Implement LSH

Group members : ¶

1. Mintra Sojiphan 6220422057 DS6

2. Kavin Singhakhet 6310422040 DS7

3. Porawoot Buranadilok 6220422064 DS6

We implement LSH using two packages with two differnt datasets

1. Package: SnaPy // Datasets: A Corpus of Plagiarised Short Answers

References: https://github.com/justinnbt/SnaPy (https://github.com/justinnbt/SnaPy)

https://ir.shef.ac.uk/cloughie/resources/plagiarism_corpus.html (https://ir.shef.ac.uk/cloughie/resources/plagiarism_corpus.html)

2. Package: datasketch // Dataset: News headlines

References: http://ekzhu.com/datasketch/lshforest.html (http://ekzhu.com/datasketch/lshforest.html)

1. Package: SnaPy // Datasets: A Corpus of Plagiarised Short Answers

We will implement LSH on plagiarised answers of five different questions namely question A, B, C, D, and E to identify near duplicate answers of each question using Jaccard similarity threshold (s) = 0.5

Step 1: Install and import packages

In [ ]:

```
pip install snapy
```

```
Collecting snapy
  Downloading https://files.pythonhosted.org/packages/68/59/cdf153f7a39159
3060a3fcd06f13f8127dc79f675d99ef576b69f49365a0/snapy-1.0.2-py3-none-any.wh
l
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-pack
ages (from snapy) (1.19.5)
Installing collected packages: snapy
Successfully installed snapy-1.0.2
```

In [ ]:

```
pip install mmh3
```

```
Collecting mmh3
  Downloading https://files.pythonhosted.org/packages/fa/7e/3ddcab0a9fcea0
34212c02eb411433db9330e34d626360b97333368b4052/mmh3-2.5.1.tar.gz
Building wheels for collected packages: mmh3
  Building wheel for mmh3 (setup.py) ... done
  Created wheel for mmh3: filename=mmh3-2.5.1-cp36-cp36m-linux_x86_64.whl
size=37849 sha256=3f28dc26ac98dc67078e2b1835f83f8d0ceb68783fbb8283be366793
d5b16299
  Stored in directory: /root/.cache/pip/wheels/38/b4/ea/6e4e321c625d3320c0
c496bf4088371546d8fce5f1dd71b219
Successfully built mmh3
Installing collected packages: mmh3
Successfully installed mmh3-2.5.1
```

In [ ]:

```
from snapy import MinHash, LSH
from google.colab import drive
import matplotlib.pyplot as plt
import networkx as nx
```

Step 2: Connect Google Colab with Google drive

In [ ]:

```
drive.mount("/content/drive")
```

```
Mounted at /content/drive
```

Step 3: Create function to perform LSH

In [ ]:

```
def LSH_task(task):

  #import data
  files=[]
  x='/content/drive/MyDrive/corpus/'+task+'.txt'
  files.append(x)
  for i in range(1,20):
    z='/content/drive/MyDrive/corpus/'+task+ ' ('+str(i)+').txt'
    files.append(z)

  docs=[]
  for file in files:
    file = open(file)
    text = file.read()
    docs.append(text)
  file.close()

  print(docs)
  print()
  print('#task: ',len(docs))
  labels=[]
  labels.append(task)
  for i in range(1,20):
    a=task+'('+str(i)+')'
    labels.append(a)

  seed =3

  #Create MinHash object.
  minhash = MinHash(docs, n_gram=9, permutations=100, hash_bits=64, seed=3)
  print('Signatures metric: ',len(minhash.signatures),'x',len(minhash.signatures[0]))
  print('#permutations used to create signatures:',minhash.permutations)
  #print('Minhash Signatures for each text:')
  #for i in minhash.signatures:
    #print(i)

  # Create LSH model.
  lsh = LSH(minhash, labels, no_of_bands=50)

  # Query to find near duplicates for each doc.
 # print()
  #for i in labels:
    #print('Near duplicate for answer',i,':',lsh.query(i, min_jaccard=0.1))

  # Check contents of documents.
  #print(lsh.contains())

  # Return adjacency list for all similar texts.
  adjacency_list = lsh.adjacency_list(min_jaccard=0.1)
  #print()
  #print('adjacency_lists: ',adjacency_list)
  #print()
  # Returns edge list for use creating a weighted graph.
  edge_list = lsh.edge_list(min_jaccard=0.1, jaccard_weighted=True)
  #print('edge lists: ',edge_list)
  #print()

  # Create Undirected weighted graph.
  fig=plt.figure(figsize =(10,6))
```

```
    fig.set_facecolor("#181818")
    title="Near duplicate answer of question "+task
    fig.suptitle(title,color= '#cccccc',fontsize=18)
    G = nx.Graph()
    for i in edge_list:
      G.add_edge(i[0],i[1],weight=i[2])
    e1=[(u,v) for (u,v,d) in G.edges(data=True) if d['weight'] > 0.5]
    e2=[(u,v) for (u,v,d) in G.edges(data=True) if d['weight'] <= 0.5]
    pos=nx.spring_layout(G)

    # nodes
    nx.draw_networkx_nodes(G,pos,node_size=900,node_color='yellow')

    # edges
    edge1=nx.draw_networkx_edges(G,pos,edgelist=e1,width=4,edge_color='red')
    edge2=nx.draw_networkx_edges(G,pos,edgelist=e2,width=1,edge_color='red',style='dashe
d')

    # labels
    nx.draw_networkx_labels(G,pos,font_family='sans-serif',font_size=10,font_color='#0000
00',font_weight=150)
    keys=[(i[0],i[1]) for i in edge_list]
    values= [(i[2]) for i in edge_list]
    edge_labels = dict(zip(keys, values))
    nx.draw_networkx_edge_labels(G,pos,edge_labels= edge_labels,font_color='red')
    fig.set_facecolor("#181818")
    plt.axis('off')
    fig.legend((edge1, edge2), ('sim(i,j) > 0.5', 'sim(i,j) <= 0.5'),loc=1,fontsize=12)
    plt.savefig("LSH_graph.png")
    plt.show()
```

Step 4: Perform LSH on set of 20 plagiarised answers of the following questions to identify near duplicate answers using Jaccard similarity threshold (s) = 0.5

Question A: The result illustrates that there are 3 pairs of answers which can be identified as near duplicate answers as shown below:

1. Answer A(1) and A(18) with Jaccard similarity of 0.9

2. Answer A(1) and A(6) with Jaccard similarity of 0.74

3. Answer A(6) and A(18) with Jaccard similarity of 0.7

In [ ]:

```
LSH_task('A')
```

```
['Object oriented programming is a  style of programming that supports enc
apsulation, inheritance, and polymorphism. Inheritance means derived  a ne
w class from the base class.  We can also say there are parents class and
child classes in inheritance.  Inheritance was firstly derived in 1967.\nT
he child class has all the features of parents class or we can say the bas
e class more over it may also include some additional features.  Inheritan
ce is used for modification and implementation new features in computer pr
ogramming language.It is possible that child class has all the attributes
of parents class but it is not possible that all the attributes of child c
lass must have in base class or parent class.\nI categorization in compute
r language also inheritance is a useful tool.categorization define as a po
werful  feature.it  has been also used in generalisation and in human lear
ning. In some areas less information need to be stored.\nGenerlisation als
o some time known as inheritance. The main reason behind this is a hierarc
hi structure of objects and classes. We can understand  this mechanism by
some examples: like fruit is aq main class and mangoes apple ,orange is ch
ild classs of the main class.So obviously inherit all the properties of fr
uit class.\n', 'In object-oriented programming, inheritance is a way to fo
rm new classes (instances of which are called objects) using classes that
have already been defined. The inheritance concept was invented in 1967 fo
r Simula.\n\nThe new classes, known as derived classes, take over (or inhe
rit) attributes and behavior of the pre-existing classes, which are referr
ed to as base classes (or ancestor classes). It is intended to help reuse
existing code with little or no modification.\n\nInheritance provides the
support for representation by categorization in computer languages. Catego
rization is a powerful mechanism number of information processing, crucial
to human learning by means of generalization (what is known about specific
entities is applied to a wider group given a belongs relation can be estab
lished) and cognitive economy (less information needs to be stored about e
ach specific entity, only its particularities).\n\nInheritance is also som
etimes called generalization, because the is-a relationships represent a h
ierarchy between classes of objects. For instance, a "fruit" is a generali
zation of "apple", "orange", "mango" and many others. One can consider fru
it to be an abstraction of apple, orange, etc. Conversely, since apples ar
e fruit (i.e., an apple is-a fruit), apples may naturally inherit all the
properties common to all fruit, such as being a fleshy container for the s
eed of a plant.\n\nAn advantage of inheritance is that modules with suffic
iently similar interfaces can share a lot of code, reducing the complexity
of the program. Inheritance therefore has another view, a dual, called pol
ymorphism, which describes many pieces of code being controlled by shared
control code.\nInheritance is typically accomplished either by overriding
(replacing) one or more methods exposed by ancestor, or by adding new meth
ods to those exposed by an ancestor.\n\nComplex inheritance, or inheritanc
e used within a design that is not sufficiently mature, may lead to the Yo
-yo problem.', 'Inheritance is a basic concept of Object-Oriented Programm
ing where\nthe basic idea is to create new classes that add extra detail t
o\nexisting classes. This is done by allowing the new classes to reuse\nth
e methods and variables of the existing classes and new methods and\nclass
es are added to specialise the new class. Inheritance models the\n“is-kind
-of” relationship between entities (or objects), for example,\npostgraduat
es and undergraduates are both kinds of student. This kind\nof relationshi
p can be visualised as a tree structure, where ‘student’\nwould be the mor
e general root node and both ‘postgraduate’ and\n‘undergraduate’ would be
more specialised extensions of the ‘student’\nnode (or the child nodes). I
n this relationship ‘student’ would be\nknown as the superclass or parent
class whereas, ‘postgraduate’ would\nbe known as the subclass or child cla
ss because the ‘postgraduate’\nclass extends the ‘student’ class.\n\nInher
itance can occur on several layers, where if visualised would\ndisplay a l
arger tree structure. For example, we could further extend\nthe ‘postgradu
ate’ node by adding two extra extended classes to it\ncalled, ‘MSc Studen
t’ and ‘PhD Student’ as both these types of student\nare kinds of postgrad
```

```
uate student. This would mean that both the ‘MSc\nStudent’ and ‘PhD Studen
t’ classes would inherit methods and variables\nfrom both the ‘postgraduat
e’ and ‘student classes’.\n', "Inheritance is a basic concept in object or
iented programming. It models the reuse of existing class code in new clas
ses – the “is a kind of” relationship.\n\nFor example, a house is a kind o
f building; similarly, an office block is a kind of building. Both house a
nd office block will inherit certain characteristics from buildings, but a
lso have their own personal characteristics – a house may have a number of
occupants, whereas an office block will have a number of offices. However,
these personal characteristics don't apply to all types of buildings.\n\nI
n this example, the building would be considered the superclass – it conta
ins general characteristics for other objects to inherit – and the house a
nd office block are both subclasses – they are specific types and speciali
se the characteristics of the superclass.\n\nJava allows object inheritanc
e. When one class inherits from another class, all the public variables an
d methods are available to the subclass.\n\npublic class Shape {\n\n  priv
ate Color colour;\n\n  public void setColour(Color newColour){\n\n    colo
ur = newColour;\n\n  }\n\n}\n\npublic class Circle extends Shape {\n\n  pr
ivate int radius;\n\n  public void setRadius(int newRadius){\n\n    radius
= newRadius;\n\n  }\n\n}\n\nIn this example, the Circle class is a subclas
s of the Shape class. The Shape class provides a public setColour method,
which will be available to the Circle class and other subclasses of Shape.
However, the private variable colour (as defined in the Shape class) will
not be available for direct manipulation by the Circle class because it is
not inherited. The Circle class specialises the Shape class, which means t
hat setRadius is available to the Circle class and all subclasses of Circl
e, but it isn't available to the superclass Shape. \n", 'inheritance in ob
ject oriented programming is where a new class is formed using classes whi
ch have allready been defined. These classes have have some of the behavio
r and attributes which where existent in the classes that it inherited fro
m. The peropos of inheritance in object oriented programming is to minimiz
e the reuse of existing code without modification.\n\nInheritance allowes
classes to be categorized, similer to the way humans catagorize. It also p
rovides a way to generalize du to the "is a" relationship between classes.
For example a "cow" is a generalization of "animal" similarly so are "pig
s" & cheaters". Defeining classes in this way, allows us to define attribu
tes and behaviours which are commen to all animals in one class, so cheate
rs would natuarly inheart properities commen to all animals.\n\nThe advant
age of inheritance is that classes which would otherwise have alot of simi
lar code , can instead shair the same code, thus reducing the complexity o
f the program. Inheritance, therefore, can also be refered to as polymorph
ism which is where many pieces of code are controled by shared control cod
e.\n\nInheritance can be accomplished by overriding methods in its ancesto
r, or by adding new methods. \n', 'Inheritance in object oriented programm
ing is a way to form new classes using classes that have already been defi
ned. The new classes, known as derived classes, inherit attributes and beh
aviour of the existing classes, which are referred to as base classes. Wit
h little or no modification, it is intended to help reuse existing code. I
t is typically accomplished either by overriding one or more methods expos
ed by ancestor, or by adding new methods to those exposed by an ancestor\n
\nInheritance is also sometimes called generalization, because there is-a
relationships represent a hierarchy between classes of objects. A ‘fruit’,
for instance, is a generalization of "orange", "mango", "apples" and many
others. One can consider fruit to be an abstraction of apple, orange, etc.
Since apples are fruit (i.e., an apple is-a fruit), conversely apples may
naturally inherit all the properties common to all fruit, such as being a
fleshy container for the seed of a plant.\n\nAn advantage of inheritance i
s that modules with sufficiently similar interfaces can share a lot of cod
e reducing the complexity of the program. \n', 'In object-oriented program
ming, inheritance is a way to form new classes (instances of which are cal
led objects) using classes that have already been defined. The inheritance
```

```
concept was invented in 1967 for Simula. The new classes, known as derived
classes, take over (or inherit) attribute and behaviour of the pre-existin
g classes, which are referred to as base classes (or ancestor classes). It
is intended to help reuse existing code with little or no modification. In
heritance provides the support for representation by categorization in com
puter languages. Categorization is a powerful mechanism number of informat
ion processing, crucial to human learning by means of generalization (what
is known about specific entities is applied to a wider group given a belon
gs relation can be established) and cognitive economy (less information ne
eds to be stored about each specific entity, only its particularities). In
heritance is also sometimes called generalization, because the is-a relati
onships represent a hierarchy between classes of objects. For instance, a
"fruit" is a generalization of "apple", "orange", "mango" and many others.
One can consider fruit to be an abstraction of apple, orange, etc. Convers
ely, since apples are fruit (i.e., an apple is-a fruit), apples may natura
lly inherit all the properties common to all fruit, such as being a fleshy
container for the seed of a plant. An advantage of inheritance is that mod
ules with sufficiently similar interfaces can share a lot of code, reducin
g the complexity of the program. Inheritance therefore has another view, a
dual, called polymorphism, which describes many pieces of code being contr
olled by shared control code. Inheritance is typically accomplished either
by overriding (replacing) one or more methods exposed by ancestor, or by a
dding new methods to those exposed by an ancestor. \n', 'In object oriente
d programming, objects are grouped together into classes according to thei
r type, structure and the functions that can be performed on them. Inherit
ance is a process in object oriented programming in which objects acquire
(or inherit) the properties of objects of another class. It is therefore u
sed to create relationships between one object and another. Each class gro
ups together objects of a similar type, with similar properties. New class
es can be formed by this process whose objects will have properties of bot
h the classes from which this new class is formed. A superclass has all of
the properties of the subclasses below it. At the same time subclasses are
each distinctive from each other but related via the superclass. Subclasse
s are said to ‘extend’ superclasses. Due to these relationships, object or
iented programmes tend to be easier to modify since they do not need to be
changed when a new object, with different properties is added. Instead, a
new object is made to inherit properties of objects which already exist. I
nheritance can be divided into two main processes: single inheritance and
multiple inheritance. Single inheritance means that the class can only inh
erit from one other class, whereas multiple inheritance allows for inherit
ance from several classes.\n', 'Inheritance is one of the basic concepts o
f Object Oriented Programming.  It’s objective is to add more detail to pr
e-existing classes whilst still allowing the methods and variables of thes
e classes to be reused.  The easiest way to look at inheritance is as an
“…is a kind of” relationship.  For example, a guitar is a kind of string i
nstrument, electric, acoustic and steel stringed are all types of guita
r.\nThe further down an inheritance tree you get, the more specific the cl
asses become.  An example here would be books.  Books generally fall into
two categories, fiction and non-fiction.  Each of these can then be sub-di
vided into more groups.  Fiction for example can be split into fantasy, ho
rror, romance and many more.  Non-fiction splits the same way into other t
opics such as history, geography, cooking etc.  History of course can be s
ub-divided into time periods like the Romans, the Elizabethans, the World
Wars and so on.\n', 'Inheritance is a method of forming new classes using
predefined classes. The new classes are called derived classes and they in
herit the behaviours and attributes of the base classes. It was intended t
o allow existing code to be used again with minimal or no alteration. It a
lso offers support for representation by categorization in computer langua
ges; this is a powerful mechanism of information processing, vital to huma
n learning by means of generalization and cognitive economy. Inheritance i
s occasionally referred to as generalization due to the fact that is-a rel
```

