top)
    return max_area
```

```
Values
                         12
                   1
                                               Indices
def getMaxAreaFromHisto(hist: List[Any]) -> int:
    len_hist = len(hist)
    s = [] # type: List[Any]
    max_area = 0
    i = ⊙
    while (i < len_hist) or isNotEmpty(s):</pre>
        if (i < len_hist) and
       (isEmpty(s) or hist[getTop(s)] <= hist[i]):
             s.append(i)
             i += 1
        else:
             tp = s.pop()
             area_with_top = hist[tp] *
              (<u>i</u> if isEmpty(s)
                  else (i - 1) - getTop(s))
            max_area = max(max_area, area_with_top)
    return max_area
```

