**RV College of Engineering®**

**(Autonomous Institution Affiliated to VTU, Belagavi)**



# TITLE: GRAPH ALGORITHMS IN BIOINFORMATICS

Experiential Report

Submitted by

**Neha N - 1RV20CS094**

**Shubhprada K P – 1RV20CS163**

**Sravan Karthick T- 1RV20CS168**

**Ujwal P- 1RV20CS179**

**CSE A**

Submitted to

# Prof. Veena Gadad

**Introduction**

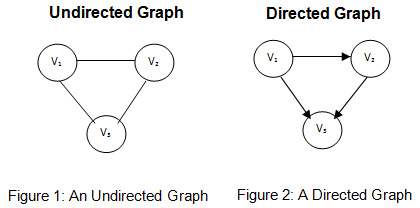
A genome is all the DNA in an organism including its genes. DNA is made up of 4 chemicals (abbreviated as A, C, T, G) that are repeated millions of times in a genome. The order of these species is very important as it determines what type of species the organism is.

Biologists are interested in genome sequencing because they can learn a lot by studying a genome. They can understand the function of a human gene by studying a similar gene in a fly or even in bacteria. There are numerous applications in medicine(genomes of fungi-producing bacteria), agriculture(oil palm genome), biotechnology(genomes of energy producing cyanobacteria), etc.

Sequencing technologies can read only a small amount of bases at a time and to read the whole genome, it takes a lot of attempts. As a result, we get subsets of DNA in millions. The process of reconstructing the original sequences is known as genome assembly. Software used in this task are known as Genome Assemblers. Genome assembly is a complex and challenging task.

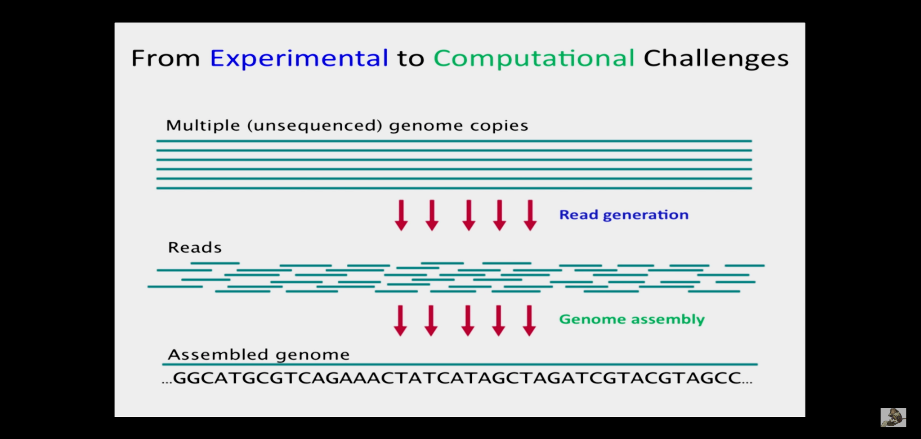
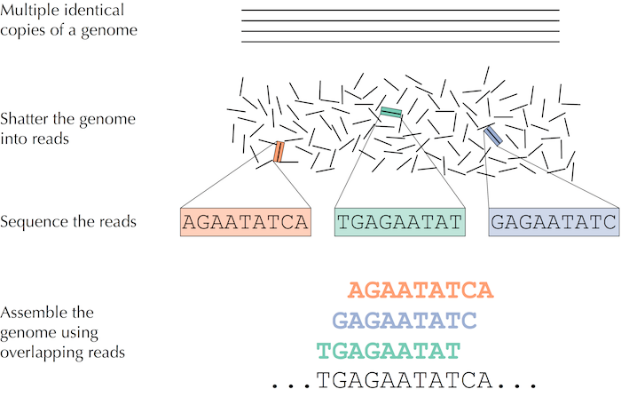
This task can be accomplished using graph algorithms.

A Graph is a collection V of vertices and a collection E of edges, each of which connects a pair of vertices. Graphs represent connections between objects. Graphs are of two types-directed graph and undirected graph.