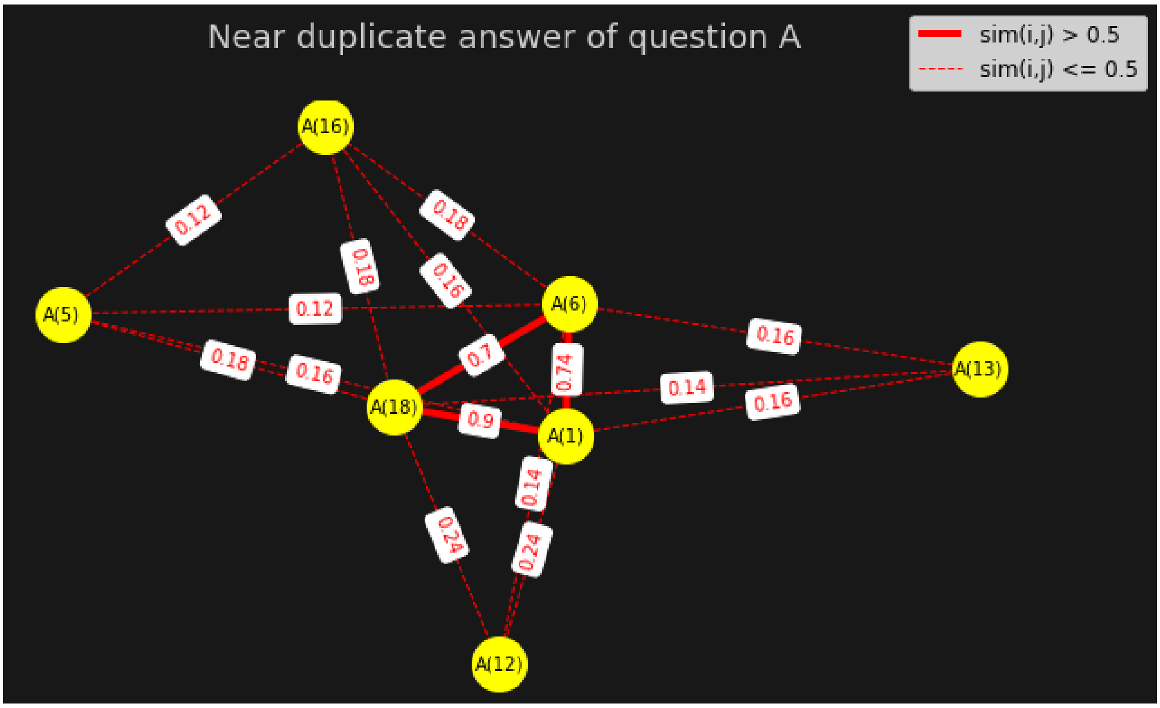
```
ationships represent a hierarchy between classes of objects. Inheritance h
as the advantage of reducing the complexity of a program since modules wit
h very similar interfaces can share lots of code. Due to this, inheritance
has another view called polymorphism, where many sections of code are bein
g controlled by some shared control code. Inheritance is normally achieved
by overriding one or more methods exposed by ancestor, or by creating new
methods on top of those exposed by an ancestor. Inheritance has a variety
of uses. Each different use focuses on different properties, for example t
he external behaviour of objects, internal structure of an object, inherit
ance hierarchy structure, or software engineering properties of inheritanc
e. Occasionally it is advantageous to differentiate between these uses, as
it is not necessarily noticeable from context. \n', 'Inheritance allows pr
ograms developed in an Object Orientated language to reuse code without ha
ving it replicated unnecessarily elsewhere within the program.\n\nTo achie
ve this, the programmer has to note generalisations and similarities about
various aspects of the program. \n\nFor example, a program could exist to
model different forms of transport. At first glance, a car and a train may
not have much in common. But abstractly, both will have a speed at which t
hey are travelling, a direction, and a current position.\nMethods utilisin
g this data can be specified high up in the inheritance hierarchy, for exa
mple in a ‘Transport’ class. For example you could have a method which wor
ks out the new position of a train after travelling x minutes in direction
y. Likewise, you might want to be able to find out the same information fo
r an object of the type car. \nInheritance means that if such a method was
defined in the superclass of the train and car classes, any car or train o
bject can utilise it. \n\nThe train and car subclasses are said to ‘exten
d’ the Transport class, as they will have additional characteristics which
they don’t share. E.g. passenger capacity would be a class variable of bot
h car and train (but have different values), and a train may have methods
along the lines of ‘is toilet engaged’.\nIf you then wanted to add additio
nal forms of transport, such as an aeroplane, you may wish for that also t
o have a ‘toilet engaged’ function. Then you could have an extended hierar
chy, where a Mass Transport class extends the Transport class. Under which
you’d have a train and aeroplane, which would inherit characteristics from
both super classes.', ' Inheritance is an important feature in object orie
ntated programming. This is because it allows new classes to be made that
extend previous classes and to go into more detail. \n\nThis is carried ou
t by allowing the new class to reuse the existing class methods and variab
les, whilst also creating class specific methods and variables. This means
that the new class, the subclass, is a more specialised version of the ori
ginal, or superclass.\n\nBecause of this it means that the subclass can us
e all the public methods and variables from the superclass; however any pr
ivate methods or variables are still private. \n\nAlso it should be noted
that a class can only extend one class, e.g. can only be a subclass to one
superclass. However a superclass can have more then one subclass and a cla
ss can both be a subclass and a superclass. If this occurs then all of the
non-private methods and variables can be used by the most specialised clas
s.\n\nThis means that inheritance is used when types have common factors a
nd these would be put into the superclass. Then the subclass/es then exten
d these to add more detail. An example of this could be using a superclass
of employee and then to have two subclasses called fulltime and part time.
As employee could have name, address and other details whilst full time co
uld just have salary and part time could work out the salary from part tim
e hours worked, as the full time members of staff wouldn’t need these.\n',
"Inheritance is a way to form new classes (instances of which are called o
bjects) using classes that have already been defined. The new classes, kno
wn as derived classes, take over (or inherit) attributes and behavior of t
he pre-existing classes, which are referred to as base classes (or ancesto
r classes). It is intended to help reuse existing code with little or no m
odification.\n\nAn advantage of inheritance is that modules with sufficien
tly similar interfaces can share a lot of code, reducing the complexity of
```

```
the program. Inheritance therefore has another view, a dual, called polymo
rphism, which describes many pieces of code being controlled by shared con
trol code.\n\nInheritance is typically accomplished either by overriding 
(replacing) one or more methods exposed by ancestor, or by adding new meth
ods to those exposed by an ancestor.\n\nIn defining this inheritance hiera
rchy we have already defined certain restrictions, not all of which are de
sirable. Singleness: using single inheritance, a subclass can inherit from
only one superclass. Visibility: whenever client code has access to an obj
ect, it generally has access to all the object's superclass data. Static: 
the inheritance hierarchy of an object is fixed at instantiation when the 
object's type is selected and does not change with time.\n", 'When we talk
about inheritance in object-oriented programming languages, which is a con
cept that was invented in 1967 for Simula, we are usually talking about a 
way to form new classes and classes are instances of which are called obje
cts and involve using classes that have already been defined. \nDerived cl
asses  are intended to help reuse existing code with little or no modifica
tion and  are the new classes that  take over (or inherit) attributes and 
behavior of the pre-existing classes, usually referred to as base classes 
(or ancestor classes). \nCategorization in computer languages is a powerfu
l way number of processing information and inheritance provides the suppor
t for representation by categorization. Furthermore,  it is fundamental fo
r helping humans learn by means of generalization in what is known about s
pecific entities is applied to a wider group given a belongs relation can 
be established and cognitive processing which involves less information be
ing acquired to be stored about each specific entity, but in actual fact o
nly its particularities.\nAn instance of a "fruit" is a generalization of 
"apple", "orange", "mango" and many others. Inheritance can also sometimes
be referred to as generalization, because is-a relationships represent a h
ierarchy amongst classes of objects. It can be considered that fruit is an
abstraction of apple, orange, etc. Conversely, since apples are fruit, the
y may naturally inherit all the properties common to all fruit, such as be
ing a fleshy container for the seed of a plant.\n Modules with sufficientl
y similarities in interfaces would be able to share a lot of code and ther
efore reducing the complexity of the program. This can be known as one of 
the advantages of inheritance. Therefore inheritance can be known to have 
a further view, a dual, which describes many parts of code that are under 
control of shared control code, named as polymorphism. \nOn the other han
d, inheritance is normally accomplished either by replacing one or more me
thods exposed by ancestor, or by adding new methods to those exposed by an
ancestor. A well known term used for this replacing act is called overridi
ng.\n', 'In object-oriented programming, inheritance is the ability to spe
cify one class to be a subclass of another; this leads to a hierarchy of c
lasses, with the child classes inheriting and specialising - and sometimes
adding to - the functionality and data structures of the parent classes. T
he hierarchy that is formed is also useful for the organisation of classes
and objects, as it defines a relationship between the child and the parent
(the child class is a “kind of” the parent class). Inheritance is useful f
or situations where several classes share common features, such as needed 
functions or data variables. In addition to this, child classes can be ref
erenced in terms of their parent classes, which can be useful when storing
large data structures of objects of several classes, which can all be refe
renced as one base class. Inheritance is a core aspect of object-oriented 
programming, and is available in some form or another in most, if not all,
object oriented languages available today. Most of these languages provide
an “extend” keyword, which is used to subclass another. Also, the function
s and data variables that are inherited by the subclasses can be controlle
d through the use of visibility modifiers.', 'Inheritance is a concept in 
Object Oriented programming where a child- or sub-class inherits character
istics from a parent- or super-class.  The concept takes its name from gen
etic inheritance where a child can inherit genetic characteristics from it
s parents.\n\nInheritance, at its simplest, allows programmers to model a 
```

```
relationship where one object is a kind of another.  For instance two clas
ses, one representing an undergraduate student and another representing a
post-graduate student could both be said to belong to a more generalised c
lass representing all students.  Similarly, we could say that dogs and cat
s are two kinds of animal, or that bridges and skyscrapers are two types o
f man-made structure.\n\nSubclasses are said to extend or specialise their
superclasses.  Attributes (variables) and behaviours (functions) that are
common between classes can be included in the definition of the superclas
s, leaving the subclass definitions containing only the attributes and beh
aviours that are unique to that class.\n\nInheritance can be used to creat
e a multiple level architecture of classes.  In such an architecture even
the bottom-most subclasses inherit all of the attributes and behaviours th
at are defined in the very top-most superclasses.  This can save the progr
ammer time because it renders unnecessary a lot of code duplication.', 'In
object-oriented programming, inheritance is a way to form new classes\n(in
stances of which are called objects) using classes that have already\nbeen
defined.\n\nInheritance is also sometimes called generalization, because t
he is-a\nrelationships represent a hierarchy between classes of objects. F
or\ninstance, a "fruit" is a generalization of "apple", "orange", "mang
o"\nand many others. One can consider fruit to be an abstraction of appl
e,\norange, etc. Conversely, since apples are fruit (i.e., an apple is-a\n
fruit), apples may naturally inherit all the properties common to all\nrui
t, such as being a fleshy container for the seed of a plant.\n\nInheritanc
e is typically accomplished either by overriding (replacing)\none or more
methods exposed by ancestor, or by adding new methods to\nthose exposed by
an ancestor.', 'Inheritance is the ability of a subclass to inherit defaul
t, protected and public attributes and methods from its superclasses. Each
object (except java.lang.Object) can be cast to an object of one of its su
perclasses. However an object cannot be cast to a class which is no relati
ve of it. Here is an example of inheritance:\nWe have the class of all liv
ing things which have attributes like weight and age. We have the classes
of animals, plants, viruses and fungi that are subclasses of the class of
all living things. The animals have their unique attributes (organs, hair,
etc.) and methods (walking, mating, etc.). They also inherit the attribute
s and methods of its superclass. Animals can be treated (cast) to living t
hings. However, animals cannot be treated as fungi.\nIn object oriented pr
ogramming inheritance is also dependant on access level modifiers. For exa
mple private attributes and methods cannot be inherited. Virtual attribute
s and methods can be shadowed/overridden. In Java all attributes and metho
ds are implicitly virtual. Object variable can store a reference to the sa
me class or a subclass (i.e. this or more specialised version). However, o
bject variables cannot store references to a superclass (i.e. less special
ised version) of the original class.\n', 'In object-oriented programming,
inheritance is a way to form new classes (instances of which are called ob
jects) using classes that have already been defined. The inheritance conce
pt was invented in 1967 for Simula\nInheritance provides the support for r
epresentation by categorization in computer languages. Categorization is a
powerful mechanism number of information processing, crucial to human lear
ning by means of generalization and cognitive economy (less information ne
eds to be stored about each specific entity, only its particularities).\nT
he new classes, known as derived classes, take over (or inherit) attribute
s and behavior of the pre-existing classes, which are referred to as base
classes (or ancestor classes). It is intended to help reuse existing code
with little or no modification.\nInheritance is also sometimes called gene
ralization, because the is-a relationships represent a hierarchy between c
lasses of objects. For instance, a "fruit" is a generalization of "apple",
"orange", "mango" and many others. One can consider fruit to be an abstrac
tion of apple, orange, etc. Conversely, since apples are fruit (i.e., an a
pple is-a fruit), apples may naturally inherit all the properties common t
o all fruit, such as being a fleshy container for the seed of a plant.\nAn
advantage of inheritance is that modules with sufficiently similar interfa
```

```
ces can share a lot of code, reducing the complexity of the program. Inher
itance therefore has another view, a dual, called polymorphism, which desc
ribes many pieces of code being controlled by shared control code.\nInheri
tance is typically accomplished either by overriding (replacing) one or mo
re methods exposed by ancestor, or by adding new methods to those exposed
by an ancestor.\nComplex inheritance, or inheritance used within a design
that is not sufficiently mature, may lead to the Yo-yo problem.\n', 'The i
dea of inheritance in OOP refers to the formation of new classes with the
already existing classes. The concept of inheritance was basically formula
ted for Simula in 1967.\nAs a result, the newly created inherited or deriv
ed classes inherit the properties and behavior of the classes from which t
hey are derived. These original classes are either called base classes or
sometimes referred to as ancestor classes.\nThe idea of inheritance is to
reuse the existing code with little or no modification at all.\nThe basic
support provided by inheritance is that it represents by categorization in
computer languages. The power mechanism number of information processing t
hat is crucial to human learning by the means of generalization and cognit
ive economy is called categorization. Where generalization if the knowledg
e of specific entities and is applied to a wider group provided that belon
gs relation can be created. On the other hand cognitive economy is where l
ess information needs to be stored about each specific entity except for s
ome particularities.\nThere are examples where we can have modules with si
milar interfaces. The advantage that inheritance provides is that it makes
such modules share a lot of code which consequently reduces the complexity
of the program.\n']

#task:  20
Signatures metric:  20 x 100
#permutations used to create signatures: 100
```



Question B: The result illustrates that there are no pairs of answers which can be identified as near duplicate answers

