

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using namespace std;
4
5 class OrgChart {
6 public:
7     char name;
8     vector<OrgChart*> directReports;
9
10    OrgChart(char name) {
11        this->name = name;
12        this->directReports = {};
13    }
14
15    void addDirectReports(vector<OrgChart*> directReports);
16 };
17
18 struct OrgInfo {
19     OrgChart *lowestCommonManager;
20     int numImportantReports;
21 };
22
23 OrgInfo getOrgInfo(OrgChart *manager, OrgChart *reportOne, OrgChart *r
24
25 // O(n) time | O(d) space - where n is the number of people
26 // in the org and d is the depth (height) of the org chart
27 OrgChart *getLowestCommonManager(OrgChart *topManager, OrgChart *repor
28     OrgChart *reportTwo) {
29     return getOrgInfo(topManager, reportOne, reportTwo).lowestCommonMana
30 }
31
32 OrgInfo getOrgInfo(OrgChart *manager, OrgChart *reportOne,
33     OrgChart *reportTwo) {
```

Solution 1

Solution 2

Solution 3

```
1 using namespace std;
2
3 class OrgChart {
4 public:
5     char name;
6     vector<OrgChart*> directReports;
7
8     OrgChart(char name) {
9         this->name = name;
10        this->directReports = {};
11    }
12
13    void addDirectReports(vector<OrgChart*> directReports);
14 };
15
16 OrgChart *getLowestCommonManager(OrgChart *topManager, OrgChart *repor
17     OrgChart *reportTwo) {
18     // Write your code here.
19     return NULL;
20 }
21
```

Our Tests

Custom Output

Submit Code

1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.

2

3 using namespace std;

4

5 class OrgChart {

6 public:

7 char name;

8 vector<OrgChart*> directReports;

9

10 OrgChart(char name) {

11 this->name = name;

12 this->directReports = {};

13 }

14

15 void addDirectReports(vector<OrgChart*> directReports);

16 };

17

18 struct OrgInfo {

19 OrgChart *lowestCommonManager;

20 int numImportantReports;

21 };

22

23 OrgInfo getOrgInfo(OrgChart *manager, OrgChart *reportOne, OrgChart *r

24

25 // O(n) time | O(d) space - where n is the number of people

26 // in the org and d is the depth (height) of the org chart

27 OrgChart *getLowestCommonManager(OrgChart *topManager, OrgChart *repor

28 OrgChart *reportTwo) {

29 return getOrgInfo(topManager, reportOne, reportTwo).lowestCommonMana

30 }

31

32 OrgInfo getOrgInfo(OrgChart *manager, OrgChart *reportOne,

33 OrgChart *reportTwo) {

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 standardize.seed(1234, random = getRNGseed())
2 standardize.seed(1234, random = random)
3 string_output = standardize.string_output()
4 for col in standardize
5     standardize.insert_col, col standardize
6 }
7
8 standardize.add(col to standardize)
9     standardize.add(col)
10     standardize.add(col)
11     standardize.add(col)
12     standardize.add(col)
13     standardize.add(col)
14 }
```

Run or submit code when you're ready.