To accomplish genome sequencing using graph algorithms, the concept of Eulerian path is used. An Eulerian path is a path through a directed graph that passes through all the nodes such that every edge is traversed only once. Small sequences called reads (k-mers) of a genome are depicted on a graph called the de-Bruijn graph and an Eulerian path through the de-Bruijn graph gives the required genome sequence.

We have tried to implement this in our work.

**Objectives**

1. To understand graph as a data structure and how it can be used in bioinformatics.
2. To understand the Eulerian algorithm and its applications.
3. To assemble a genome sequence given its paired k-mer composition using de Bruijn graphs
4. To implement the above in c++ and python.

**Literature Review**

1. Literature review 1

|  |  |
| --- | --- |
| **TITLE** | BIOINFORMATICS ALGORITHMS: AN ACTIVE LEARNING APPROACH, Chapter 3 |
| **AUTHOR** | Phillip Compeau, Pavel A. Pevzner |

|  |  |
| --- | --- |
| **SUMMARY** | **PUBLICATION DETAILS** |
| * Chapter 3 of this book is titled ‘How do we assemble genomes?’ * The book outlines the procedure of assembling genomes using deBruijn graphs and Eulerian path * It draws parallelism to other problems such as the Konigsberg bridges problem, and the exploding newspaper problem * It also talks about the real problems while using these algorithms and how they can be solved | TITLE: Bioinformatics Algorithms: An Active Learning Approach, Volume  AUTHORS: Phillip Compeau, Pavel Pevzner  EDITION: 2, illustrated  PUBLISHER: Active Learning Publishers, 2014  ISBN: 0990374602, 9780990374602 |

1. Literature review 2

|  |  |
| --- | --- |
| **TITLE** | Genome Assembly: A Review |
| **AUTHOR** | Arun Kumar, Vishal Verma |

|  |  |
| --- | --- |
| **SUMMARY** | **PUBLICATION DETAILS** |
| * This paper explains that after getting reads from sequencing technologies, how complex and challenging is the task of reconstructing the original genome. * It discusses De Bruijn and OLC graph-based solutions for assembling genome. * It discusses major challenges in genome assembly using these techniques. * It also discusses major error handling techniques. | Published in: 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)  Conference Location: Coimbatore, India  Date of Conference: 19-20 March 2021  Date Added to IEEE *Xplore*: 03 June 2021  Publisher: IEEE  Electronic ISSN: 2575-7288  Print on Demand (PoD) ISSN: 2469-5556 |

1. Literature review 3

|  |  |
| --- | --- |
| **TITLE** | [Enabling graph appliance for genome assembly](https://ieeexplore.ieee.org/document/7364056) |
| **AUTHOR** | R. Singh, J. A. Graves, S. Lee, S. R. Sukumar and M. Shankar |

|  |  |
| --- | --- |
| **SUMMARY** | **PUBLICATION DETAILS** |
| * This paper discusses the feasibility and scalability issues of de Bruijn graph assembly. * A de Bruijn graph represents a collection of read sequences by billions of vertices and edges, which require large amounts of memory and computational power to store and process. * This is a major drawback. It discusses how shared memory systems can be leveraged to overcome some of these issues | Published in: [2015 IEEE International Conference on Big Data (Big Data)](https://ieeexplore.ieee.org/xpl/conhome/7347101/proceeding)  Conference Location: Santa Clara, CA, USA  Date of Conference: 29 Oct.-1 Nov. 2015  Date Added to IEEE *Xplore*: 28 December 2015  Publisher: IEEE |

1. Literature review 4

|  |  |
| --- | --- |
| **TITLE** | De Bruijn Graph-Based Whole-Genomic Sequence Assembly Algorithms and Applications |
| **AUTHOR** | X. Kang, S. Tang, Y. Ma, R. Liu and Y. Wang |

|  |  |
| --- | --- |
| **SUMMARY** | **PUBLICATION DETAILS** |
| * This paper discusses several latest sequence assemblers and their algorithms. * It conducts a comparative study of some typical de Bruijn graph-based assemblers by stimulating the sequencing and assembly of Zebra fish genome data. | Published in: [2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing](https://ieeexplore.ieee.org/xpl/conhome/6679957/proceeding)  Conference Location: Beijing, China  Date of Conference: 20-23 Aug. 2013  Date Added to IEEE *Xplore*: 12 December 2013  Publisher: IEEE |