In [ ]:

```
LSH_task('B')
```

```
['PageRank is a link analysis algorithm used by the Google Internet search
engine that assigns a numerical weighting to each element of a hyperlinked
set of documents, such as the World Wide Web, with the purpose of "measuri
ng" its relative importance within the set. The algorithm may be applied t
o any collection of entities with reciprocal quotations and references. Th
e numerical weight that it assigns to any given element E is also called t
he PageRank of E and denoted by PR(E).\nThe name "PageRank" is a trademark
of Google, and the PageRank process has been patented (U.S. Patent 6,285,9
99 ). However, the patent is assigned to Stanford University and not to Go
ogle. Google has exclusive license rights on the patent from Stanford Univ
ersity. The university received 1.8 million shares in Google in exchange f
or use of the patent; the shares were sold in 2005 for $336 million.\nGoog
le describes PageRank:\n“ \tPageRank relies on the uniquely democratic nat
ure of the web by using its vast link structure as an indicator of an indi
vidual page\'s value. In essence, Google interprets a link from page A to
page B as a vote, by page A, for page B. But, Google looks at more than th
e sheer volume of votes, or links a page receives; it also analyzes the pa
ge that casts the vote. Votes cast by pages that are themselves "importan
t" weigh more heavily and help to make other pages "important". \t”\nIn ot
her words, a PageRank results from a "ballot" among all the other pages on
the World Wide Web about how important a page is. A hyperlink to a page co
unts as a vote of support. The PageRank of a page is defined recursively a
nd depends on the number and PageRank metric of all pages that link to it
("incoming links"). A page that is linked to by many pages with high PageR
ank receives a high rank itself. If there are no links to a web page there
is no support for that page.\nGoogle assigns a numeric weighting from 0-10
for each webpage on the Internet; this PageRank denotes a site’s importanc
e in the eyes of Google. The PageRank is derived from a theoretical probab
ility value on a logarithmic scale like the Richter Scale. The PageRank of
a particular page is roughly based upon the quantity of inbound links as w
ell as the PageRank of the pages providing the links. It is known that oth
er factors, e.g. relevance of search words on the page and actual visits t
o the page reported by the Google toolbar also influence the PageRank. In
order to prevent manipulation, spoofing and Spamdexing, Google provides no
specific details about how other factors influence PageRank.\nNumerous aca
demic papers concerning PageRank have been published since Page and Brin
\'s original paper. In practice, the PageRank concept has proven to be vul
nerable to manipulation, and extensive research has been devoted to identi
fying falsely inflated PageRank and ways to ignore links from documents wi
th falsely inflated PageRank.\nOther link-based ranking algorithms for Web
pages include the HITS algorithm invented by Jon Kleinberg (used by Teoma
and now Ask.com), the IBM CLEVER project, and the TrustRank algorithm.\n',
'PageRankalgorithm is also known as link analysis algorithm.  It has been
used by google. The algorithm may be applied to any collection of entities
with reciprocal quotations and hyperlinked set of documents, such as the W
orld Wide Web, with the purpose of "measuring references. The name "PageRa
nk" is a trademark of Google, and the PageRank process has been patented
(U.S. Patent 6,285,999 ). The numerical weight that it assigns to any give
n element E is also called the PageRank of E and denoted by PR(E).\nThe na
me "PageRank" is a trademark of Google, and the PageRank process has been
patented (U.S. Patent 6,285,999 ). However, the patent is assigned to Stan
ford University and not to Google. Google has exclusive license rights on
the patent from Stanford University. \nIn other words, a PageRank results
from a "ballot" among all the other pages on the World Wide Web about how
important a page is. A hyperlink to a page counts as a vote of support. Th
e PageRank of a page is defined recursively and depends on the number and
PageRank metric of all pages that link to it ("incoming links"). \nNumerou
s academic papers concerning PageRank have been published since Page and B
rin\'s original paper.[4] In practice, the PageRank concept has proven to
be vulnerable to manipulation, and extensive research has been devoted to
identifying falsely inflated PageRank and ways to ignore links from docume
```

```
nts with falsely inflated PageRank\n\t"\n\n', 'PageRank is a link analysis
algorithm used by the Google Internet search engine that assigns a numeric
al weighting to each element of a hyperlinked set of documents, such as th
e World Wide Web, with the purpose of "measuring" its relative importance
within the set. Google assigns a numeric weighting from 0-10 for each webp
age on the Internet; this PageRank? denotes a site's importance in the eye
s of Google.\n\nThe PageRank? is derived from a theoretical probability va
lue on a logarithmic scale like the Richter Scale. The PageRank? of a part
icular page is roughly based upon the quantity of inbound links as well as
the PageRank? of the pages providing the links. The algorithm may be appli
ed to any collection of entities with reciprocal quotations and reference
s. The numerical weight that it assigns to any given element E is also cal
led the PageRank? of E and denoted by PR(E).\n\nIt is known that other fac
tors, e.g. relevance of search words on the page and actual visits to the
page reported by the Google toolbar also influence the PageRank?. Other li
nk-based ranking algorithms for Web pages include the HITS algorithm inven
ted by Jon Kleinberg (used by Teoma and now Ask.com), the IBM CLEVER proje
ct, and the TrustRank? algorithm. \n', "PageRank (PR) refers to both the c
oncept and the Google system used\nfor ranking the importance of pages on
the web. The "PageRank" of a\nsite refers to its importance or value on th
e web in relation to the\nrest of the sites that have been "PageRank"ed.\n
\nThe algorithm basically works like a popularity contest – if your site\n
is linked to by popular websites, then your site is considered more\npopul
ar. However, the PR doesn't just apply to the website as a whole\n– differ
ent pages within a website get given different PRs dependent\non a number
of factors:\n\n* Inbound links (backlinks) – how many pages (other than th
e ones on your website) link to this particular page\n\n* Outbound links
(forward links) – how many external pages the particular page links to\n\n
* Dangling links – how many pages with no external links are linked to fro
m a particular page\n\n* Deep links – how many links that are not the home
page are linked to from a particular page\n\nPR tries to emulate a "random
surfer". The algorithm includes a\ndampening factor, which is the probabil
ity that a random surfer will\nget bored and go and visit a new page - by
default, this is 0.85. A\nvariation on this is the "intentional surfer", w
here the importance of\na page is based on the actual visits to sites by u
sers. This method is\nused in the Google Toolbar, which reports back actua
l site visits to\nGoogle.\n", 'There are many attributes which infulance t
he ranking of a page in google, The main too are the content, key words, a
nd links. The content of a webpage generaly gives a good idea about what t
he page is about, however, there are some flaws in this, for example, for
along time ibm web page didnt contain the word computer dispite it being s
trongly associated with them. To solve this problem, web pages can assign
itself key words, which contribute to its ranking in searches.\n\nThe seco
nd method is the use of links. the more sights which links to your web pag
e and the higher the rank of those sights, the higher the rank of your sit
e will be. This method is used as links are seen as an adoursment of a sig
ht.\n\nWith both these methods of ranking web pages, there are issues. key
words can be compromised by sparming, google solves this problem by penoli
zing such activity. Useing links to rank a page also has its problems, for
example, link farms which have recursive links, for the sole perpos of rai
sing there ranking, google takels this by useing a dampaning algorthem.
\n', 'PageRank algorithm is patented by Stanford University. It is a link
analysis algorithm employed by the Google Internet search engine that assi
gns a value used to measure the importance to each element of a hyperlinke
d set of documents, such as the WWW, with the purpose of " measuring" its
relative significance within the set.\n\nGoogle owns exclusive license rig
hts on the patent from Stanford University. The University received 1.8 mi
llion shares in Google in return for use of the patent. \n', 'PageRank is
a link analysis algorithm used by the Google Internet search engine that a
ssigns a numerical weighting to each element of a hyperlinked set of docum
ents, such as the World Wide Web, with the purpose of "measuring" its rela
```

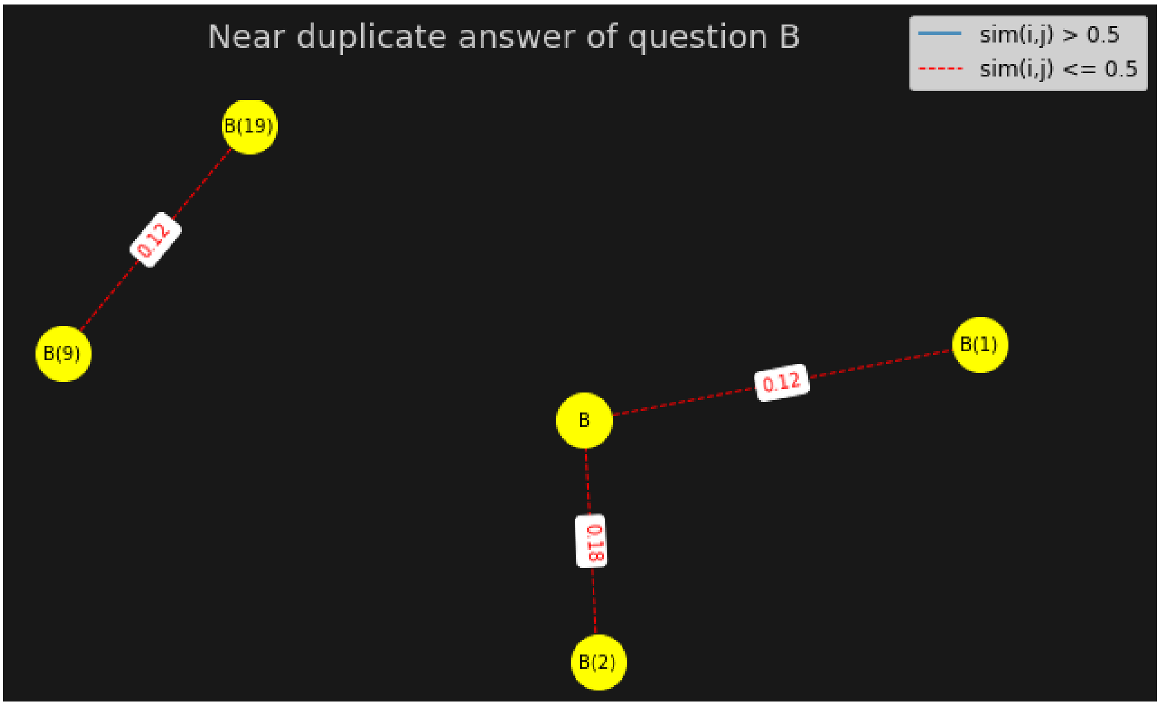
```
tive importance within the set. The algorithm may be applied to any collec
tion of entities with reciprocal quotations and references. PageRank Uses
in google toolbar: Measures popularity of a site ,Marketing value,Updated
periodically, in google directory: PageRank: sort links within categories;
Volunteers evaluate, classify, annotate;Open Directory project using PageR
ank. \n', 'The PageRank algorithm is used to designate every aspect of a s
et of hyperlinked documents with a numerical weighting. It is used by the
Google search engine to estimate the relative importance of a web page acc
ording to this weighting. The system uses probability distribution to dete
rmine the odds that a person randomly clicking on links will arrive at any
given page. Following this, each web page is given a ranking of 0-10 accor
ding to its relevance to a search. The PageRank is calculated by taking in
to consideration the number of inbound links, and the PageRank of the page
s supplying these links. This means therefore that if a webpage is linked
to others that have a high ranking, then it too will receive a high rank.
\n\nDue to the nature of the PageRank system, it is susceptible to manipul
ation and has been exploited so that certain pages are given a false, exag
gerated ranking. In these cases, only Goggle has access to the genuine Pag
eRank. However, much research has been conducted into methods of avoiding
links from documents with a false PageRank to try and iron out the bugs in
this system and from 2007 Google has actively penalized schemes which try
to increase rankings artificially.\n', 'A website’s page rank, is how ‘imp
ortant’ it is on the web.  It is essentially a popularity meter.  Populari
ty or importance is determined by the amount of links relating to the page
there are, there are four different types.  Inbound, links from other page
s to yours.  Outbound, links from your page to others.  Dangling, links to
a page which has no links to others.  Deep, links to a specific page, usua
lly bypassing the homepage.  The page rank algorithm takes the probability
of a ‘random surfer’ becoming bored and requesting another ‘random page’
(otherwise known as the dampening factor) away from 1 and divides this num
ber by the number of pages in the system, adding it to the dampening facto
r multiplied by the page rank of a linked page divided by the number of ou
tbound links on that linked page.  Adding on this last section for every o
ther page linked to from the original page.  Google uses this algorithm to
assist intentional surfers in finding the best websites to suit their need
s.  One of the problems with this popularity algorithm is that it is easil
y manipulated and can give false values, hence the frequent recalculating
of page ranks.\n', 'PageRank is a link analysis algorithm used by the Goog
le Internet search engine that assigns a numerical weighting to each eleme
nt of a hyperlinked set of documents, such as the World Wide Web, with the
purpose of "measuring" its relative importance within the set. Google assi
gns a numeric weighting from 0-10 for each webpage on the Internet; this P
ageRank denotes a site’s importance in the eyes of Google. The PageRank is
derived from a theoretical probability value on a logarithmic scale like t
he Richter Scale. PageRank is a probability distribution used to represent
the likelihood that a person randomly clicking on links will arrive at any
particular page. PageRank can be calculated for collections of documents o
f any size. It is assumed in several research papers that the distribution
is evenly divided between all documents in the collection at the beginning
of the computational process. The PageRank computations require several pa
sses, called "iterations", through the collection to adjust approximate Pa
geRank values to more closely reflect the theoretical true value. A probab
ility is expressed as a numeric value between 0 and 1. A 0.5 probability i
s commonly expressed as a "50% chance" of something happening. Hence, a Pa
geRank of 0.5 means there is a 50% chance that a person clicking on a rand
om link will be directed to the document with the 0.5 PageRank. The PageRa
nk theory holds that even an imaginary surfer who is randomly clicking on
links will eventually stop clicking. The probability, at any step, that th
e person will continue is a damping factor d. Various studies have tested
different damping factors, but it is generally assumed that the damping fa
ctor will be set around 0.85. \n', 'The  algorithm that Google uses to ass
```

```
ign a weighting to each element of a linked set of documents, with the pur
pose of "measuring" its relative importance within the set. \nA particular
websites PageRank results from a "vote" from other pages on the Internet a
bout how important that website actually is. A link to a page is seen as a
vote of support. The PageRank depends on the PageRank rating and number of
all pages that have links to it. Additionally, if a page is linked to by p
ages with a high PageRank rating, this increases the rating of the origina
l page.\nThe PageRank scale ranges from 0-10. The rating of a certain page
is generally based upon the quantity of inbound links as well as the perce
ived quality of the pages providing the links. \nPageRank could be describ
ed as a probability distribution representing the chance that someone rand
omly clicking on links will reach a certain page. The PageRank calculation
s require iterations through the collection of web pages to alter approxim
ate PageRank values to accurately reflect the actual rank.\nIn order to pr
event spamming, Google releases little information on the way in which a P
ageRank is calculated. The PageRank algorithm has led to many sites being
spammed with links in an attempt to artificially inflate the PageRank of t
he linked page, notably in blog comments and message boards. In 2005 a ‘no
follow’ tag was added as an attribute of a HTML link to be used where Goog
le shouldn’t change the PageRank of the linked page as a result of the lin
k. ', 'The first thing to consider when talking about Google’s PageRank al
gorithm, is that a PageRank is essentially how important that web page is
to the internet. So in essence it is a popularity contest between WebPage
s.\n\nOriginally search engines used highest keyword density, however this
could be abused if keyword spamming was implemented. Instead Google uses a
system that is based on sites linking to each other, e.g. the more importa
nt a site is that is linked to yours the higher your site will become. \n
\nThe algorithm Google actually users is based on 4 factors, total number
of pages, dampening factor, PageRank of a single page and the number of ou
tbound links. A dampening factor is used to counter random surfers, who ge
t bored and then switch to other pages. This formula is then re-used until
the results seem to converge together, to find the PageRank, so it is calc
ulated iteratively.\n\nPageRank is used by Google to measure a popularity
of the site and a number between 0-10 is assigned to each webpage dependin
g on their PageRank. This allows Google to calculate a marketing value for
different WebPages.\n\nAlso it should be noted that the PageRank is period
ically updated every 3 to 6 months, this is counter hackers influence on d
ifferent PageRanks.\n', "The PageRank is a recursive algorithm used by Goo
gle to determine which webpages are more important than others. The algori
thm considers the importance of a webpage to be reflected by how many othe
r webpages link to that page, and the importance of those pages.\n\nFor ea
ch page that links to a page A, the PageRank between zero and one is calcu
lated iteratively according to the following two key factors: The probabil
ity of a user navigating away from a page randomly; the PageRank of any pa
ge that links to A, divided by the total number of outbound links from tha
t page. This assumes that a link among many outbound links is less valuabl
e than a link among fewer outbound links. A variation of the PageRank meth
od bases the importance of a webpage on how many visits the page gets. \n
\nThe method can be abused when people deliberately link to sites in order
to raise a site's PageRank. However, it is still a good indicator for sear
ch engines to use as a variable in deciding on the most appropriate result
s according to a query. \n", 'PageRank is a link analysis algorithm that i
s used by search engine such as Google Internet that assigns a numerical w
eighting to every element of a hyperlinked set of documents, like the Worl
d Wide Web, with the hope of "measuring" the relative importance held in t
he set. The algorithm may be applied to any numbr of entities with recipro
cal quotations and references. The weight taking a numerical value which a
ssigns to any given element E is also known as the PageRank of E and is de
noted by PR(E).\nA trademark of Google has the name "PageRank"  and this p
rocess has been patented (U.S. Patent 6,285,999\xa0). Nevertheless, the pa
tent is assigned to the University of Stanford and not to Google. Google h
```

```
as exclusive license rights on the patent from the University of Stanford
and the university received 1.8 million shares in Google in exchange for u
se of the patent; the in the year 2005, shares were sold for $336 millio
n.\n', "The Google search engine uses a link analysis algorithm called Pag
eRank to assign a relative numerical importance to a set of hyperlinked do
cuments, such as the World Wide Web.\nFor a given page, it's importance (t
he PageRank value) results from a ballot among all the other pages in the
set. For a page to give a vote to another, it must link to it, and so the
PageRank depends on the number of incoming links, anf the PageRank of thos
e pages that provide the links. Pages that are linked to by many high rank
ing pages will themselves obtain a high rank. If a page has no incoming li
nks, there is no support for that page.\nThe PageRank is a numeric weighti
ng of 0 to 10, and denotes how important a site is in Google's eyes. Like
the Richter Scale, the PageRank is a value on a logerithmic scale that is
derived from a probability. In addition to the quantity and quality of inb
ound links,other factors affect the PageRank, such as the number of visits
to the page and the search words that are used on the page.\nTo prevent si
tes from manipulating or spoofing PageRank, very little details are provid
ed by Google as to what factors actually affect it.", 'PageRank is an algo
rithm that was developed by Google to provide the most relevant search res
ults to its users queries.  PageRank, along with similar algorithms develo
ped by Google’s competitors for their search engines, is part of the secon
d generation of technologies designed to rate the importance of web pages:
the first, which was solely based on keywords in the page content and meta
-data, could easily be influenced by those wishing to obtain a higher rank
ing for their less-relevant pages.\n\nThe different with PageRank is that
it tries to determine a web page’s relevance to users by attempting to det
ermine its importance.  It does this by assigning it a value of importance
that is dependant upon the number of web sites that link to that page, tak
ing into account the importance value, or PageRank, of those pages.  The P
ageRank is computed iteratively, and it is found that the PageRank values
converge fairly rapidly.\n\nAlthough it is much better than simple keyword
-based ranking algorithms, PageRank is not infallible: we have an internet
where advertising revenue can make up most - and quite frequently all – of
a web site’s income and the people that run these web sites will always be
trying to trick the system into giving their pages a higher PageRank.   On
e of Google’s attempts to counter this is their Google Toolbar browser plu
gin.\n\nGoogle Toolbar is a free tool which provides a number of useful fu
nctions in a convenient location: the users web browser window.  Google’s
payoff is that it gets to track the behaviour of actual users.  This allow
s them to see whether their PageRank algorithm is accurate in assigning hi
gh PageRank values to the most relevant web pages and, just as importantl
y, low values to those that are irrelevant and try to fool the system.
\n', 'The PageRank algorithm used by google harnesses the implicit collect
ive\nintelligence present in the structure of the world wide web. Any page
on\nthe Internet will generally link to at least one other, by modelling t
his\nlink structure as a graph, we can build up a symbolic representation
of\nthe world wide web.\n\nAs the basic level, the nodes with the highest
degrees can be considered\nthe most "popular" and by inference the most im
portant - which can be used\nto rank the pages when returning search resul
ts.\n\nExpanding on this theory, we can then say that the links from an im
portant\npages are themselves more important. Using this idea we can adjus
t the\nrankings of our pages so that pages linked to be the most important
pages\nare considered more relevant.\n\nThe actual Google PageRank algorit
hm is much more complex than this, but\nfollows the same underlying princi
ples. It incorporates some more advanced\nreasoning to avoid website creat
ors exploiting their knowledge of the algorithm\nto artificially increase
their PageRank through use of web-rings and other\nsimilar reciprocal hype
rlinking schemes.', 'Page rank algorithm is used to determine a webpage’s
importance or relevance in the web dependant on certain criteria. The crit
eria may include numbers of word matches with the search terms, number of
```

```
other webpages that link this one and/or cite it as a source, number of un
ique visits for certain amount of time etc. There are some techniques that
try to fool the search engines like link farms, keyword spamming and a lot
of meta tags. The last two are somewhat easier to be dealt with (simply by
being ignored most of the time). Link farms are groups of sites that are p
roducing links between each other pursuing higher link counts. The reason
for such manipulations is the pursuit of higher page rank so even higher n
umber of users will see the page which will lead to higher income. Link fa
rms can be exploited by joining to them and get inbound linkage but refuse
to add links for one's own site to the sites from the link farm. Google's
toolbar tries to follow the intentional user model by counting the visits
from actual users (i.e. not computer bots) to a website. Page ranks can be
calculated either recursively or iteratively. One of the most important us
es of page rank is its meaning to advertising.\n', 'Since the develop of t
he Web 2.0, Google as one of the most popular search engine in the world,
there are many algorithms in the web search. Accordingly, implementations
of link analysis algorithms will typical discount such "internal" links. T
he word computer can be exploited by web search engines such as Google. Th
us, the web is just like a graph, and the PageRank, which is our first tec
hnique for analysing the link which is  assigns to every node in\nthe web
graph a numerical score between 0 and 1. Since the PageRank is the most im
portant algorithms which is used in the Google engine. For example, there
are four pages group: A, B, C and D. If every page link to A, then  A's Pa
geRank value shoule be the total value of  B, C and D .\nPR(A) = PR(B) + P
R(C) + PR(D)\nMoreover, there is a q = 0.15 which is be use in the web pag
e, like the general algorithm below:\n \nHowever, the disadvantage is of P
ageRank algorithm is that the renew system is too slow.\n', 'PageRank is a
probability distribution used to represent the likelihood that a person ra
ndomly clicking on links will arrive at any particular page. . It is assum
ed in several research papers that the distribution is evenly divided betw
een all documents in the collection at the beginning of the computational
process. PageRank can be calculated for collections of documents of any si
ze The PageRank computations require several passes, called "iterations",
through the collection to adjust approximate PageRank values to more close
ly reflect the theoretical true value.\nA probability is expressed as a nu
meric value between 0 and 1. A 0. 5 probability is commonly expressed as a
"50% chance" of something happening. Hence, a PageRank of 0.5 means there
is a 50% chance that a person clicking on a random link will be directed t
o the document with the 0.5 PageRank.\nSimplified algorithm\nHow PageRank
Works\nAssume a small universe of four web pages: A, B, C and D. The initi
al approximation of PageRank would be evenly divided between these four do
cuments. Hence, each document would begin with an estimated PageRank of 0.
25.\nIn the original form of PageRank initial values were simply 1. This m
eant that the sum of all pages was the total number of pages on the web. L
ater versions of PageRank (see the below formulas) would assume a probabil
ity distribution between 0 and 1. Here we\'re going to simply use a probab
ility distribution hence the initial value of 0.25.\n']

#task:  20
Signatures metric:  20 x 100
#permutations used to create signatures: 100
```



