Ryan Solanki
Alan Vuong
Quan Nguyen
Reed Wilson

Big O Analysis College Touring Project

Method 1: Plan Trip - O(n) - n + n + 1 + 1 + 1 + 1 + 1 + (n+1) + 1 + 1 = 3n + 8

```
void pathCustom::on_planTrip_button_clicked()
{
  CheckboxChanged(); +n (Runs “CheckboxChanged” at O(n) time)
  efficiencyAlgo(&collegeNamesVector, &sortedCollegeNamesVector, &collegesByDistance,
ui->selectStartingCampus->currentText() +1); +n (Runs “efficiencyAlgo” at O(n) time) +1 for
ui->selectStartingCampus->currentText()
  QWidget *container = new QWidget; +1
  QVBoxLayout *vBoxLayout = new QVBoxLayout; +1

  container->setLayout(vBoxLayout);  +1

  ui->scrollArea_displayTrip->setWidget(container); +1

  total loop instructions: n + 1
  for(int i = 0; i < collegeNamesLabelVector.size(); i++) + 1 (loop overhead) + n (loop contents)
  {
    vBoxLayout->addWidget(collegeNamesLabelVector[i]); +1 add widget in loop
  }
  ui->startTrip_button->show(); +1
  ui->planTrip_button->hide(); +1
}
```

Method 3: efficiencyAlgo - O(n) - 12 + n + 1(loop overhead) + 10n = 11n + 13

```
void pathCustom::efficiencyAlgo(QVector<QString> *colleges,
                QVector<QString> *routeNames,
                QVector<double> *routeDistances,
                QString currentCollege)
{
   if(colleges->empty())+1 (Selection) { return; }

   QString nextSchool;          +1/
   double temp = 0;             +1
   double distance = 0;         +1
   double minDist = 1000000;    +1
   int minIndex;                +1

    total loop instructions: 1(loop overhead) + 10n
   for(int i=0; i < colleges->size(); i++) {

      QSqlQuery *query = new QSqlQuery();+1

      query->prepare("SELECT * FROM Colleges WHERE "
                  "Colleges.starting_college == '" + currentCollege + "' AND "
                  "Colleges.ending_college == '" + colleges->at(i) + "'"); +1

      if(query->exec())+1 (Selection)  {
         query->next(); +1
         distance = query->value(2).toDouble();+1

      temp = distance;+1

      if ( temp < minDist )+1 (Selection)  {
         minDist = temp;+1
         nextSchool = colleges->at(i);+1
         minIndex = i;+1
      }
   }
}

   colleges->erase(colleges->begin()+minIndex); +1
   QLabel* tempSchool = new QLabel(nextSchool); +1
   collegeNamesLabelVector.push_back(tempSchool); +1
   routeNames->push_back(nextSchool); +1
   routeDistances->push_back(minDist); +1
   totalDistance = totalDistance + minDist; +1
```

```
    efficiencyAlgo(colleges, routeNames, routeDistances, nextSchool); +n for recursion
}
```

Method 3: CheckboxChanged() - O(n) - 1 + (3n + 1) + 1 + (2n+1) = 5n + 4

```
void pathCustom::CheckboxChanged()
{
    int checkedCount = 0; +1

Total loop contents 3n + 1
    for(int i = 0; i < checkBoxVector.size(); i++)  + 1 (loop overhead) + n (loop contents)
    {
        if(checkBoxVector[i]->checkState() == Qt::CheckState::Checked) +1 selection
        {
            collegeNamesVector.push_back(tempcollegeNamesVector[i]); +1
            checkedCount++; +1
        }
    }

//Choose largest selection statement
    if(checkedCount == 11) +1 selection
    {
Total loop contents 2n + 1
        for(int i = 0; i < checkBoxVector.size(); i++)  + 1 (loop overhead) + n (loop contents)
        {
            if(checkBoxVector[i]->checkState() == Qt::CheckState::Unchecked) +1 selection
            {
                checkBoxVector[i]->setDisabled(true); +1
            }
        }
    }
Disregard smaller selection statement
    else
    {
        for(int i = 0; i < checkBoxVector.size(); i++)
        {
            checkBoxVector[i]->setDisabled(false);
        }
    }
}
```