Job description

A. Apply the I5H algorithm for vectors in d-dimensional space based on the Manhattan metric
(1), as well as the random projection algorithm on the hypercube for the same metric. The program will
implemented so that, taking as input vector a, it returns a) the nearest neighbor to a
and (opiis) the vectors within radius B from a. (The design of the code should allow
its easy extension to vector spaces with another metric, e.g., p-PoIpi, or different spaces.

B. Implement the LSH algorithm and the random projection algorithm for polygonal curves such as
were presented in the lesson on finding, within a set of polygonal curves, the nearest
curve earth a. Each polygonal curve is defined as a sequence of points in Euclidean space

12, with a different number of points. The program will support the distance function between
Dynamic Time warping curves.

i) To LSH KaumuAwy otic StaAé€etc ametkoviel KALMUAEG O€ KKALTUAEG TIAEYLLaTOG» (grid curves), ol oTToiEG
will be represented as vectors. Implement storage and retrieval on mesh curves
a) with the I5H algorithm for metric Margin and b) with the random projection algorithm in
hypercube for the same metric.

[) Random projection also reduces curves to vectors and performs vector search. That's all
result from the possible matches (ĩ{anesthi) of the data with curves as long as possible
length of the query curve (amyY). The search in the vectors that represent the curves
it will also be done a) with the 15H algorithm for the metric of Miami and b) with the randomization algorithm
hypercube projection for the same metric.

The design of the code should allow it to be easily extended for different functions
distance curves as well as the future use of individual functions in future ones tasks,

ENTRANCE

A,

1) A tab-separated text file for the data set input (ααίαεĩ)
(i80-seraia6a!), with the following notation:
item idl X11 X12 wee Xld
item idN XN1 XN2 wee ANd

onou Xij Stavbopatoc i otov d-didotato EukAgideto ywpo. Ta ovopata (item_idK) can be
unique integers or strings.

2) Text file containing the search set i.e. the vectors α, and contains at least one vector. (Oiiis) The positive aevith B is given in the first line. When XX0 is given program finds only the nearest neighbor of the vectors in the data set.

The program initially asks the user for the path of ἀαιαει. After creating the structure search, the program prompts the user for the path of the search file and the file output. After running the algorithm and producing the results, the program prompts by the user if he wants to terminate the program or repeat the search for a different one set / search file. The search file has the following format:

```
Radius: <double> //bonus
item 1491 X11 X12 wee X1ld
item idSQ koi XΩ2 wee XQd
```

Item names in the search set item_idSj (1<=j<=Q) can be unique integers or strings.

For C5H, the following parameters can be given optionally on the command line: integer k of the Γ5H functions Pi used to define ᾳ, the integer |. of the tables fragmentation. Anda k, |. are not given, the program uses invalid values Kξ4, L=5.

For the random projection on the hypercube, the following optional parameters can be given to command line: the dimension in which the points K(ξα1, the maximum number allowed of candidate points to be checked M, the maximum allowed number of vertices of the hypercube probes will be checked. Ot default tec etvat: K=3, M=10, probes=2.

Input and search files can also be given via command line parameters. So the execution will be done through the command:

```
S./lsh -d <input file> -q <query file> -k <int> -L <int> -o <output file>
```

```
S./cube -d <input file> -q <query file> -k <int> -M <int> -probes <int> -o
<output file>
```

B.
1) A text file for the input of the SedSouévwv set (apyeio dataset) of polygons
of curves separated by colons (180-σ6ραΓαΐΐθς), which will have the following
notation:

curve idl mL (x11,y11) (x12,y12) wee (xlm1i,ylml1l)

curve _idN mN (xN1,yN1) (xN2,YN2) eee (xNmN, ylmN)

Onou (xij,yij) olouvtetaypévec double tou j onueiou of thc curve], where | "More
and more multitudes of
points of the curve |.

2) Text file that includes the set of curves for which we are looking for
nearest neighbor: this is the search set, which contains at least one curve (file
search).

The program initially asks the user for the path of the desired file, the algorithm
that
is selected for curve search (case ΐ or ΐ and the search algorithm of
of vectors resulting from the curves (15H or hypercube). After creating the
structure
search, the program prompts the user for the path of the search file and the file
output of the results. After running the algorithm and producing the results,
the program asks the user if he wants to end the program or if he wants to repeat
the
search for a different search set / file. The file has the following format for
problems in vector space, respectively according to the format of the input file:

curve _ids1 them]. (xS11,yS11) (xS12,yS12) eae (xSlmS1,ySi1mS1)

curve _idSN mSN (xSN1,ySN1) (xSN2,ySN2) eae (xSNmSN, ySNmSN)

Curve names in the search set eoiine i65] (1«Xi«XΩ) can be unique
integers or strings.

Optional napdyetpoc can be given on the command line:
e For 5H curves, the integer {. of the mesh curves per input curve, while if
dev Sivetat, ypnoworoteitat default ty L_grid=4.
o For the random projection algorithm, the factor can be given on the command line
approximation ε, while if not given, the absolute value εXth.5 is used.

Input and search files can also be given via command line parameters,
so the execution will be done through the command:

```
$./curve grid lsh -d <input file> -q <query file> -k_ vec <int> -L grid <int> -o
<output file>

$./curve grid hypercube -d <input file> -q <query file> -k hypercube <int> -M
<int> -probes <int> -L_grid -o <output file>
$./curve projection lsh -d <input file> -q <query file> -k_ vec <int> -L_vec <int>
-e <double> -o <output file>

$./curve projection hypercube -d <input file> -q <query file> -k hypercube <int>
-M <int> -probes <int> -e <double> -o <output file>

exit
```

A.

Text file containing for each item in the search set using the suitable labels: a) the name of the approximately nearest earth found and the its distance from ɑ̣, b) the distance of g amd Tov true nearest neighbor (via exhaustive search), Y) the time of finding (a) and d) the time of finding (c) as Kal (bonus) Ta names of H-radius neighbors. The output file must follow the following format:

Query: itemJd

Nearest neighbor: itemY
distanceLSH: <double>
distanceTrue: <double>
tLSH: <double>

tTrue: <double>

R-near neighbors: //bonus
itemJ //bonus
itemk //bonus
itemw //bonus

and so on.

B.

Text file containing for each curve of the search set using the proper tags? a) the name of the nearest neighbor found by I5H / algorithm random projection, and its distance from 6. The output file must follow the template below:

Query: curved

Method: {LSH, Projection}
HashFunction: {LSH, Hypercube}
Found Nearest neighbor: curveY
True Nearest neighbor: curveX
distanceFound: <double>
distanceTrue: <double>

Compare the results of the 4 variant algorithm [15H {lines/ 15H 1.1 5H {the curves / Hypercube, Random Projection / LSH L1, RandomProjection / Hypercube] we mpo₵ a) To péytoTo (of all the objects of the search set) proximity fraction Ξ Approximate closest distance earth / True nearest neighbor distance, b) the average time to find it approximately nearest neighbor. Comment the results in the program documentation.

Additional requirements

1)

The program should be well organized with separation of declarations / definitions functions, structures, and data types into logical groups that correspond to separate header and source code files. Code quality (eg avoid PIEPIOI |eaKs). Compiling the program must be done using the PpiãKe tool and the existence of a suitable MaKeῄIe.

the deliverable must be adequately documented with full code commentary and the existence of file which includes at least: a) title and description of the program, b) list of code / header files and their description, c) its compilation instructions of the program, ӧ) instructions for using the program and e) full details of the students who they developed.

The implementation of the program should be done using a version management system software and collaboration (aἰτ).