Question C: The result illustrates that there are no pairs of answers which can be identified as near duplicate answers

In [ ]:

```
LSH_task('C')
```

```
['Vector space model (or term vector model) is an algebraic model for repr
esenting text documents (and any objects, in general) as vectors of identi
fiers, such as, for example, index terms. It is used in information filter
ing, information retrieval, indexing and relevancy rankings. Its first use
was in the SMART Information Retrieval System.\nA document is represented
as a vector. Each dimension corresponds to a separate term. If a term occu
rs in the document, its value in the vector is non-zero. Several different
ways of computing these values, also known as (term) weights, have been de
veloped. One of the best known schemes is tf-idf weighting (see the exampl
e below).\nThe definition of term depends on the application. Typically te
rms are single words, keywords, or longer phrases. If the words are chosen
to be the terms, the dimensionality of the vector is the number of words i
n the vocabulary (the number of distinct words occurring in the corpus).\n
The vector space model has the following limitations:\n   1. Long document
s are poorly represented because they have poor similarity values (a small
scalar product and a large dimensionality)\n   2. Search keywords must pre
cisely match document terms; word substrings might result in a "false posi
tive match"\n   3. Semantic sensitivity; documents with similar context bu
t different term vocabulary won\'t be associated, resulting in a "false ne
gative match".\n   4. The order in which the terms appear in the document
is lost in the vector space representation.\n', 'The definition of term de
pends on the application. Typically terms are single words, keywords, or l
onger phrases. If the words are chosen to be the terms, the dimensionality
of the vector is the number of words in the vocabulary A document is repre
sented as a vector. Each dimensions corresponds to a separate terms. If a
term occurs in the document, its value in the vector is non-zero.\nRelevan
cy rankings of documents in a keyword search can be calculated, using the
assumptions of document similarities theory, by comparing the deviation of
angles between each document vector and the original query vector where th
e query is represented as same kind of vector as the documents.\nLIMITATIO
N:\nThere is some limitation of vector space model.\nModels based on and e
xtending the vector space model include:\n•\tGeneralized vector space mode
l.\n•\t(enhanced) Topic-based Vector Space Model [1] (eTVSM) — Extends the
vector space model by removing the constraint that the term-vectors be ort
hogonal. In contrast to the generalized vector space model the (enhanced)
Topic-based Vector Space Model does not depend on concurrence-based simila
rities between terms. The enhancement of the enhanced Topic-based Vector S
pace Model  (compared to the not enhanced one) is a proposal on how to der
ive term-vectors from an Ontology. \n \n\n', 'The vector space model (also
called, term vector model) is an algebraic model used to represent text do
cuments, as well as any objects in general, as vectors of identifiers. It
is used in information retrieval and was first used in the SMART Informati
on Retrieval System.\n\nA document is represented as a vector and each dim
ension corresponds to a separate term. If a term appears in the document t
hen its value in the vector is non-zero. Many different ways of calculatin
g these values, also known as (term) weights, have been developed. One of
the best known methods is called tf-idf weighting.\n\nThe definition of te
rm depends on the application but generally terms are single words, keywor
ds, or longer phrases. If the words are chosen to be the terms, the dimens
ionality of the vector is the number of words in the vocabulary, which is
the number of distinct words occurring in the corpus.\n\nThe vector space
model has several disadvantages. Firstly, long documents are represented b
adly because they have poor similarity values. Secondly, search keywords m
ust accurately match document terms and substrings of words might result i
n a "false-positive match". Thirdly, documents with similar context but di
fferent term vocabulary will not be associated, resulting in a "false-nega
tive match". Finally, the order in which the terms appear in the document
is lost in the vector space representation. \n', 'Vector space model is an
algebraic model for representing text documents (and in general, any objec
ts) as vectors of identifiers, such as, for example, index terms. Its firs
t use was in the SMART Information Retrieval System. It is used in informa
```

```
tion filtering, information retrieval, indexing and relevancy rankings.\n
\nA document is represented as a vector, and each dimension corresponds to
a separate term. If a term occurs in the document, its value in the vector
is non-zero. Several different ways of computing these values, also known
as (term) weights, have been developed. The definition of term depends on
the application. Typically terms are single words, keywords, or longer phr
ases. If the words are chosen to be the terms, the dimensionality of the v
ector is the number of words in the vocabulary (the number of distinct wor
ds occurring in the corpus).\n\nOne of the best known schemes is tf-idf we
ighting, proposed by Salton, Wong and Yang. In the classic vector space mo
del, the term specific weights in the document vectors are products of loc
al and global parameters.\n\nRelevancy rankings of documents in a keyword
search can be calculated, using the assumptions of document similarities t
heory, by comparing the deviation of angles between each document vector a
nd the original query vector where the query is represented as same kind o
f vector as the documents.\n\nThe vector space model has the following lim
itations:\n\n    * Search keywords must precisely match document terms; wo
rd substrings might result in a "false positive match";\n    * Semantic se
nsitivity; documents with similar context but different term vocabulary wo
n\'t be associated, resulting in a "false negative match";\n    * The orde
r in which the terms appear in the document is lost in the vector space re
presentation;\n    * Long documents are poorly represented because they ha
ve poor similarity values (a small scalar product and a large dimensionali
ty).\n', 'The vector space model is where each document is viewed as a bag
of words, where there order has little significance. Each document is a ve
ctor where each word is a dimension. The vector is then constucted of the
frequency of eacher word (dimension). The draw back to this approach is th
at the length of the document as an inpact on the vector, to compensate fo
r this you can comput the cosine similarity between your two comparism doc
uments. This will find the difference between the two vectors (the dot pro
duct), ignoreing the size of them.\n\nInorder to query the search space, t
he query can also be represented as a vector, then you find the document w
hos vector has the greatest cosine similarities to your query. There are a
number of wighting sceems which can be incoperated inorder to increase the
accuracy of the vextors.\n\nThere are some drawbacks with this approach, C
omputing the cosine similarities between each vector can be expensive as t
he number of dimensions can be in the thousands, To tackle this problem yo
u can use inverted indexs and then a series heuristics inorder to inprove
on this.\nto top\n', 'An algebraic model for representing text documents a
nd any objects in general is known by the name Vector space model. It repr
esents these as vectors of identifiers, index terms are one illustration o
f these. The Vector Space model was first used in the SMART Information Re
trieval System, and it is utilised variously in indexing, information filt
ering, indexing and information retrieval.\n\nA document has representatio
n as a vector. Every dimension is precisely related to a separate term. Th
e way in which term is defined depends entirely on the application: typica
lly ‘terms’ are either single words, keywords or longer phrases. The dimen
sionality of the vector is the number of words in the vocabulary, if it is
the words that are chose to be the terms. So the same rule applies with ke
ywords and indeed longer phrases.\n\nIf a term occurs in the document, its
value in the vector is non-zero. Several different ways of computing these
values, additionally known as (term) weights, have been developed. One of
the most famous schemes is tf-idf weighting. \n', 'The representation of a
set of documents as vectors in a common vector space is known as the vecto
r space vector space model and is fundamental to a host of information ret
rieval (IR) operations including scoring documents on a query, document cl
assification, and document clustering.We first develop the basic ideas und
erlying vector space scoring; a pivotal step in this development is the vi
ew of queries as vectors in the same vector space as the document collecti
on. \n', 'The vector space model is an algebraic model used to represent t
ext documents (and any objects, generally) as vectors of identifiers, for
```

