Solution 1

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

using namespace std;

class OrgChart {
public:
  char name;
  vector<OrgChart *> directReports;

  OrgChart(char name) {
    this->name = name;
    this->directReports = {};
  }

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

struct OrgInfo {
  OrgChart *lowestCommonManager;
  int numImportantReports;
};

OrgInfo getOrgInfo(OrgChart *manager, OrgChart *reportOne, OrgChart *reportTwo);

// O(n) time | O(d) space - where n is the number of people
// in the org and d is the depth (height) of the org chart
OrgChart *getLowestCommonManager(OrgChart *topManager, OrgChart *reportOne,
                                 OrgChart *reportTwo) {
  return getOrgInfo(topManager, reportOne, reportTwo).lowestCommonManager;
}

OrgInfo getOrgInfo(OrgChart *manager, OrgChart *reportOne,
                   OrgChart *reportTwo) {
  int numImportantReports = 0;
  for (OrgChart *directReport : manager->directReports) {
    OrgInfo orgInfo = getOrgInfo(directReport, reportOne, reportTwo);
    if (orgInfo.lowestCommonManager != NULL)
      return orgInfo;
    numImportantReports += orgInfo.numImportantReports;
  }
  if (manager == reportOne || manager == reportTwo)
    numImportantReports++;
  OrgChart *lowestCommonManager = numImportantReports == 2 ? manager : NULL;
  OrgInfo newOrgInfo = {lowestCommonManager, numImportantReports};
  return newOrgInfo;
}

```