Algorithms TSP

Chapter 1

TSP

Author

Christian Prather

Version

1.0

Date

2020-1-31

Warning

Do not run over n = 11

1.1 Compilation

To compile this was built on a linx system with gcc and was only ever tested on a linux system was testes with clion

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Point Class Reference

`#include <Point.h>`

Public Member Functions

- Point (int x, int y)
- Point ()

Public Attributes

- int x
- int y

Friends

- bool operator< (const Point &leftPoint, const Point &rightPoint)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 Point() [1/2]

```
Point::Point (
            int x,
            int y )  [inline]
```

4.1.1.2 Point() [2/2]

```
Point::Point ( )  [inline]
```

4.1.2 Friends And Related Function Documentation

4.1.2.1 operator<

```
bool operator< (
            const Point & leftPoint,
            const Point & rightPoint )  [friend]
```

4.1.3 Member Data Documentation

4.1.3.1 x

```
int Point::x
```

4.1.3.2 y

```
int Point::y
```

The documentation for this class was generated from the following file:

- /home/christian/Documents/Algorithms_Project1/Point.h

Chapter 5

File Documentation

5.1 /home/christian/Documents/Algorithms_Project1/CMakeLists.txt File Reference

5.2 /home/christian/Documents/Algorithms_Project1/exhaustive.cpp File Reference

```
#include "exhaustive.h"
#include "helpers.h"
#include <algorithm>
```

Functions

- int getPermutations ()
- double getMinDist ()
- void createPermutations (vector< Point > allPoints, int numberOfPoints)
- void calculateExhaustive ()

Variables

- int permutations = 0
- double minDist
- vector< vector< Point > > potentialPaths

5.2.1 Function Documentation

5.2.1.1 calculateExhaustive()

```
void calculateExhaustive ( )
```

5.2.1.2 createPermutations()

```
void createPermutations (
            vector< Point > allPoints,
            int numberOfPoints )
```

5.2.1.3 getMinDist()

```
double getMinDist ( )
```

5.2.1.4 getPermutations()

```
int getPermutations ( )
```

5.2.2 Variable Documentation

5.2.2.1 minDist

```
double minDist
```

5.2.2.2 permutations

```
int permutations = 0
```

5.2.2.3 potentialPaths

```
vector<vector<Point> > potentialPaths
```

5.3 /home/christian/Documents/Algorithms_Project1/exhaustive.h File Reference

```
#include <algorithm>
#include <iostream>
#include <vector>
#include "Point.h"
```

Functions

- void createPermutations (vector< Point > allPoints, int numberOfPoints)
- void calculateExhaustive ()
- int getPermutations ()
- double getMinDist ()

5.3.1 Function Documentation

5.3.1.1 calculateExhaustive()

```
void calculateExhaustive ( )
```

5.3.1.2 createPermutations()

```
void createPermutations (
            vector< Point > allPoints,
            int numberOfPoints )
```

5.3.1.3 getMinDist()

```
double getMinDist ( )
```

5.3.1.4 getPermutations()

```
int getPermutations ( )
```

5.4 /home/christian/Documents/Algorithms_Project1/helpers.cpp File Reference

```
#include "helpers.h"
#include "math.h"
#include <time.h>
```

Functions

- double calculateDistance (Point current, Point option)
- int generateRandomPoints (int min, int max)
- int generateRandomN ()
- int generateBigRandomN ()

5.4.1 Function Documentation

5.4.1.1 calculateDistance()

```
double calculateDistance (
            Point current,
            Point option )
```

5.4.1.2 generateBigRandomN()

```
int generateBigRandomN ( )
```

5.4.1.3 generateRandomN()

```
int generateRandomN ( )
```

5.4.1.4 generateRandomPoints()

```
int generateRandomPoints (
            int min,
            int max )
```

5.5 /home/christian/Documents/Algorithms_Project1/helpers.h File Reference

```
#include "Point.h"
```

Functions

- double calculateDistance (Point current, Point option)
- void addToPath (Point nextPoint)
- int generateRandomPoints (int min, int max)
- int generateRandomN ()
- int generateBigRandomN ()

5.5.1 Function Documentation

5.5.1.1 addToPath()

```
void addToPath (
            Point nextPoint )
```

5.5.1.2 calculateDistance()

```
double calculateDistance (
            Point current,
            Point option )
```

5.5.1.3 generateBigRandomN()

```
int generateBigRandomN ( )
```

5.5.1.4 generateRandomN()

```
int generateRandomN ( )
```

5.5.1.5 generateRandomPoints()

```
int generateRandomPoints (
            int min,
            int max )
```

5.6 /home/christian/Documents/Algorithms_Project1/main.cpp File Reference

```
#include <iostream>
#include <fstream>
#include "Point.h"
#include "helpers.h"
#include "exhaustive.h"
#include <sstream>
#include <queue>
#include <chrono>
#include <thread>
```

Functions

- void loadFile (string fileName)
- void generateFiles (int min, int max)
- void primaryLogic (Point startPoint)
- void buildRandomList (int n, int min, int max)
- void cleanUp ()
- void analysis ()
- int main ()

Variables

- int numberPoints = 10
- double finalDistance = 0
- int numberOfFiles = 4
- vector< Point > thePath
- vector< Point > allPoints
- vector< Point > copyAllPoints

5.6.1 Function Documentation

5.6.1.1 analysis()

```
void analysis ( )
```

Primary analysis file

5.6.1.2 buildRandomList()

```
void buildRandomList (
            int n,
            int min,
            int max )
```

Used when testing manually

5.6.1.3 cleanUp()

```
void cleanUp ( )
```

5.6.1.4 generateFiles()

```
void generateFiles (
            int min,
            int max )
```

\ A function for generating test files

5.6.1.5 loadFile()

```
void loadFile (
            string fileName )
```

\ A function for loading a parsing test files

5.6.1.6 main()

```
int main ( )
```

5.6.1.7 primaryLogic()

```
void primaryLogic (
            Point startPoint )
```

\Function used for nearest-neigbor calculations

5.6.2 Variable Documentation

5.6.2.1 allPoints

```
vector<Point> allPoints
```

5.6.2.2 copyAllPoints

```
vector<Point> copyAllPoints
```

5.6.2.3 finalDistance

```
double finalDistance = 0
```

5.6.2.4 numberOfFiles

```
int numberOfFiles = 4
```

5.6.2.5 numberPoints

```
int numberPoints = 10
```

5.6.2.6 thePath

```
vector<Point> thePath
```

5.7 /home/christian/Documents/Algorithms_Project1/Point.cpp File Reference

```
#include "Point.h"
```

5.8 /home/christian/Documents/Algorithms_Project1/Point.h File Reference

```
#include <vector>
```

Classes

- class Point

Index