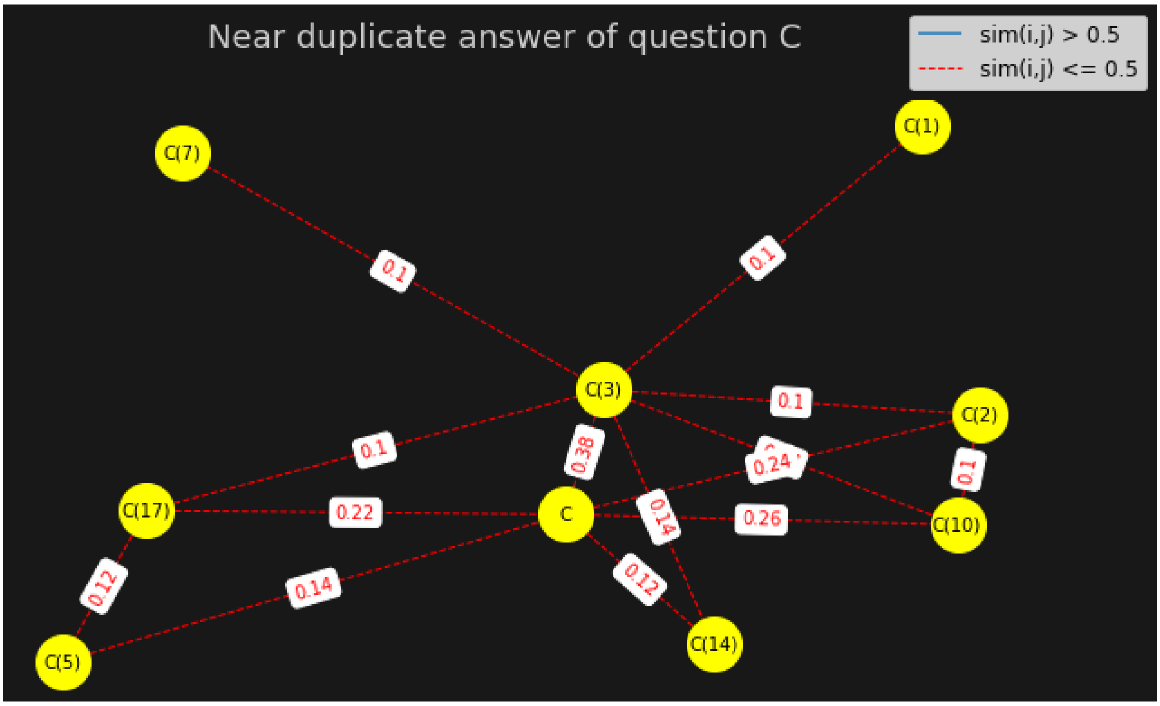
```
instance index terms. Its applications include information filtering, info
rmation retrieval, indexing and relevancy rankings. With reference to this
model, documents are represented as vectors. Each dimension corresponds to
a separate term. The value of a vector is non-zero if a term occurs in the
document. Several different ways have been developed of calculating these
values (also known as term weights). One of the best known schemes is tf-i
df (term frequency-inverse document frequency) weighting. \n\nThe model ca
n be used to determine the relevancy rankings of documents in a keyword se
arch, using the assumptions of document similarities theory, by comparing
the original query vector (where the query is represented as same kind of
vector as the documents) and the deviation of angles between each document
vector.\n\nThe classic vector space model was put forward by Salton, Wong
and Yang and is known as term frequency-inverse document frequency model.
In this classic model the term specific weights in the document vectors ar
e products of local and global parameters. In a simpler Term Count Model t
he term specific weights are just the counts of term occurrences and there
fore do not include the global parameter. \n\n', 'The algebraic model for
representing text documents and objects as vectors of identifiers is calle
d the vector space model.  It is used in information filtering, indexing,
relevancy rankings and information retrieval. It was first used in the SMA
RT Information Retrieval System.\n When a document is represented as a vec
tor, each dimension corresponds to a separate term. A term which occurs in
the document has a value in the vector of non-zero. Other ways of computin
g these values, or weights, have been developed. The most popular is tf-id
f weighting.\nDepending on the application, the definition of term varies.
Single words, keywords and occasionally longer phrases are used for terms.
The dimensionality of the vector, if words are used as terms, is the total
number of words available for use.  By using the assumptions of the docume
nt similarities theory, the relevancy rankings of documents in a keyword s
earch can be worked out by comparing the deviation of angles between vecto
rs both within the document and the original query where the vectors of bo
th are the same type.\nThe limitations of the vector space model are thus.
Due to poor similarity values long documents are poorly represented.  Fals
e positive matches may be returned if search keywords do not precisely mat
ch document terms.  False negative matches could be returned when document
s share a context but have different term vocabulary.  Vector space repres
entation results in the loss of the order which the terms are in the docum
ent.\n', 'Within Information Retrieval each document in a set can be repre
sented as a point in high-dimensional vector space, this representation is
called the vector space model. Information Retrieval queries are also repr
esented as vectors in the same vector space; these are then used in conjun
ction with the document vectors to find relevant documents. The two vector
s are compared and the documents with a higher document-query similarity a
re ranked higher in terms of relevance. There are a variety of techniques
that can be used to compare the two vectors; the most frequently used meth
od for the vector space model is the Cosine Coefficient, which calculates
the angle between the two vectors and produces a value between 0 and 1.
\n', 'A Vector space model (or term vector model) is an algebraic way of r
epresenting text documents (and any objects, in general) as vectors of ide
ntifiers, such as index terms. It is used in information filtering, inform
ation retrieval, indexing and relevancy rankings. Its first application wa
s in the SMART Information Retrieval System.\nA document can be represente
d as a vector. Every dimension relates to a different term. If a term appe
ars in the document, the terms value in the vector is non-zero. Many diffe
rent methods of calculating these values, sometimes known as (term) weight
s, have been developed. tf-idf weighting is one of the most well known sch
emes. (see below example).\nThe definition of a term depends on the applic
ation. Normally a term is a single word, keyword, or a longer phrase. If t
he words are chosen to be the terms, the dimensionality of the vector is t
he number of words in the vocabulary (the number of distinct words occurri
ng in the corpus).\nThe vector space model has some limitations:\n1.\tLong
```

```
er documents are represented poorly because the documents have poor simila
rity values (namely a small scalar product and a large dimensionality)\n
2.\tSearch keywords have to precisely match document terms; word substring
s could potentially result in a "false positive match"\n3.\tSemantic sensi
tivity; documents with a similar context, but different term vocabulary wo
n\'t be associated, resulting in a "false negative match".\n4.\tThe order
in which terms appear in the document is lost in a vector space representa
tion.\n', 'A Vector space model is an algebraic model for representing tex
t documents as vectors of identifiers. A possible use for a vector space m
odel is for retrieval and filtering of information. Other possible uses fo
r vector space models are indexing and also to rank the relevancy of diffe
ring documents.\nTo explain further vector space models, basically a docum
ent is characterized by a vector. With each separate term corresponding to
the differing dimensions. There has been multiple ways of trying to comput
e the different possible values for vector space models with the most reco
gnised being the tf-idf weighting.\nThe differing application has a direct
influence on what the definition of the term means. A normal term is usual
ly a single word, keywords or longer phrases. The number of unique words i
n the vocabulary denotes the dimensionality, if words are used for the ter
ms.\nHowever whilst vector space modelling is useful there are 4 key probl
ems with using it, they are; that the order of the terms are lost, keyword
s must be precise if searched for, bigger documents have a poor similarity
value due to being poorly represented and two documents based on the same
topic won’t be associated if term vocabulary differs.\n', 'In the vector s
pace model (VSM), documents take the form of "bags of words" - a standard
information retrieval approach which represents documents as in a mathemat
ical "bag" structure, recording what terms are present and how often they
occur. \n\nThe vector space model is used in information retrieval to dete
rmine how similar documents are to one another, and how similar documents
are to a search query. \n\nIn a collection of documents, each document can
be viewed as a vector of n values (the terms in the document), where each
term is an axis. Queries can also be represented as vectors on this vector
space model, and so deciding which document matches the query the closest
becomes a matter of selecting the document vector which is nearest to the
query vector. \n\nThe query vector is compared to each document vector in
turn using a "vector similarity measure", which is the cosine of the angle
between the query vector and the document vector. \n\nThis equation is cal
culated by dividing the dot product of the query vector and the document v
ector by the modulus of the query vector multiplied by the modulus of the
document vector. The denominator takes into account differences in the len
gth of the vector, and has the effect of "normalising" the length. Whichev
er document returns the highest cosine similarity score is considered to b
e the closest matching document to the query. \n', 'nformation retrieval
(IR) is the science of searching for documents, for information within doc
uments and for metadata about documents, as well as that of searching rela
tional databases and the World Wide Web. IR is interdisciplinary, based on
computer science, mathematics, library science, information science, infor
mation architecture, cognitive psychology, linguistics, statistics and phy
sics. There is overlap in the usage of the terms data retrieval, document
retrieval, information retrieval, and text retrieval, but each also has it
s own body of literature, theory, praxis and technologies. \nAutomated inf
ormation retrieval systems are used to reduce what has been called "inform
ation overload". Many universities and public libraries use IR systems to
provide access to books, journals and other documents. \n', 'Vector space
model, or term vector model as it is also known, is an algebraic model for
representing objects (although it is mainly used for text documents) as ve
ctors of identifiers; for example, index terms. It is used in information
retrieval and filtering, indexing and relevancy rankings, and was first us
ed in the SMART Information Retrieval System.\n\nA document is represented
as a vector, with each dimension corresponding to a separate term. If a te
rm occurs in the document, the value will be non-zero in the vector. Many
```

```
different ways of computing these values (aka (term) weights) have been de
veloped; one of the best known schemes is tf-idf weighting.\n\nThe way tha
t a \'term\' is defined depends on the application. Typically, terms are s
ingle words, keywords, or sometimes even longer phrases. If the words are
chosen as the terms, the number of dimensions in the vector is the number
of distinct words in the corpus.\n\nRelevancy ranks for documents, in a ke
yword search, can be calculated; this uses the assumptions of document sim
ilarities theory, by comparing the difference of angles between each docum
ent vector and the original query vector, where the query is represented a
s same format vector as the documents.\n\nGenerally, it is easier to calcu
late the cosine of the angle between the vectors instead of the angle itse
lf. A zero value for the cosine indicates that the query and document vect
or are orthogonal and so had no match; this means the query term did not e
xist in the document being considered.\n\nHowever, the vector space model
has limitations. Long documents are poorly represented due to their poor s
imilarity values (a small scalar product and a large dimensionality); sear
ch keywords must match precisely the document terms; word substrings might
result in a "false positive match"; similar context documents but differen
t term vocabulary won\'t be associated, leading to a "false negative matc
h"; and the order that the terms appear in the document is not represented
in the vector space model.\n', 'There are a large number of models used in
solving the problem of Information Retrieval and they are all based on one
of three mathematical bases: set theory, algebra and probabilistic.  The v
ector space model is one of these methods, and it is an algebraic model.\n
\nIn the vector space model a document is represented as a vector.  Within
this vector, each dimension corresponds to a separate term (where a term i
s typically a single word, keyword or phrase.)  If the term doesn't occur
within the document, the value in the vector is zero.  If a term occurs in
the document, its value is non-zero.\n\nTo calculate how relevant each doc
ument is in a keyword search the cosine value of the angle between the vec
tors is easier to calculate instead of the actual angle.\n\nThe vector spa
ce model, however, is not without its limitations: they have small similar
ity values, long documents are poorly represented; the order of words does
not matter; false positive matches may be brought about by terms contained
within words themselves; and documents that should match but use different
semantics may return false negative matches.   There are a number of other
models that are based on or extend the vector space model, and these are d
esigned to try to eradicate these problems.\n', 'Using the vector space mo
del for Information Retrieval models all pages\nand queries as high-dimens
ional sparse vectors. Each item in the vector\nrepresents a different keyw
ord.\n\nThe similiarity betweeen two pages or a query and a page can be co
mputed\nusing the dot product formula to find the cosine between them. Thi
s\nrepresents the angle between them, but in n-dimensional space. Results
\nwill range from -1 to 1, with 1 being a close match. Normally the vector
s\nwill not have any negative values, so results will always be greater th
an\nor equal to 0. The cosine is computed using: cos A = (|a||b|)/(a.b)',
'The vector space model (or term vector model) is an algebraic model for r
epresenting text documents (and any objects, in general) as vectors of ide
ntifiers, such as index terms. It is used in information filtering, inform
ation retrieval, indexing and relevancy rankings. It was used in the first
time in the SMART Information Retrieval System.\n\nA document is represent
ed as a vector. Each and every dimension corresponds to a separate term. I
f a term exists in a document, its value in the vector is not equal to zer
o. A couple of different algorithms of computing these values, also known
as (term) weights, have been created. One of the most popular schemes is t
f-idf weighting.\n\nThe definition of term is dependent on the applicatio
n. Typically terms are keywords, single words or longer phrases. Provided
that words are selected to be the terms, the dimensionality of the vector
is equal to the number of words in the vocabulary.\n\n\nIt is easiest to c
alculate the cosinus of the angle between the vectors instead of the angle
by the formula:\n\n    cos(theta)=v1.v2/(||v1||||v2||)\n\nA null cosinus v
```

```
alue says that the query and document vector were orthogonal and had no ma
tch which means that no term of the query was ever encountered in the docu
ment.\n', "The vector space model are the documents which are represented 
as “bags of words”.The basic idea is to represent each document as a vecto
r of certain weighted word frequencies. In order to do so, the following p
arsing and extraction steps are needed. \n1.\tIgnoring case, extract all u
nique words from the entire set of documents. \n2.\tEliminate non-content-
bearing ``stopwords'' such as ``a'', ``and'', ``the'', etc. For sample lis
ts of stopwords.\n3.\tFor each document, count the number of occurrences o
f each word. \n4.\tUsing heuristic or information-theoretic criteria, elim
inate non-content-bearing ``high-frequency'' and ``low-frequency'' words. 
\n5.\tAfter the above elimination, suppose unique words remain. Assign a u
nique identifier between  and to each remaining word, and a unique identif
ier between and to each document. \n", 'In vector space model, the documen
ts from which the information is to be retrieved are represented as vector
s. The term weighting indentifies the success or failure of the vector spa
ce method. Terms are basically the words or any indexing unit used to iden
tify the contents of a text. Furthermore, a term weighting scheme plays an
important role for the similarity measure. The similarity measures largely
identify the retrieval efficiency of a particular information retrieval sy
stem.\n This largely depends on formulas. Where the formulas depend only o
n the frequencies within the document and they not depend on inter-documen
t frequencies. The main components of the formulas are as follows:\nBinar
y:\nBinary formula gives every word that appears in a document equal relev
ance. This can be useful when the number of times a word appears is not co
nsidered important.\nTerm frequency:\nThis formula counts how many times t
he term occurs in a document. The more times a term t occurs in document d
the more likely it is that t is relevant to the document. Used alone, favo
rs common words and long documents. This formula gives more credit to word
s that appears more frequently, but often too much credit.\nAugmented norm
alized term frequency\nThis formula tries to give credit to any word that 
appears and then give some additional credit to words that appear frequent
ly.\nLogarithmic term frequency\nLogarithms are a way to de-emphasize the 
e_ect of frequency. Literature proposes log and alternate log as the most 
used\n']

#task:  20
Signatures metric:  20 x 100
#permutations used to create signatures: 100
```



Question D:The result illustrates that there are 3 pairs of answers which can be identified as near duplicate answers as shown below:

1. Answer D and D(14) with Jaccard similarity of 0.7

2. Answer D and D(18) with Jaccard similarity of 0.68

3. Answer D(18) and A(14) with Jaccard similarity of 0.62

In [ ]:

```
LSH_task('D')
```

```
['In probability theory, Bayes\' theorem (often called Bayes\' law after R
ev Thomas Bayes) relates the conditional and marginal probabilities of two
random events. It is often used to compute posterior probabilities given o
bservations. For example, a patient may be observed to have certain sympto
ms. Bayes\' theorem can be used to compute the probability that a proposed
diagnosis is correct, given that observation. (See example 2)\nAs a formal
theorem, Bayes\' theorem is valid in all common interpretations of probabi
lity. However, it plays a central role in the debate around the foundation
s of statistics: frequentist and Bayesian interpretations disagree about t
he ways in which probabilities should be assigned in applications. Frequen
tists assign probabilities to random events according to their frequencies
of occurrence or to subsets of populations as proportions of the whole, wh
ile Bayesians describe probabilities in terms of beliefs and degrees of un
certainty. The articles on Bayesian probability and frequentist probabilit
y discuss these debates in greater detail.\nBayes\' theorem relates the co
nditional and marginal probabilities of events A and B, where B has a non-
vanishing probability:\n    P(A|B) = \\frac{P(B | A)\\, P(A)}{P(B)}.\nEach
term in Bayes\' theorem has a conventional name:\n    * P(A) is the prior
probability or marginal probability of A. It is "prior" in the sense that
it does not take into account any information about B.\n    * P(A|B) is th
e conditional probability of A, given B. It is also called the posterior p
robability because it is derived from or depends upon the specified value
of B.\n    * P(B|A) is the conditional probability of B given A.\n    * P
(B) is the prior or marginal probability of B, and acts as a normalizing c
onstant.\nIntuitively, Bayes\' theorem in this form describes the way in w
hich one\'s beliefs about observing \'A\' are updated by having observed
\'B\'.\n', '"Bayes\' Theorem" or "Bayes\' Rule", or something called Bayes
ian reasoning\nThe Bayesian Conspiracy is a multinational, interdisciplina
ry, and shadowy group of scientists that controls publication, grants, ten
ure, and the illicit traffic in grad students.  The best way to be accepte
d into the Bayesian Conspiracy is to join the Campus Crusade for Bayes in
high school or college, and gradually work your way up to the inner circle
s. . \nBayes\' Theorem\n  \t \n\nLet  and  be sets. Conditional probabilit
y requires that \n \t(1) \nwhere  denotes intersection ("and"), and also t
hat \n \t(2) \nTherefore, \n \t(3) \nNow, let \n \t(4) \nso  is an event i
n  and  for  , then \n \t(5) \n \t(6) \nBut this can be written \n \t(7)
\nso \n \n\n\nThis paper proposes a new measure called scaled inverse docu
ment frequency (SIDF) which evaluates the conditional specificity of query
terms over a subset S of D and without making any assumption about term in
dependence. S can be estimated from search results, OR searches, or comput
ed from inverted index data. We have evaluated SIDF values from commercial
search engines by submitting queries relevant to the financial investment
domain. Results compare favorably across search engines and queries. Our a
pproach has practical applications for `real-world’ scenarios like in Web
Mining, Homeland Security, and keyword-driven marketing research scenario
s. \n', 'Bayes’ theorem was names after Rev Thomas Bayes and is a method u
sed\nin probability theory. This theorem aims to relate the conditional an
d\nmarginal probabilities of two random events occuring, and given\nvariou
s observations is frequently used to compute subsequent\nprobabilities. Ba
yes’ theorem is also often known as Bayes’ law.\n\nAn example of where Bay
es’ theorem may be used is in the following\nextract: “Suppose there exist
s a school with forty percent females and\nsixty percent males as student
s. The female students can only wear\nskirts or trousers in equal numbers
whereas all the male students can\nonly wear trousers. An observer randoml
y sees a student from a\ndistance and all he can see is that this student
is wearing\ntrousers. What is the probability this student is female?”\n\n
There is a debate amongst frequentists and Bayesians about how Bayes’\nthe
orem plays a major role around the beginnings of statistical\nmathematics.
Frequentist and Bayesian explanations do not agree about\nthe ways in whic
h probabilities should be assigned. This is primarily\nbecause Bayesians a
ssign probabilities in terms of beliefs whereas\nfrequentists assign proba
```

```
bilities to random events according to the\nfrequencies of them occurrin
g.\n', 'Bayes\' theorem relates the conditional and marginal probabilities
of\ntwo random events. For example, a person may be seen to have certain\n
medical symptoms; Bayes\' theorem can then be used to compute the\nprobabi
lity that, given that observation, the proposed diagnosis is\nthe right on
e.\n\nBayes\' theorem forms a relationship between the probabilities xcof
\nevents A and B. Intuitively, Bayes\' theorem in this form describes the
\nway in which one\'s recognition of \'A\' are updated by having observed
\n\'B\'.\n\nP(A | B) = P(B | A) P(A) / P(B)\n\nP(A|B) is the conditional p
robability of A given B. It is derived from or depends upon the specified
value of B, therefore it is also known as the posterior probability.\n\nP
(B|A) is the conditional probability of B given A.\n\nP(A) is the prior pr
obability A. It doesn\'t take into account any information about B, so it
is "prior".\n\nP(B) is the prior or marginal probability of B, and acts to
normalise the probability.\n\nTo derive the theorem, we begin with the def
inition of conditional\nprobability. By combining and re-arranging these t
wo equations for A\nand B, we get a the lemma called product rule for\npro
babilities. Provided that P(B) is not a zero, dividing both sides\nby P(B)
renders us with Bayes\' theorem.\n', 'In probability theory; Bayes theorem
(often called Bayes law after Rev\nThomas Bayes) relates the conditional a
nd marginal probabilities of\ntwo random events. It is used to compute pos
terior probabilities given\nobservations. For example; a person may be obs
erved to have certain\nsymptoms. Bayes theorem can be used to compute the
probability that a\nproposed diagnosis is correct.\n\nAs a formal theorem
Bayes theorem is valid in all common\ninterpretations of probability. Howe
ver, it plays a central role in\nthe debate around the foundations of stat
istics: frequentist and\nBayesian interpretations disagree about the ways
in which\nprobabilities should be assigned to each other. Bayesians descri
be\nprobabilities in terms of beliefs and degrees of uncertainty, While\nf
requentists assign probabilities to random events according to their\nfreq
uencies of occurrence or to subsets of populations as proportions\nof the
whole. The articles on Bayesian probability and frequentist\nprobability d
iscuss these debates in detail.\n', 'Baye’s theorm in connection with cond
itional probabilities is of fundamental importance, since it permits a cal
culation of PROB(AB) from PROB(BA). Statistical information that is often
gathered in great volume can therefore be avoided \n', 'Bayes Theorem is a
n important theorem relating conditional probabilities, it allows us to ca
lculate PROB(A|B) from PROB(B|A). Bayes Theorem is important because it ca
n save us from gathering vast amounts of statistical evidence. The main th
eory is PROB(A|B) = PROB(B|A) * PROB(A) /PROB(B), it means Using PROB(WIN|
RAIN) from earlier, we can find the probability that it rained on a day th
at Harry won a race.\n', "Bayes' theorem relates the conditional and margi
nal probabilities of two random events and is named after the Reverend Tho
mas Bayes (1702–1761), who studied how to compute a distribution for the p
arameter of a binomial distribution. It is valid in all common interpretat
ions of probability. It plays a central role in the debate around the foun
dations of statistics: frequentist and Bayesian interpretations disagree a
bout the ways in which probabilities should be assigned in applications. F
requentists assign probabilities to random events according to their frequ
encies of occurrence or to subsets of populations as proportions of the wh
ole, while Bayesians describe probabilities in terms of beliefs and degree
s of uncertainty. Applications of Bayes' theorem often assume the philosop
hy underlying Bayesian probability that uncertainty and degrees of belief
can be measured as probabilities. One of Bayes' results (Proposition 5) gi
ves a simple description of conditional probability, and shows that it can
be expressed independently of the order in which things occur:\nIf there b
e two subsequent events, the probability of the second b/N and the probabi
lity of both together P/N, and it being first discovered that the second e
vent has also happened, from hence I guess that the first event has also h
appened, the probability I am right [i.e., the conditional probability of
the first event being true given that the second has also happened] is P/
```

