

CSCI 136 Assignment – C++ , The Need for Speed

CSCI 136 Assignment – C++ , The Need for Speed

April 19th, 2023

Due Date: Thursday, 4/27/2023 11:59PM

Points: 30

Topic: C++ , The Need for Speed

In this assignment, you will gain experience programming in C++, using arrays, and reading from files. You will implement a greedy heuristic algorithm for the traveling salesman problem that you also used in a prior assignment in Python. This time you will use arrays instead of linked-lists.

After following all instructions below, please submit your C++ source file in Moodle. Only submit the .cpp file, not the project files or executables. Upload this as an individual file – do not zip it. Be sure the source file has a header section that includes the file name, your name, credits, a description of the file, and other useful information. Make sure your code is commented in general and thoroughly.

- `GreedyPath.cpp`

Make sure you read the Extensions section as there is an opportunity to earn extra points towards your overall final class percent score via this assignment.

Grading:

This assignment is worth 30 points total. You will be graded according to the following criteria:

Grade Item	Points
Program Compiles	5
Program Runs	5
Header Comment	3
Number of Points Output	2
Correct Calculation of Tour Distance	10
Time Elapsed Output	5
Total	30

GreedyPath.cpp

You will develop a program C++ that does the following:

- Reads a text file specified in the command line which contains coordinate points to be visited by the traveling salesman.
- Outputs the number of points.
- Constructs a path visiting each point exactly once and returning to the starting point. The path starts from the first point in the file. You will construct the path in a *greedy* fashion, going next to the closest unvisited point. In the case of a tie, visit the first of the closest unvisited points encountered in the file.
- Computes and outputs the total distance of the greedy path you found through the points including the distance from the last point back to the start.
- Computes and outputs the runtime of the program from start to finish in seconds.

The file `GreedyPathDataFiles.zip` contains several example input files. Here are the contents of one of the provided input files, `points7.txt`:

```
7
0.0  1.0
1.0  1.0
-1.0  1.0
2.0  0.0
-2.0 -2.0
1.0 -2.0
2.0 -2.0
```

First you will read the point data from the file and put it into arrays. For this program, use "parallel" arrays: one array for the x values and one for the y values. Assume that the input file is well-formed and contains at least one point. The first line of the input file is an integer indicating how many points are in the file. The subsequent lines include the x, y coordinates with one x-y pair per line.

The path you construct must start at the first point in the file. From there, the path goes to the next closest point that has not already been visited. For example, in `points7.txt`, starting at $(0.0, 1.0)$, the next closest point is $(1.0, 1.0)$. From $(1.0, 1.0)$, the next closest point would be $(0.0, 1.0)$, but since we already visited this one, the nearest unvisited point is $(2.0, 0.0)$. We would proceed from there in a similar fashion.

You will need to use your arrays of x- and y-coordinates to determine which point is closest given your current location. In the event of a tie, you will use the point that occurs first in the file and has not been visited yet. You will need to keep track of whether each point has been visited or not.

You need to compute the total distance covered by your path. You could do this in a cumulative manner, as you add points to the path, or at the end after you have formed the full path. Be sure to include the distance from the last point back to the first/starting point, as the traveling salesman must return to his starting/home point.

Here is the complete output for the provided files:

> GreedyPath.exe points7.txt

Number of Points = 7
Total Distance = 12.5765
Elapsed Time (sec) = 0

> GreedyPath.exe points5.txt

Number of Points = 5
Total Distance = 5.67899
Elapsed Time (sec) = 0

> GreedyPath.exe points3.txt

Number of Points = 3
Total Distance = 4.82843
Elapsed Time (sec) = 0

> GreedyPath.exe points3b.txt

Number of Points = 3
Total Distance = 4.82843
Elapsed Time (sec) = 0

> GreedyPath.exe tsp10.txt

Number of Points = 10
Total Distance = 1595.87
Elapsed Time (sec) = 0

> GreedyPath.exe tsp100.txt

Number of Points = 100
Total Distance = 5543.33
Elapsed Time (sec) = 0

> GreedyPath.exe tsp1000.txt

Number of Points = 1000
Total Distance = 18246
Elapsed Time (sec) = 0

> GreedyPath.exe usa13509.txt

Number of Points = 13509
Total Distance = 48699.6
Elapsed Time (sec) = 0

Yes, the elapsed time was 0 seconds for all of my test runs, including on usa13509.txt. This means that my program ran in less than 0.5 seconds for all of these. Yours might take longer ...

Helpful Hints:

Do I need to follow the given specifications? Yes. You must implement the individual program as specified.

Are there any other helpful hints? Yes.

- Look at the examples given in class this past Monday. You will find a good part of what you need in them. They will be discussed more on Friday.
- Look at the prior C++ lab assignment and associated slides.
- Look back at the prior TSP assignment in Python, recognizing that you are to use.
- Apply what you have learned this year.
- Think about what is involved in the computations you make.

Submit Your Work

After following all instructions above, please submit your C++ source file in Moodle. Only submit the .cpp file, not the project files or executables. Upload this as an individual file – do not zip it. Be sure the source file has a header section that includes the file name, your name, credits, a description of the file, and other useful information. Make sure your code is commented in general and thoroughly.

- `GreedyPath.cpp`

EXTENSIONS

For this assignment, you have the opportunity to complete the extension and earn extra points above and beyond regular score for your assignment. These points will be added as percentage points towards your final overall class percentage score. For instance, if you overall class percentage score is 0.88 at the end of the semester and you earn 5 extra points on this extension, your overall class percentage score will be increased to a 0.93, which would take you from an B+ to a A.

The objective of the extension is straight-forward: make your program run as fast as possible while still correctly producing the desired result. If you choose to take this challenge, you will do so within the same C++ file you submit for this assignment. Make sure you follow all the prior guidelines for this so you get full-credit. Do not break your submission by trying to do this extension.

You must generally follow constructs we used in class and specified for this lab to date. You may not introduce constructs that go outside the scope of what we covered. For instance, you are tasked with using C++ arrays to store points and their coordinates. You should not use different

CSCI 136 Assignment – C++ , The Need for Speed

structures for this purpose. You must also document via comments in your file what specifically you did to improve the performance/speed. Include an overview in the header comment and point out specific items in your code that you did for the purpose of efficient computation.

For this extension, you will be evaluated on one or several data sets that are not provided with the assignment. These data sets have on the order of 10,000 to 100,000 or more points in them! They will generally take longer than a second or two to run. However, they should take no longer than one or several minutes, certainly not hours.

Here are the potential points to be earned on this extension. These are progressive in that you need to fulfill requirements of prior levels.

BONUS POINTS	Requirements
1 point	<u>Your computations and output are correct per the guidelines given in the assignment and finish within a reasonable amount of time. You included comments detailing what you did to improve performance.</u>
2.5 points	Your computations and output are correct per the guidelines given in the assignment and finish within a reasonable amount of time. You included comments detailing what you did to improve performance. And, <u>your program was tied for fastest in the class.</u>
5 points	Your computations and output are correct per the guidelines given in the assignment and finish within a reasonable amount of time. You included comments detailing what you did to improve performance. And, <u>your program was the single fastest in the class.</u>
10 points	Your computations and output are correct per the guidelines given in the assignment and finish within a reasonable amount of time. You included comments detailing what you did to improve performance. And, <u>your program was faster than my (Galarus') program.</u>

Any suspected instances of code-sharing, copyright violation, or cheating of any kind will automatically result in a score of zero and will be handled separately as per guidelines given in the course syllabus and by Montana Tech. It is in your best interest to produce your own solution to the problem. It is not in your best interest to do otherwise.