

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <climits>
5 using namespace std;
6
7 vector<int> findThreeLargestNumbers(vector<int> array);
8 void updateLargest(vector<int> *threeLargest, int num);
9 void shiftAndUpdate(vector<int> *largest, int num, int idx);
10
11 // O(n) time | O(1) space
12 vector<int> findThreeLargestNumbers(vector<int> array) {
13     vector<int> threeLargest{INT_MIN, INT_MIN, INT_MIN};
14     for (int num : array) {
15         updateLargest(&threeLargest, num);
16     }
17     return threeLargest;
18 }
19
20 void updateLargest(vector<int> *threeLargest, int num) {
21     if (num > threeLargest->at(2)) {
22         shiftAndUpdate(threeLargest, num, 2);
23     } else if (num > threeLargest->at(1)) {
24         shiftAndUpdate(threeLargest, num, 1);
25     } else if (num > threeLargest->at(0)) {
26         shiftAndUpdate(threeLargest, num, 0);
27     }
28 }
29
30 void shiftAndUpdate(vector<int> *array, int num, int idx) {
31     for (int i = 0; i <= idx; i++) {
32         if (i == idx) {
33             array->at(i) = num;
34         } else {
35             array->at(i) = array->at(i + 1);
36         }
37     }
38 }
39
```

Solution 1   Solution 2   Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 vector<int> findThreeLargestNumbers(vector<int> array) {
5     // Write your code here.
6     return {};
7 }
8
```

Our Tests

Custom Output

Submit Code

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