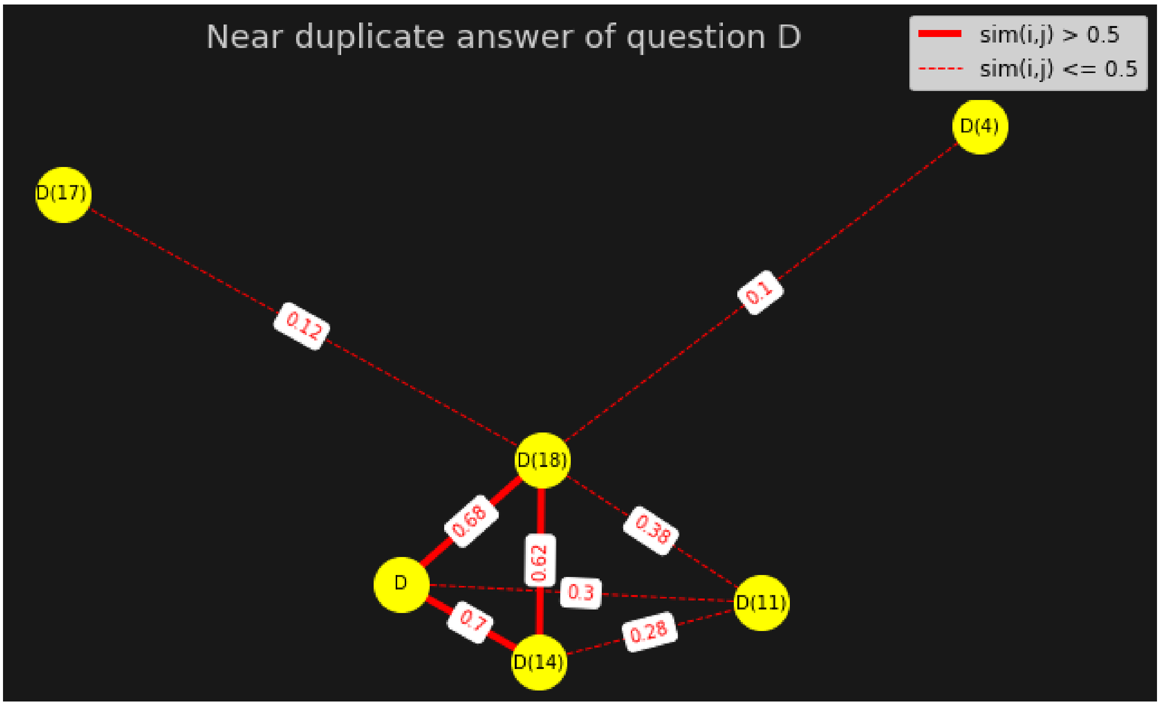
```
b. \nNote that the expression says nothing about the order in which the ev
ents occurred; it measures correlation, not causation.\n", 'Bayes’ theorem
relates the conditional and marginal probabilities of two random events.
It is mainly used to calculate the probability of one event’s outcome give
n that a previous event happened.  For example, the probability that a doc
tors diagnosis is correct given that the doctor had previously observed sy
mptoms in the patient.  Bayes’ theorem can be used for all forms of probab
ility, however it is currently at the centre of a debate concerning the wa
ys in which probabilities should be assigned in applications. \nThe theore
m states that the probability of Event A happening given Event B is the pr
obability of B given A multiplied by the probability of A regardless of B
all divided by the probability of B regardless of A which acts as a normal
ising constant.  Bayes’ theorem formed in this way basically details how o
ne’s beliefs about Event A are renewed or updated knowing that Event B hap
pened.  When calculating conditional probabilities such as these, it is of
ten useful to create a table containing the number of occurrences, or rela
tive frequencies, of each outcome for each of the variables independentl
y.\n\n', 'Bayes Theorem is a mathematical formula used to calculate condit
ional probabilities. Given the probability of event A given event B, Bayes
Theorem can be used to calculate the probability of B given A.  This is ac
hieved using the conditional probability of B given A and the prior probab
ilities of both events A and B. For example: suppose there is a bag of col
oured balls with 25 red ones and 75 black ones. Lucky Joe likes to predict
the colour of the ball he selects and he is 80% accurate. Joe records all
of his results and about 0.5% of the time he accidently records the wrong
results. Using all of this information more probabilities can be inferred,
including using Bayes Theorem to calculate various probabilities like Joe
recording correctly if he guesses correctly or Joe recording incorrectly w
hen his guess was correct (and other like combinations). \n', " In probabi
lity theory, Bayes' theorem (often called Bayes' law after Rev Thomas Baye
s) relates the conditional and marginal probabilities of two random event
s. It is often used to compute posterior probabilities given observations.
For example, a patient may be observed to have certain symptoms. Bayes' th
eorem can be used to compute the probability that a proposed diagnosis is
correct, given that observation.\nAs a formal theorem, Bayes' theorem is v
alid in all common interpretations of probability. However, it plays a cen
tral role in the debate around the foundations of statistics: frequentist
and Bayesian interpretations disagree about the ways in which probabilitie
s should be assigned in applications. \nSuppose there is a co-ed school ha
ving 60% boys and 40% girls as students. The girl students wear trousers o
r skirts in equal numbers; the boys all wear trousers. An observer sees a
(random) student from a distance; all they can see is that this student is
wearing trousers. What is the probability this student is a girl? The corr
ect answer can be computed using Bayes' theorem.\nThe event A is that the
student observed is a girl, and the event B is that the student observed i
s wearing trousers. To compute P(A|B), we first need to know:\nP(B|A'), or
the probability of the student wearing trousers given that the student is
a boy. This is given as 1.\nP(A), or the probability that the student is a
girl regardless of any other information. Since the observers sees a rando
m student, meaning that all students have the same probability of being ob
served, and the fraction of girls among the students is 40%, this probabil
ity equals 0.4.\nP(A'), or the probability that the student is a boy regar
dless of any other information (A' is the complementary event to A). This
is 60%, or 0.6.\nP(B|A), or the probability of the student wearing trouser
s given that the student is a girl. As they are as likely to wear skirts a
s trousers, this is 0.5", '\nIn probability theory, Bayes\' theorem also c
alled Bayes\' law after Rev Thomas Bayes compares the conditional and marg
inal probabilities of two random events. It is often used to calculate pos
terior probabilities given observations. For example, a patient may be obs
erved to have certain symptoms. Bayes\' theorem can be used to calculate t
he likelihood that a proposed analysis is accurate, given that observatio
```

```
n. \nAs an official theorem, Bayes\' theorem is valid in all universal int
erpretations of probability. However, it plays a fundamental role in the d
ebate around the foundations of statistics: frequentist and Bayesian inter
pretations disagree about the ways in which probabilities should be assign
ed in applications.\n Frequentists assign probabilities to random events a
ccording to their frequencies of happening or to subsets of populations as
proportions of the whole. Whilst Bayesians describe probabilities in terms
of beliefs and degrees of uncertainty. The articles on Bayesian probabilit
y and frequentist probability discuss these debates in greater detail.\nBa
yes\' theorem compares the conditional and marginal probabilities of event
s A and B, where B has a non-vanishing probability.\nEach term in Bayes\'
theorem has a conventional name:\nP(A) is the previous probability  of A.
It is "previous" in the sense that it does not take into account any infor
mation about B.\nP(A|B) is the conditional probability of A, given B. It i
s also called the subsequent probability because it is derived from or dep
ends upon the specified value of B.\nP(B|A) is the conditional probability
of B given A.\nP(B) is the previous.\n', "In probability theory, Bayes' th
eorem relates the conditional and marginal probabilities of two random eve
nts. It is often used to compute posterior probabilities given observation
s.\n\nBayes' theorem is expressed mathematically as:\n\nP(A|B) = (P(B|A)P
(A))/P(B)\n\nwhere P(A|B) is the conditional probability of A given B, P
(A) is the prior probability of A, P(B) is the prior probability of B, and
P(B|A) is the conditional probability of B given A. \n\nBayes' theorem rel
ates the conditional and marginal probabilities of two random events P(A)
and P(B), and is valid in all common interpretations of probability. For e
xample, in a school in made up of 3/5 boys and 2/5 girls, the girls wear s
kirts of trousers in equal numbers and the boys all wear trousers. If a st
udent is observed from a distance wearing trousers, Bayes theorem can be u
sed to determine the probability of this student being a girl. \n\nP(A) is
the probability of the student being a girl (which is 2/5). \nP(B|A) is th
e probability of the student wearing trousers given that the student is a
girl, which is 0.5 \nP(B) is the probability of a random student wearing t
rousers, which can be calculated as P(B) = P(B|A)P(A) + P(B|A')P(A') where
' denotes a complementary event, which is 0.8. \nTherefore the probability
using the formula is 0.25. \n\nBayes theorem is often used to compute post
erior probabilities given observations, for instance the probability that
a proposed medical diagnosis is correct, given certain observed symptoms.
\n", 'The Probability of an event happening mean considering the likelihoo
d of or the number of the instance occurring, and dividing this value by t
he total number of events. The equation for this calculation would look as
follows:\nProbability (P) = number of instance / total number of events\nO
n the other hand Probability Theory (P) usually involves assigning values
to events. For example:\n(P)=1: event is certain to occur\n(P)=0: event is
certain NOT to occur\n(P)=0.5: event occurs half of the time. \n\nThere is
also Conditional Probability which is usually interested in the way variab
les relate to each other. Bayes Theorem is the name given to an important
theorem relating\nConditional probabilities and it can be seen as a way of
understanding how the probability that a theory is true, is affected by a
new piece of evidence. It has been used in a wide variety of contexts, ran
ging from marine biology to the development of "Bayesian" spam blockers fo
r email systems. \n', 'In probability theory, Bayes\' theorem (often calle
d Bayes\' law after Rev Thomas Bayes) relates the conditional and marginal
probabilities of two random events. It is often used to compute posterior
probabilities given observations (for example, a patient may be observed t
o have certain symptoms). Bayes\' theorem can be used to compute the proba
bility that a proposed diagnosis is correct, given that observation.\n\nAs
a formal theorem, Bayes\' theorem is valid in all common interpretations o
f probability. However, it plays a central role in the debate around the f
oundations of statistics; frequentist and Bayesian interpretations disagre
e about the ways in which probabilities should be assigned in application
s. Frequentists assign probabilities to random events according to their f
```

```
requencies of occurrence or to subsets of populations as proportions of th
e whole, while Bayesians describe probabilities in terms of beliefs and de
grees of uncertainty. The articles on Bayesian probability and frequentist
probability discuss these debates in greater detail.\n\nBayes\' theorem re
lates the conditional and marginal probabilities of events A and B, where
B has a non-vanishing probability:\n\n    P(A|B) = (P(B | A) x P(A)) / P
(B).\n\nEach term in Bayes\' theorem has a conventional name:\nP(A) is the
prior probability or marginal probability of A. It is "prior" in the sense
that it does not take into account any information about B.\n P(A|B) is th
e conditional probability of A, given B. It is also called the posterior p
robability because it is derived from or depends upon the specified value
of B.\nP(B|A) is the conditional probability of B given A.\nP(B) is the pr
ior or marginal probability of B, and acts as a normalizing constant.\n\nI
ntuitively, Bayes\' theorem in this form describes the way in which one\'s
beliefs about observing \'A\' are updated by having observed \'B\'.', "Bay
es' theorem (often called Bayes' law) connects the conditional and margina
l probabilities of two arbitrary events. One of its uses is calculating po
sterior probabilities given observations. \n\nBayes' theorem plays a key r
ole in the debate around the principles of statistics: frequentist and Bay
esian interpretations disagree about the ways in which probabilities shoul
d be assigned in applications. \n\nBayes' theorem is useful in evaluating
the result of drug tests.  If a test can identify a drug user 99% of the t
ime, and can identify a non-user as testing negative 99% of the time, it m
ay seem to be a relatively accurate test.  However, Bayes' theorem will re
veal the flaw that despite the apparently high accuracy of the test, the p
robability that an employee who tested positive actually did use drugs is
only about 33%.", "In probability theory, the prior and conditional probab
ilities\nof two random events are related by Bayes' theorem. The theorem i
s\noften used when we have observations and wish to compute posterior\npro
babilities.\n\nFor example, given an observation that a patient is seen to
have certain\nsymptoms, we can use Bayes' theorem to compute the probabili
ty that a\nsuggested diagnosis is correct.\n\nP(A) is the prior probabilit
y of A. P(A|B) is the conditional probabilty\nof A given B. P(B|A) is the
conditional probabilty of B given A. P(B) is\nthe prior probability of B,
and must be non-zero. Bayes' theorem is given\nby P(A|B) = (P(B|A)P(A))/(P
(B)).", "In probability theory, Bayes' theorem (or Bayes' law after Rev Th
omas Bayes) provides relation between the conditional and marginal probabi
lities of two random events. It is usually used to calculate posterior pro
babilities given observations. For example: a patient might be observed to
show certain symptoms. Bayes' theorem could be used to compute the probabi
lity that a certain diagnosis is right, given that observation.\n\nSince i
t is a formal theorem, Bayes' theorem holds in all popular interpretations
of probability. \nBayes' theorem relates the conditional and marginal prob
abilities of events a and b, where b has a non-vanishing probability:\n\n
P(a|b) = P(a|b)P(a)/P(b)\n\nTerms in Bayes' theorem are named by a convent
ion:\n\nP(A) is the prior probability or marginal probability of A. It doe
s not take into account any information about B and therefore is considere
d “prior”.\nP(A|B) is the conditional probability of A, given B. It it is
derived from or depends upon the specified value of B. Usually it is calle
d the posterior probability \nP(B|A) is the conditional probability of B g
iven A.\nP(B) (a.k.a. the normalizing constant) is the prior or marginal p
robability of B.\n\nObviously, Bayes' theorem describes the way in which o
ne's assumptions about observing the event'a' are changed by having observ
ed the event 'b'.\n", 'In probability theory, Bayes\' theorem relates the
conditional and marginal probabilities of two random events. It is usually
be used to compute posterior probabilities given observations. For instanc
e, a patient may be observed to have certain symptoms. Bayes\' theorem can
be used to compute the probability that a proposed diagnosis is correct, g
iven that observation. \nAs a formal theorem, Bayes\' theorem is valid in
all common interpretations of probability. However, it plays a central rol
e in the debate around the foundations of statistics: frequentist and Baye
```

```
sian interpretations disagree about the ways in which probabilities should
be assigned in applications. The articles on Bayesian probability and freq
uentist probability discuss these debates in greater detail. Frequentists
assign probabilities to random events according to their frequencies of oc
currence or to subsets of populations as proportions of the whole. At the
same time,  Bayesians describe probabilities in terms of beliefs and degre
es of uncertainty. \nBayes\' theorem relates the conditional and marginal
probabilities of events A and B, where B has a non-vanishing probabilit
y:\nEach term in Bayes\' theorem has a conventional name:\n•\tP(A) is the
prior probability or marginal probability of A. It is "prior" in the sense
that it does not take into account any information about B. \n•\tP(A|B) is
the conditional probability of A, given B. It is also called the posterior
probability because it is derived from or depends upon the specified value
of B. \n•\tP(B|A) is the conditional probability of B given A. \n•\tP(B) i
s the prior or marginal probability of B, and acts as a normalizing consta
nt. \nIntuitively, Bayes\' theorem in this form describes the way in which
one\'s beliefs about observing \'A\' are updated by having observed \'B
\'.\n', 'Bayes\' Theorem is a simple mathematical formula used for calcula
ting conditional probabilities. Bayes\' Theorem is a theorem of probabilit
y theory originally stated by the Reverend Thomas Bayes. It figures promin
ently in subjectivist or Bayesian approaches to epistemology, statistics,
and inductive logic. It can be seen as a way of understanding how the prob
ability that a theory is true is affected by a new piece of evidence. It h
as been used in a wide variety of contexts, ranging from marine biology to
the development of "Bayesian" spam blockers for email systems. In the phil
osophy of science, it has been used to try to clarify the relationship bet
ween theory and evidence. Many insights in the philosophy of science invol
ving confirmation, falsification, the relation between science and pseudos
ience, and other topics can be made more precise, and sometimes extended o
r corrected, by using Bayes\' Theorem. Subjectivists, who maintain that ra
tional belief is governed by the laws of probability, lean heavily on cond
itional probabilities in their theories of evidence and their models of em
pirical learning. Bayes\' Theorem is central to these enterprises both bec
ause it simplifies the calculation of conditional probabilities and becaus
e it clarifies significant features of subjectivist position. Indeed, the
Theorem\'s central insight – that a hypothesis is confirmed by any body of
data that its truth renders probable – is the cornerstone of all subjectiv
ist methodology. ']

#task:  20
Signatures metric:  20 x 100
#permutations used to create signatures: 100
```