1. Literature review 5

|  |  |
| --- | --- |
| **TITLE** | Toward a more holistic method of genome assembly assessment |
| **AUTHOR** | Adam Thrash, Federico Hoffmann & Andy Perkins |

|  |  |
| --- | --- |
| **SUMMARY** | **PUBLICATION DETAILS** |
| * This paper discusses how to improve the efficiency of massive genome assembly process. * This paper also reports on the current state of assembly assessment methods, compares them, and provides a comprehensive method that encompasses several aspects of quality assessment. * The advantages and disadvantages of various methods are given and described, with recommendations based on analytical speed and user friendliness. | AUTHORS: Thrash, A., Hoffmann, F. & Perkins, A.  TITLE: Toward a more holistic method of genome assembly assessment.  Published in: BMC Bioinformatics 21, 249 (2020).  Published: 06 July 2020 |

1. Literature review 6

|  |  |
| --- | --- |
| **TITLE** | AN INTRODUCTION TO BIOINFORMATICS ALGORITHM, chapter 8 |
| **AUTHOR** | Neil C. Jones, Pavel A. Pevzner |

|  |  |
| --- | --- |
| **SUMMARY** | **PUBLICATION DETAILS** |
| * The book discusses graph and its application to solve various bioinformatics problems like DNA sequencing, protein sequencing etc * It discusses Eulerian and Hamiltonian algorithms and how they can be used in genomic sequencing | TITLE: An introduction to bioinformatics algorithm  AUTHORS: Neil C. Jones, Pavel A. Pevzner  PUBLISHER: Ane Books Pvt. Ltd  ISBN: 978-8180520785  PUBLICATION DATE: 1 January 2004 |

1. Literature review 7

|  |  |
| --- | --- |
| **TITLE** | Accelerating De Bruijn Graph-Based Genome Assembly for High-Throughput Short Read Data |
| **AUTHOR** | Kun Zhao; Weiguo Liu; Gerrit Voss; Wolfgang Mueller-Wittig |

|  |  |
| --- | --- |
| **SUMMARY** | **PUBLICATION DETAILS** |
| * The generated reads by emerging next-generation sequencing technologies are significantly shorter compared to the traditional Sanger shotgun sequencing method. * This poses challenges for de novo assembly algorithms in terms of both accuracy and efficiency. * In this paper, a scalable parallel algorithm to accelerate the de Bruijn graph-based genome assembly for high-throughput short read data is presented | Published in: [2013 International Conference on Parallel and Distributed Systems](https://ieeexplore.ieee.org/xpl/conhome/6754316/proceeding)  Date of Conference: 15-18 Dec. 2013  Date Added to IEEE *Xplore*: 01 May 2014  Electronic ISBN:978-1-4799-2081-5  Print ISSN: 1521-9097  INSPEC Accession Number: 14267954  DOI: [10.1109/ICPADS.2013.68](https://doi.org/10.1109/ICPADS.2013.68)  Publisher: IEEE  Conference Location: Seoul, Korea (South) |

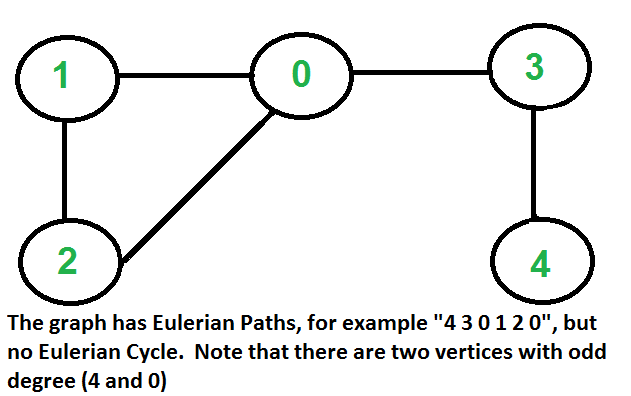
**Methodology**

As discussed above genome assembly using de-Bruijn graph involves constructing the de-Bruijn graph and traversing through the graph following the Eulerian path. First, we will discuss Eulerian path algorithm for any given directed graph and then apply to the de-Bruijn graph for genome assembly. The code for Eulerian path traversal given any directed graph is written in C++ followed by the code for genome assembly using de-Bruijn graph in Python.