Question E: The result illustrates that there are no pairs of answers which can be identified as near duplicate answers

In [ ]:

```
LSH_task('E')
```

```
['In mathematics and computer science, dynamic programming is a method of
solving problems that exhibit the properties of overlapping subproblems an
d optimal substructure (described below). The method takes much less time
than naive methods.\nThe term was originally used in the 1940s by Richard
Bellman to describe the process of solving problems where one needs to fin
d the best decisions one after another. By 1953, he had refined this to th
e modern meaning. The field was founded as a systems analysis and engineer
ing topic that is recognized by the IEEE. Bellman\'s contribution is remem
bered in the name of the Bellman equation, a central result of dynamic pro
gramming which restates an optimization problem in recursive form.\nThe wo
rd "programming" in "dynamic programming" has no particular connection to
computer programming at all, and instead comes from the term "mathematical
programming", a synonym for optimization. Thus, the "program" is the optim
al plan for action that is produced. For instance, a finalized schedule of
events at an exhibition is sometimes called a program. Programming, in thi
s sense, means finding an acceptable plan of action, an algorithm.\nOptima
l substructure means that optimal solutions of subproblems can be used to
find the optimal solutions of the overall problem. For example, the shorte
st path to a goal from a vertex in a graph can be found by first computing
the shortest path to the goal from all adjacent vertices, and then using t
his to pick the best overall path, as shown in Figure 1. In general, we ca
n solve a problem with optimal substructure using a three-step process:\n
1. Break the problem into smaller subproblems.\n   2. Solve these problems
optimally using this three-step process recursively.\n   3. Use these opti
mal solutions to construct an optimal solution for the original problem.\n
The subproblems are, themselves, solved by dividing them into sub-subprobl
ems, and so on, until we reach some simple case that is solvable in consta
nt time.\nFigure 2. The subproblem graph for the Fibonacci sequence. That
it is not a tree but a DAG indicates overlapping subproblems.\nTo say that
a problem has overlapping subproblems is to say that the same subproblems
are used to solve many different larger problems. For example, in the Fibo
nacci sequence, F3 = F1 + F2 and F4 = F2 + F3 — computing each number invo
lves computing F2. Because both F3 and F4 are needed to compute F5, a naiv
e approach to computing F5 may end up computing F2 twice or more. This app
lies whenever overlapping subproblems are present: a naive approach may wa
ste time recomputing optimal solutions to subproblems it has already solve
d.\nIn order to avoid this, we instead save the solutions to problems we h
ave already solved. Then, if we need to solve the same problem later, we c
an retrieve and reuse our already-computed solution. This approach is call
ed memoization (not memorization, although this term also fits). If we are
sure we won\'t need a particular solution anymore, we can throw it away to
save space. In some cases, we can even compute the solutions to subproblem
s we know that we\'ll need in advance.\n', 'Dynamic programming is a metho
d of providing solutions to potential problems exhibiting the properties o
f overlapping sub problems and optimal structure. This is highly used in d
ynamic programming. The advantage being the less time consumption in compa
rison to other amateur methods.\nIt has to be kept in mind that the term p
rogramming in the field has got nothing to do with computer programming at
all. On the other hand it is derived from the term mathematical programmin
g which is a similar word used for optimization. Here by meaning that a pr
ogram can be an optimal plan for the produced action. The typical example
could be of a finalized schedule of events at an exhibition. This leads to
the concept of programming being a helper in finding an acceptable plan of
action, which can also be termed as an algorithm\nThe subproblems are, the
mselves, solved by dividing them into sub-subproblems, and so on, until we
reach some simple case that is solvable in constant time.\nOverlapping sub
problems means that the same subproblems are used to solve many different
larger problems. Example could be of Fibonacci sequence; F3 = F1 + F2 and
F4 = F2 + F3 — computing each number involves computing F2. Because both F
3 and F4 are needed to compute F5, a naive approach to computing F5 may en
d up computing F2 twice or more. It means that whenever we encounter with
```

```
overlapping subproblems, a naive approach may waste to,e recomputing optim
al solutions to the already solved subproblems.\n', ' Dynamic programming
is a method for efficiently solving a broad range of search and optimizati
on problems which exhibit the characteristics of overlappling.  Dynamic pr
ogramming. Design technique, like divide-and-conquer method.\n\nThe leadin
g and most up-to-date textbook on the far-ranging algorithmic methododogy
of Dynamic Programming, which can be used for optimal control, ...\nThe wo
rd Programming in the name has nothing to do with writing computer program
s. Mathematicians use the word to describe a set of rules which anyone can
follow to solve a problem. They do not have to be written in a computer la
nguage. \nDynamic programming was the brainchild of an American Mathematic
ian, Richard Bellman, who described the way of solving problems where you
need to find the best decisions one after another. In the forty-odd years
since this development, the number of uses and applications of dynamic pro
gramming has increased enormously. \nFor example, in 1982 David Kohler use
d dynamic programming to analyse the best way to play the game of darts.
\n\n1.\t   In recent years, dynamic programming languages develope very fa
stly, especially PHP and Ruby. There is no doubt that They have already be
came the first choice for many programmerers when developing web applicati
ons..When you learn a new natural language and you start to use it you nat
urally, you find yourself using new concepts and paradigms that enrich the
use of the language you already know; expect the same result with computer
languages.\n\n', 'Dynamic Programming is an algorithm design technique use
d for optimisation problems, such as minimising or maximising. Like divide
and conquer, Dynamic Programming solves problems by combining solutions to
sub-problems. However, unlike divide and conquer, sub-problems are not alw
ays independent as sub-problems may share sub-sub-problems but solution to
one sub-problem may not affect the solutions to other sub-problems of the
same problem.\n\nThere are four steps in Dynamic Programming:\n\n1. Charac
terise structure of an optimal solution.\n\n2. Define value of optimal sol
ution recursively.\n\n3. Compute optimal solution values either top-down w
ith caching or bottom-up in a table.\n\n4. Construct an optimal solution f
rom computed values.\n\nAn example of the type of problem for which Dynami
c Programming may be used is: given two sequences, X=(x1,...,xm) and Y=(y
1,...,yn) find a common subsequence whose length is maximum.\n\nDynamic Pr
ogramming reduces computation by solving sub-problems in a bottom-up fashi
on and by storing solution to a sub-problem the first time it is solved. A
lso, looking up the solution when a sub-problem is encountered again helps
reduce computation. However, the key in Dynamic Programming is to determin
e the structure of optimal solutions. \n', 'Dynamic programming is a metho
d for solving mathematical programming problems that exhibit the propertie
s of overlapping subproblems and optimal substructure. This is a much quic
ker method than other more naive methods. The word "programming" in "dynam
ic programming" relates optimization, which is commonly referred to as mat
hematical programming. Richard Bellman originally coined the term in the 1
940s to describe a method for solving problems where one needs to find the
best decisions one after another, and by 1953, he refined his method to th
e current modern meaning.\n\nOptimal substructure means that by splitting
the programming into optimal solutions of subproblems, these can then be u
sed to find the optimal solutions of the overall problem. One example is t
he computing of the shortest path to a goal from a vertex in a graph. Firs
t, compute the shortest path to the goal from all adjacent vertices. Then,
using this, the best overall path can be found, thereby demonstrating the
dynamic programming principle. This general three-step process can be used
to solve a problem:\n\n1. Break up the problem different smaller subproble
ms.\n\n2. Recursively use this three-step process to compute the optimal p
ath in the subproblem.\n\n3. Construct an optimal solution, using the comp
uted optimal subproblems, for the original problem.\n\nThis process contin
ues recursively, working over the subproblems by dividing them into sub-su
bproblems and so forth, until a simple case is reached (one that is easily
solvable). \n', 'In computer science; dynamic programming is a way of solv
```

```
ing problems consist of overlapping subproblems and optimal substructure.
The method is more effiecent than naive methods.\n\nThe term was first coi
ned in the 1940s by Richard Bellman to describe the process of solving pro
blems where you need to find the best decisions consecutavly. In 1953 he h
ad refined this to the modern meaning. The field was founded as a systems
analysis and engineering topic that is recognized by the IEEE. Bellman equ
ation is a central result of dynamic programming which restates an optimiz
ation problem in recursive form.\n\ndynamic programming has little connect
ion to computer programming at all, and instead comes from the term mathem
atical programming, a synonym for optimization. Thus, the program is the b
est plan for action that is produced. For instance, a events schedule at a
n exhibition is sometimes called a program. Programming means finding a pl
an of action. \n', 'Dynamic programming (DP) is an extremely powerful, gen
eral tool for solving optimization difficulties on left-right-ordered ite
m, for example character strings. It is similar to divide and conquer, how
ever is differentiated as its subproblems are not independent. It is easil
y applicable, in relative terms, once understood. However until one has wi
tnessed enough examples, it looks like magic.\n\nDP minimizes computation
by solving subproblems from the base upwards, storing solution to a subpro
blem when it is initially conquered, and looking up the solution when the
subproblem is experienced for a second time. \n', 'dynamic programming is
a method of solving problems that exhibit the properties of overlapping su
bproblems and optimal substructure (described below). The method takes muc
h less time than naive methods. The word "programming" in "dynamic program
ming" has no particular connection to computer programming at all, and ins
tead comes from the term "mathematical programming", a synonym for optimiz
ation. Thus, the "program" is the optimal plan for action that is produce
d. For instance, a finalized schedule of events at an exhibition is someti
mes called a program. Programming, in this sense, means finding an accepta
ble plan of action, an algorithm. \n', 'Dynamic programming is an algorith
mic technique used to solve certain optimization problems where the object
is to find the best solution from a number of possibilities. It uses a so
called ‘bottom-up’ approach, meaning that the problem is solved as a set o
f sub-problems which in turn are made up of sub-sub-problems.Sub-problems
are then selected and used to solve the overall problem. These sub-problem
s are only solved once and the solutions are saved so that they will not n
eed to be recalculated again. Whilst calculated individually, they may als
o overlap. When any sub-problem is met again, it can be found and re-used
to solve another problem. Since it searches all possibilities, it is also
very accurate. This method is far more efficient than recalculating and th
erefore considerably reduces computation. It is widely used in computer sc
ience and can be applied for example, to compress data in high density bar
codes.\n\nDynamic programming is most effective and therefore most often u
sed on objects that are ordered from left to right and whose order cannot
be rearranged. This means it works well on character chains for example.
\n\n', 'In mathematics and computer science, dynamic programming is a meth
od of solving problems that exhibit the properties of overlapping sub prob
lems and optimal substructure. The term was originally used in the 1940s b
y Richard Bellman to describe the process of solving problems where one ne
eds to find the best decisions one after another. By 1953, he had refined
this to the modern meaning. Bellman\'s contribution is remembered in the n
ame of the Bellman equation, a central result of dynamic programming which
restates an optimization problem in recursive form. The word "programming"
in "dynamic programming" has no particular connection to computer programm
ing at all, and instead comes from the term "mathematical programming", a
synonym for optimization. Thus, the "program" is the optimal plan for acti
on that is produced. For instance, a finalized schedule of events at an ex
hibition is sometimes called a program. Programming, in this sense, means
finding an acceptable plan of action, an algorithm.\nDynamic programming u
sually takes one of two approaches, the top-down approach, the problem is
broken into sub problems, and these sub problems are solved and the soluti
```