**Part 1: Eulerian path given any directed graph**

Part A: Algorithm

Eulerian Path is a path in graph that visits every edge exactly once.



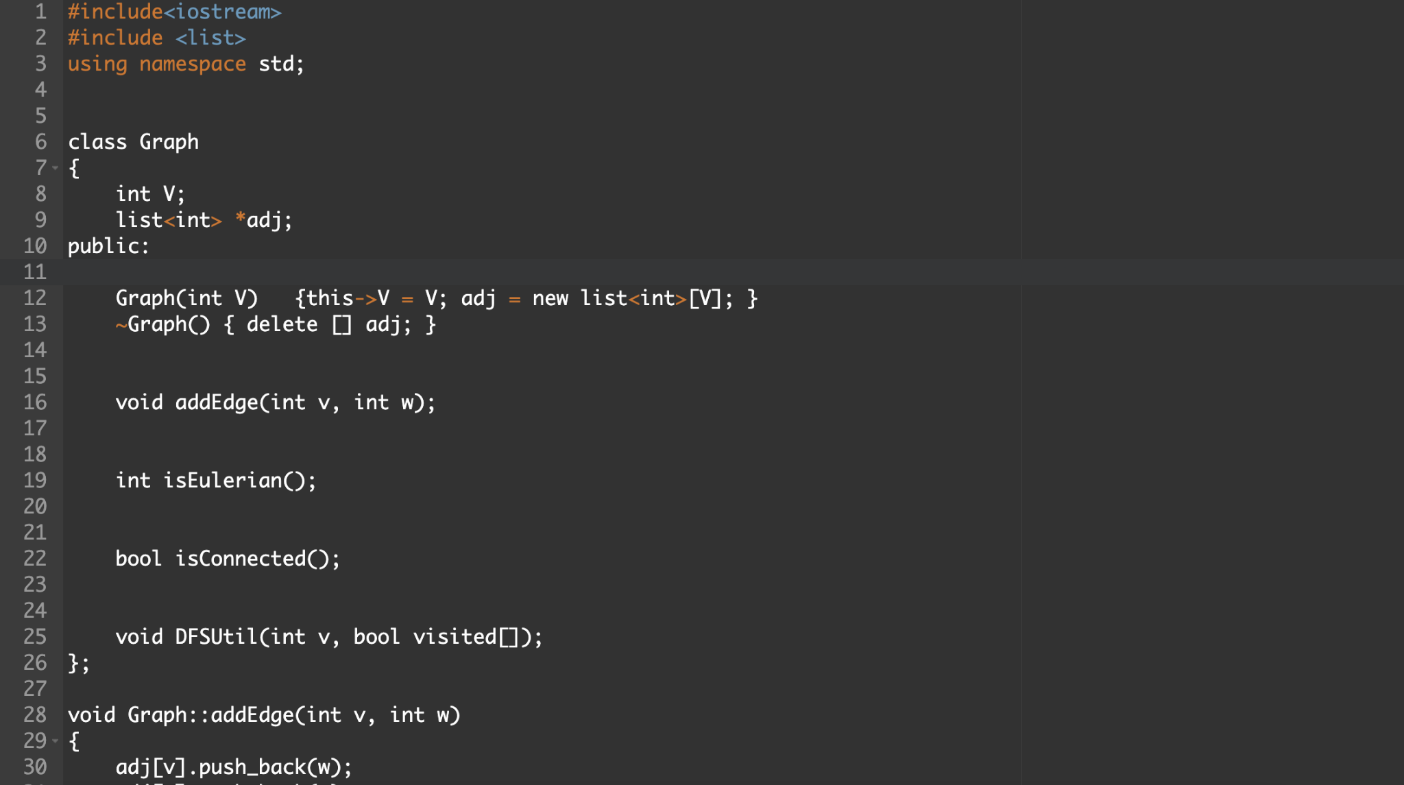
Conditions:

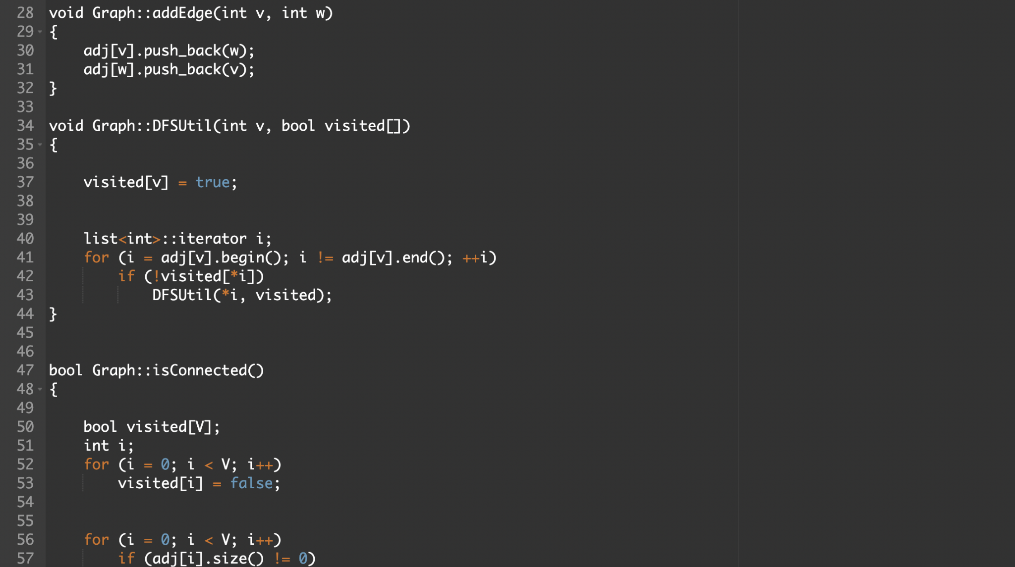
* every edge is visited once
* start vertex != end vertex
* exactly two vertices must have odd degree
* all vertices with non-zero degree are connected

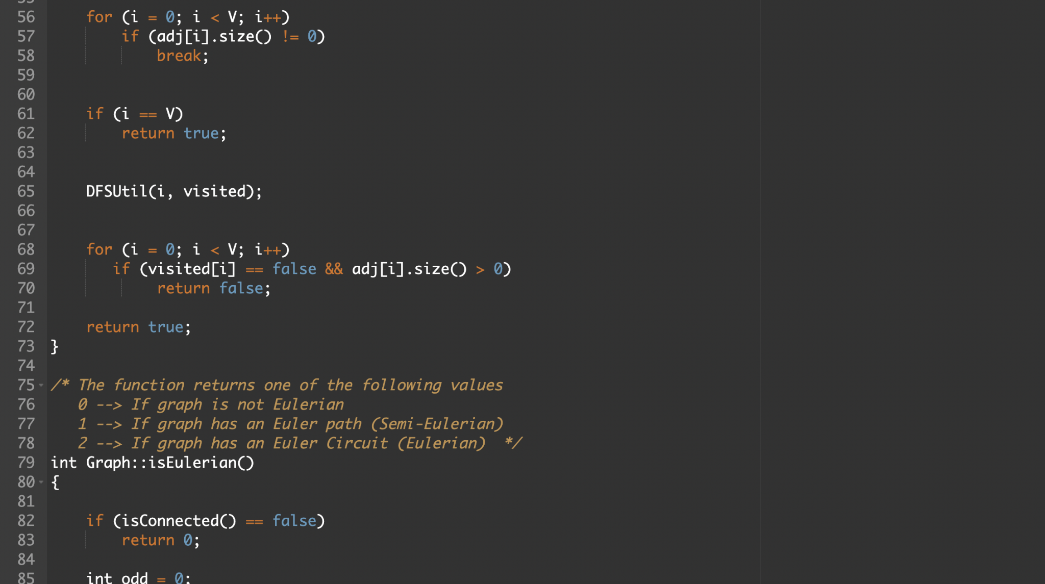
Connectivity Check: do DFS and mark all nodes in component. Check if any node with degree > 0 is unvisited ⇒ if true ⇒ not euler graph