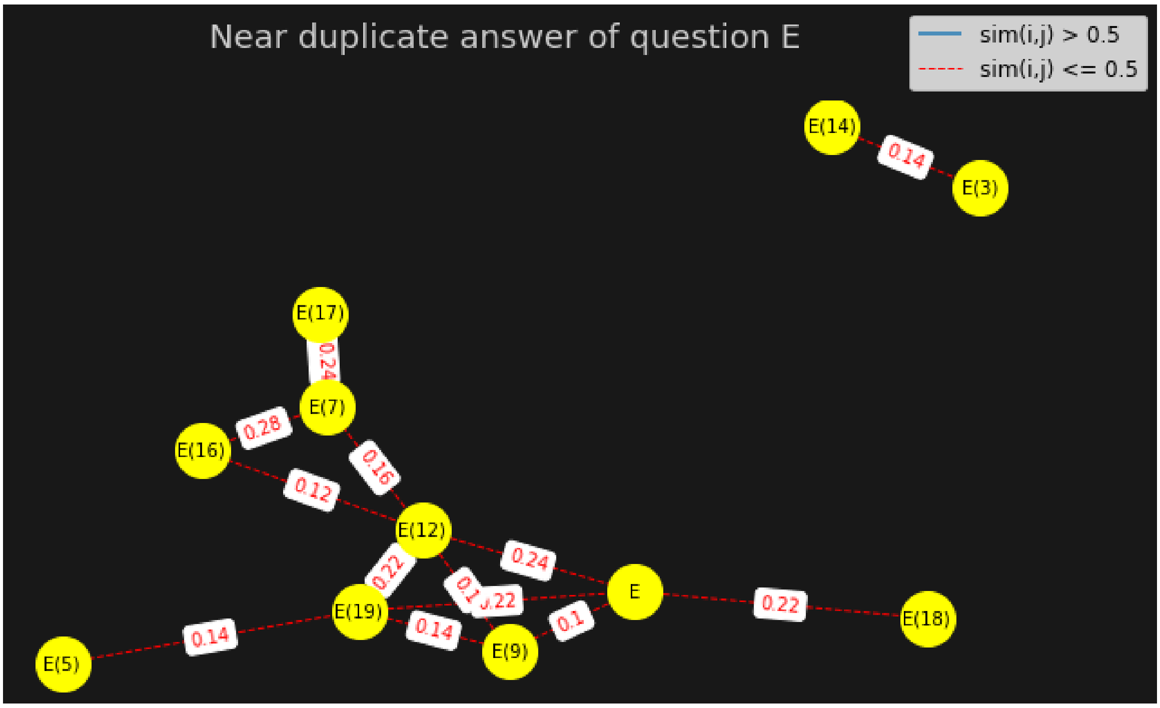
```
ons remembered, in case they need to be solved again. This is recursion an
d memorization combined together and the bottom-up approach, all sub probl
ems that might be needed are solved in advance and then used to build up s
olutions to larger problems. This approach is slightly better in stack spa
ce and number of function calls, but it is sometimes not intuitive to figu
re out all the sub problems needed for solving the given problem.\nSome pr
ogramming languages can automatically memorize the result of a function ca
ll with a particular set of arguments, in order to speed up call-by-name.
Some languages make it possible portably (e.g. Scheme, Common Lisp or Per
l), some need special extensions.This is only possible for a referentially
transparent function.\n\n', 'Dynamic programming is a faster method of sol
ving problems that make use of optimal substructure, overlapping sub-probl
ems and memoization. It has no relationship to computer programming; inste
ad it is a process of finding a satisfactory algorithm. \n\nOptimal substr
ucture is the process of using the optional solutions to sub problems to f
ind the optimal solution to the overall problem. When the same sub problem
solutions can be used to solve various bigger problems it is said to have
overlapping-sub problems. Memoization is used in order to save time the so
lutions are stored rather than be recomputed. A solution can be disposed o
f once we are positive that it will no longer be required, in some cases a
solution to a future problem can be computed in advance. \n\nThere are two
main approaches for dynamic programming. The first is the bottom up approa
ch. Although it is not always simple to find all of them, any required sub
problems are solved in advance and then used to create solutions to larger
problems. The other method is the top down approach which is a method that
combines memorization and recursion. The main problem is divided into sub
problems which are solved and stored for future use.\n', 'Dynamic Programm
ing is a very powerful mathematical technique, often utilised in programmi
ng, for solving optimization problems. Normally, minimizing or maximizin
g.\n\x91Greedy\x92 algorithms focus on making the best local choice at eac
h decision making stage. Without a proof of correctness, such an algorithm
is likely to fail. With Dynamic Programming, we can design our own algorit
hm which searches for all possibilities (which ensures correctness) whilst
storing the results to avoid having to recomputed (leading to computationa
l efficiency).\nDynamic Programming solves problems by combining the solut
ions of subproblems. These subproblems are not, however, independent. Subp
roblems can share subsubproblems, but the solution to one subproblem doesn
\x92t necessarily affect the solutions to other subproblems stemming from
the same problem.\nDynamic Programming reduces computation time by solving
subproblems in a \x91bottom-up\x92 way. It stores the solution to a subpro
blem the first time it is solved, meaning that it can look up the solution
when that subproblem is encountered subsequently. \nThe key to Dynamic Pro
gramming is to find the structure of optimal solutions. The steps required
are as follows:\n1.\tGeneralise the structure of an optimal solution\n2.\t
Recursively define the value of an optimal solution\n3.\tCompute the optim
al solution values either top-down (with caching), or bottom-up using a ta
ble\n4.\tGenerate the optimal solution of these computed values\n', '\nIn
mathematics and computer science, dynamic programming is a method of solvi
ng problems, that exhibit the properties of overlapping subproblems and op
timal substructure. The method takes much less time than naive methods.\nT
he term was originally used in the 1940s to describe the process of solvin
g problems where one needs to find the best decisions one after another.
\nThe field was founded as a systems analysis and engineering topic that i
s recognized by the IEEE\nThe word "programming" in "dynamic programming"
has no particular connection to computer programming at all, and instead c
omes from the term "mathematical programming", a synonym for optimization.
Thus, the "program" is the optimal plan for action that is produced. For i
nstance, a finalized schedule of events at an exhibition is sometimes call
ed a program. Programming, in this sense, means finding an acceptable plan
of action, an algorithm.\nOptimal substructure means that optimal solution
s of subproblems can be used to find the optimal solutions of the overall
```

```
problem. For example, the shortest path to a goal from a vertex in a graph
can be found by first computing the shortest path to the goal from all adj
acent vertices, and then using this to pick the best overall path.\nIn gen
eral, we can solve a problem with optimal substructure using a three-step
process:\n1.Break the problem into smaller subproblems.\n2.solve these pro
blems optimally using this three-step process recursively.\n3.Use these op
timal solutions to construct an optimal solution for the original proble
m.\nThe subproblems are, themselves, solved by dividing them into sub-subp
roblems, and so on, until we reach some simple case that is solvable in co
nstant time.\n', 'Dynamic programming is a problem-solving method which so
lves recursive problems. The term is derived from mathematical programming
which is commonly referred to as optimisation, hence dynamic programming i
s an optimal method of solving the problems and takes much less time than
naïve methods. \n\nDynamic programming uses the properties of optimal subs
tructure, overlapping subproblems and memoization to create an algorithm t
o solve such problems. Optimal substructure means that the structure of th
e problem is made up of sub-problems which can be used to find the solutio
n to the problem overall. A problem with overlapping subproblems means tha
t the same subproblems may be used to solve many different larger problem
s. Each sub-problem is solved by being divided into sub-subproblems, until
a case is reached which is solvable in constant time. Memoization stores s
olutions which have already been computed in order to reduce unnecessary r
e-computation. \n\nDynamic programming can be divided into two main approa
ches: top-down and bottom-up. The top-down approach breaks the problem int
o subproblems, which are solved and remembered, using a combination of mem
oization and recursion. The bottom-up approach solves all subproblems that
might be need in advance, and then uses these solutions to build up the so
lutions to the bigger problem. \n', 'Dynamic Programming (DP) is in basic
terms an algorithm design technique that is used for optimization problems
and often involves minimizing or maximizing. \nFurthermore, by combining s
olutions to subproblems, DP solves problems. Subproblems may include and c
ontain many other subsubproblems and even in such cases, the solution to o
ne subproblem may not affect the solutions to other subproblems involved i
n the same problem. \nBy solving subproblems in a bottom-up fashion, which
is basically when storing solution to a subproblem the first time it is so
lved and looking up to find the solution when a subproblem is come across
once more,  this would cause DP to reduce computations.  \nThe following i
s a generalization path to be taken in Dynamic Programming:\nFirstly it is
needed to Characterize the structure of an optimal solution. Secondly to d
efine the value of the optimal solution recursively. Furthermore, to compu
te the optimal solution values either by following a top-down method with
caching, or a bottom-up method in a table. The last point would be to cons
truct an optimal solution from the computed values.\n', "In the field of c
omputer science, term 'dynamic programming' relates to the style of progra
mming that breaks a large problem down into smaller subproblems, and gener
ally allows for the finding of the optimal solution. When the problem is s
plit into subproblems, these themselves may be split into smaller problem
s, and so on, until they cannot be reduced any more.\nIt is also common fo
r dynamic programming to make use of recursion, and the saving of previous
results for faster computation later; this also leads to higher efficienc
y, as calculations are not being redone. For example, when a problem is re
duced into sub problems, and those are then reduced further, it may be tha
t there are common subsubproblems, and so only one calculation needs to be
done and the result saved to help solve more than one subproblem.\nAn exam
ple of this gain in efficiency is a path-finding problem. If there are two
distinct routes in a network of 10 nodes, tagged A to J, then if the two r
outes share a common section (say, between nodes B and D), the cost of tha
t section should be calculated for the first route and saved. Then, when t
he second route is being processed, the cost of B to D does not need to be
calculated again.\nIn general, dynamic programming is used on optimisation
problems, where the most efficient solution is needed. Areas where this so
```

```
rt of programming is useful is in AI, computer graphics, compression routi
nes, and biomedical applications.", 'Dynamic Programming is a method of so
lving problems that exhibit the properties of overlapping subproblems and
optimal substructure.  The term was originally used in the 1940s by Richar
d Bellman.\n\nThe word "programming" in "dynamic programming" has no parti
cular connection to computer programming at all, and instead comes from th
e term "mathematical programming", a synonym for optimization.  The "progr
am" is the optimal plan for action that is produced.\n\nFor instance, a fi
nalized schedule of events at an exhibition is sometimes called a program.
Programming, in this sense, means finding an acceptable plan of action. \n
\nTo say that a problem has overlapping subproblems is to say that the sam
e subproblems are used to solve many different larger problems.  Optimal s
ubstructure means that optimal solutions of subproblems can be used to fin
d the optimal solutions of the overall problem.', 'In computer science and
mathematics, dynamic programming\nis a method of problem solving that util
ises the properties\nof overlapping subproblems and optimal substructure.
And thus\nthe method takes much less time than more naive methods.\n\nIn
"dynamic programming", the word "programming" has no\nreal connection to c
omputer programming at all, it actually\ncomes from the term "mathematical
programming",\na synonym for optimisation. Thus, the "program" is the opti
mal\nplan of action that is being produced. For example, a\nschedule of ev
ents at an exhibition is sometimes called a\nprogramme. Programming, in th
is sense, means finding an\nacceptable plan, an algorithm.', 'In mathemati
cs and computer science, dynamic programming is a method of solving proble
ms that exhibit the properties of overlapping subproblems and optimal subs
tructure.\n \nThe word "programming" in "dynamic programming" has no parti
cular connection to computer programming at all, and instead comes from th
e term "mathematical programming", a synonym for optimization. Programmin
g, in this sense, means finding an acceptable plan of action, an algorith
m.\n\nOptimal substructure means that optimal solutions of subproblems can
be used to find the optimal solutions of the overall problem. In general,
we can solve a problem with optimal substructure using a three-step proces
s:\n\n   1. Break the problem into smaller subproblems.\n   2. Solve these
problems optimally using this three-step process recursively.\n   3. Use t
hese optimal solutions to construct an optimal solution for the original p
roblem.\n\nThe subproblems are, themselves, solved by dividing them into s
ub-subproblems, and so on, until we reach some simple case that is solvabl
e in constant time.\n\nTo say that a problem has overlapping subproblems i
s to say that the same subproblems are used to solve many different larger
problems. For example, in the Fibonacci sequence, F3 = F1 + F2 and F4 = F2
+ F3 — computing each number involves computing F2. Because both F3 and F4
are needed to compute F5, a naive approach to computing F5 may end up comp
uting F2 twice or more. This applies whenever overlapping subproblems are
present: a naive approach may waste time recomputing optimal solutions to
subproblems it has already solved.\n\nIn order to avoid this, we instead s
ave the solutions to problems we have already solved. Then, if we need to
solve the same problem later, we can retrieve and reuse our already-comput
ed solution. If we are sure we won\'t need a particular solution anymore,
we can throw it away to save space. In some cases, we can even compute the
solutions to subproblems we know that we\'ll need in advance.\n\ndynamic p
rogramming makes use of:\n\n    Overlapping subproblems\n    Optimal subst
ructure\n    Memoization\n\nDynamic programming usually takes one of two a
pproaches:\n\n    Top-down approach\n    Bottom-up approach\n\n\n', 'In ma
thematics and computer science, dynamic programming is a methodology of th
e solution of the problems that exhibit the properties of overlapping subp
roblems and optimal substructure (described below). The methodology takes
much less time rather than naive methods.\nThe term was originally used du
ring the 1940s by Richard Bellman to describe the process of solving probl
ems where one needs to find the best decisions one after another. By 1953,
he had refined this to the modern meaning. The field was founded as a syst
ems analysis and engineering topic that is recognized by the IEEE. Bellman
```

```
\'s contribution is remembered in the name of the Bellman equation, a cent
ral result of dynamic programmer, which restates an optimization problem i
n recursive form.\nThe word "programming" in "dynamic programming" has no
particular connection to computer programming in general , and instead of
this it comes from the term "mathematical programming", a synonym for opti
mization. Therefore, the "program" is the optimal plan for action that is
produced. For example, a finalized schedule of events at an exhibition is
sometimes called a program. \nOptimal substructure means that optimal solu
tions of subproblems can be used to find the optimal solutions of the over
all problem. For instance, the shortest path to a goal from a vertex in a
graph can be found by first computing the shortest path to the goal from a
ll adjacent vertices. After this, it is using this to pick the best overal
l path. In a word, we can solve a problem with optimal substructure using
a three-step process.\n']

#task:  20
Signatures metric:  20 x 100
#permutations used to create signatures: 100
```



2. Package: datasketch // Dataset: News headlines

We will implement MinHash LSH Forest,which takes a MinHash data sketch of the query set, and returns the top-k matching sets that have the approximately highest Jaccard similarities with the query set, to create news headlines recommendation

Step 1: Install and Import packages

In [ ]:

```
pip install datasketch
```

```
Requirement already satisfied: datasketch in /usr/local/lib/python3.6/dist
-packages (1.5.3)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.6/dis
t-packages (from datasketch) (1.19.5)
```

In [ ]:

```
import numpy as np
import pandas as pd
import re
import time
from datasketch import MinHash, MinHashLSHForest
```

Step 2: Acquire data using web scraping package (BeautifulSoup)

In [ ]:

```
from bs4 import BeautifulSoup
import requests

url1 = 'https://www.reuters.com/news/archive/technologynews?view=page&page=1&pageSize=1
0'
url2 = 'https://www.reuters.com/news/archive/technologynews?view=page&page=2&pageSize=1
0'
url3 = 'https://www.reuters.com/news/archive/technologynews?view=page&page=3&pageSize=1
0'
url4 = 'https://www.reuters.com/news/archive/technologynews?view=page&page=4&pageSize=1
0'
url5 = 'https://www.reuters.com/news/archive/technologynews?view=page&page=5&pageSize=1
0'

def web_scraping_news(url):
    r = requests.get(url)
    html = r.text
    soup = BeautifulSoup(html,'lxml')
    div_tag = soup.find_all('h3',attrs={'class':"story-title"})
    news = [i for i in div_tag]

    return news
```

In [ ]:

```
news1 = web_scraping_news(url1)
news2 = web_scraping_news(url2)
news3 = web_scraping_news(url3)
news4 = web_scraping_news(url4)
news5 = web_scraping_news(url5)
news = news1 + news2 + news3 + news4 + news5
titles = [str(i) for i in news]

#Extract News Headline
headlines = [a.strip(''''<h3 class="story-title">
                                                                ''').rstrip('</') for a
in titles]

for headline in headlines:
    print(headline)
```

```
U.S. thanks Taiwan for support for auto chips in key trade meeting
Robinhood lifts trading restrictions on all stocks, including GameStop
PayPal says to shut domestic payments business in India
Shares of Tencent-backed Kuaishou triple in HK debut as retail frenzy cont
inues
Australian drone firm reshapes strategy over Google pull-out threat
Facebook faces a reckoning in Myanmar after blocked by military
GameStop, 'Reddit rally' stocks slide more, Yellen vows scrutiny
Yellen seeks to 'understand deeply' GameStop frenzy as market regulators m
eet
In GameStop saga, U.S. regulator examining all aspects and parties: source
s
Samsung considers Austin for $17 billion chip plant, eyes tax breaks of at
least $806 million: documents
Biden calls for expanded efforts to protect LGBTQ rights globally
Biden set to accept more refugees after years of Trump restrictions
U.S. House punishes Republican congresswoman over incendiary remarks
Amazon bucks UK's grim labour market with 1,000 apprenticeships
Looming Apple privacy changes weigh on Snap despite revenue growth
SoftBank third-quarter earnings recovery seen driven by IPO boom
Activision Blizzard's annual sales forecast tops estimates on 'Call of Dut
y' boost
Pinterest beats estimates on ad spending recovery, strong user growth
T-Mobile beats quarterly postpaid phone additions estimates
Reddit rally' stocks bounce on day after selloff, then dip after hours
U.S. Treasury's Yellen to meet financial regulators Thursday to discuss  v
olatility
Koss family rakes it in from Reddit-fueled rally
Instagram removes hundreds of accounts tied to username hacking
U.S. Senate Democrats push ahead on road to new COVID-19 relief
America is back' - Biden touts muscular foreign policy in first diplomatic
speech
Trump rejects call to testify at his impeachment trial
BNY Mellon, Google Cloud technology to predict Treasury settlement failure
s
Texas court considers hearing on changing venue of Google antitrust case
Exclusive-White House pulled into unionization effort at Amazon facility
Google phone cameras will read heart, breathing rates with AI help
GameStop rises, AMC dips in early U.S. premarket trading
Xilinx to supply chips to Fujitsu for U.S. 5G network gear
Hit by shortage, Volkswagen demands boost to Europe chip sector
Elon Musk, back on Twitter, turns his attention to Dogecoin
EU electric and plug-in hybrid car sales jump to over 1 million in 2020
Exclusive: China's Ant to hive off credit data in revamp; sees IPO in 2 ye
ars - sources
Biden says war in Yemen 'has to end,' U.S. will continue to support Saudi
Arabia
Biden to pursue arms control, seeks to engage China: U.S. envoy
In uneasy truce, House Republicans fail to punish Greene or Cheney
Nokia warns of "challenging" year as it plays catch-up
Mazda expects chip shortage to affect about 7,000 vehicles in February
Uber's Mideast business Careem sees recovery slowing as infections rise
NatWest latest UK bank to switch to Mastercard debit cards from Visa
Ethereum scales record peak before futures launch
Five things to watch in Reddit stocks trading mania
Taiwan says auto chip shortage not a main topic for coming U.S. meeting
Parler CEO John Matze says he was fired by board
Robinhood to allow buying fractional shares of GameStop, AMC
Alibaba sets initial price guidance on $5 billion bond offering: term shee
t
U.S. House Republican leader does not plan to oust Cheney from leadership
```

```
post: CNN
Pentagon, stumped by extremism in ranks, orders stand-down in next 60 days
Biden decides to stick with Space Force as branch of U.S. military
Two Google engineers resign over firing of AI ethics researcher Timnit Geb
ru
Amazon plans AI-powered cameras in delivery vans to improve driver safety
Qualcomm shares drop as chip supply constraints hold back sales
Ebay earnings beat on pandemic-driven surge in online shopping; shares soa
r
PayPal profit tops estimates as pandemic drives online spending to record
levels
SEC studies social media posts for signs of fraud in GameStop frenzy: Bloo
mberg
Number of GameStop shares shorted edges higher: S3 Partners
Spotify outlook weakens as pandemic uncertainty persists
Ant Group reaches deal with China regulators on restructuring - source
Daimler to spin off truck unit, sharpen investor focus on Mercedes-Benz
Yellen calls for 'acting now - and acting big' on pandemic relief
Schumer, after Biden meeting, says Democrats united on a 'bold' COVID-19 b
ill
Biden tells congressional Democrats would consider limits on who gets COVI
D-19 checks
```

Step 3: Create functions to perform news headlines recommendations

In [ ]:

```
def preprocess(text, char_ngram=5):

    return set(text[head:head + char_ngram] for head in range(0, len(text) - char_ngram
))

def get_forest(data, perms):
    start_time = time.time()

    minhash = []

    for text in data['title']:
        tokens = preprocess(text)
        m = MinHash(num_perm=perms)
        for s in tokens:
            m.update(s.encode('utf8'))
        minhash.append(m)

    forest = MinHashLSHForest(num_perm=perms)

    for i,m in enumerate(minhash):
        forest.add(i,m)

    forest.index()

    print('It took %s seconds to build forest.' %(time.time()-start_time))

    return forest

def predict(text, data, perms, num_results, forest):
    start_time = time.time()

    tokens = preprocess(text)
    m = MinHash(num_perm=perms)
    for s in tokens:
        m.update(s.encode('utf8'))

    idx_array = np.array(forest.query(m, num_results))
    if len(idx_array) == 0:
        return None # if your query is empty, return none

    result = data.iloc[idx_array]['title']

    print('It took %s seconds to query forest.' %(time.time()-start_time))

    return result
```

In [ ]:

```
data = pd.DataFrame(headlines,columns= ['title'])
data.head()
```

Out[ ]:

	title
0	U.S. thanks Taiwan for support for auto chips ...
1	Robinhood lifts trading restrictions on all st...
2	PayPal says to shut domestic payments business...
3	Shares of Tencent-backed Kuaishou triple in HK...
4	Australian drone firm reshapes strategy over G...

In [ ]:

```
forest = get_forest(data, 100)
```

```
It took 0.1442406177520752 seconds to build forest.
```

Step 4: Predict news headlines recommendations for the given title.

In [ ]:

```
title = "Stocks explained: What's going on with GameStop?"
result = predict(title, data, 100, 10, forest)
print('\n Top Recommendation(s) is(are) \n', result)
```

```
It took 0.007291078567504883 seconds to query forest.

 Top Recommendation(s) is(are)
 1      Robinhood lifts trading restrictions on all st...
11    Wisconsin governor clashes with lawmakers over...
Name: title, dtype: object
```

In [ ]:

```
title = "What is this?"
result = predict(title, data, 100, 10, forest)
print('\n Top Recommendation(s) is(are) \n', result)
```

```
 Top Recommendation(s) is(are)
 None
```