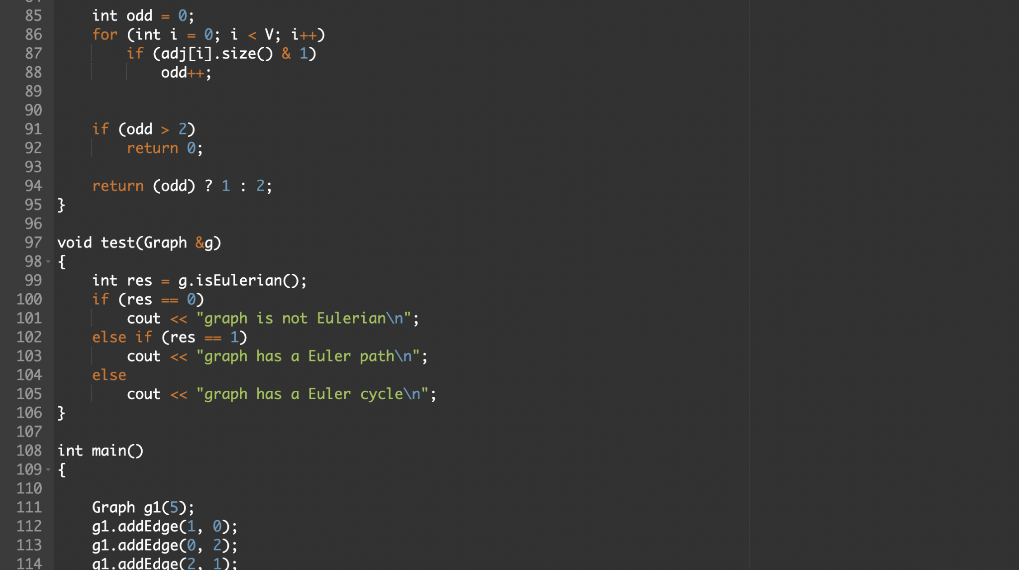
* Count odd degree vertices:
* count == 0 ⇒ eulerian graph
* count == 2 ⇒ euler path is present
* count  > 2 ⇒ non-eulerian graph

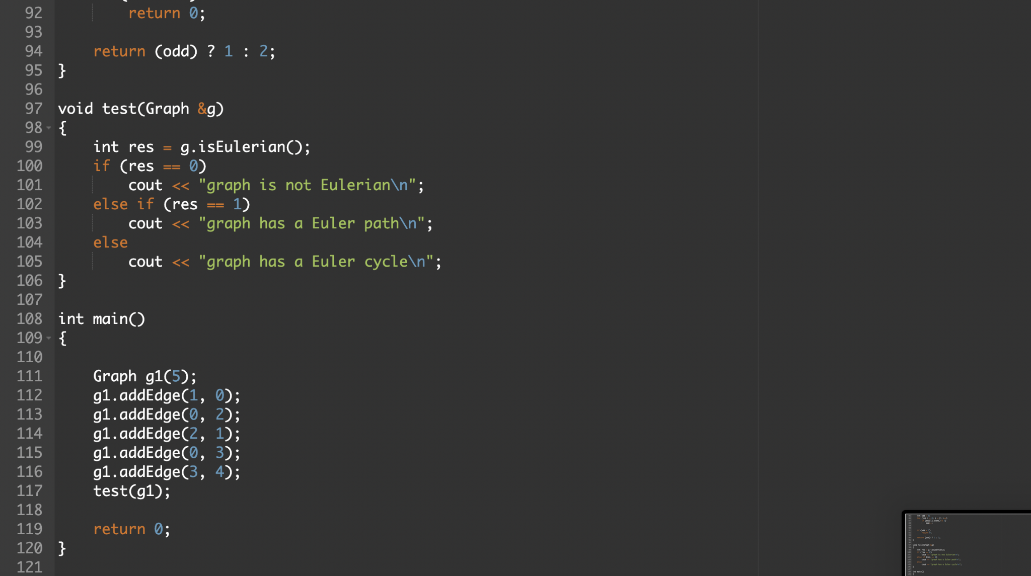
Part B: Code in C++











**Part 2: Construction of de-Bruijn graph + Eulerian path traversal of de-Bruijn graph**

Part A: Algorithm

Problem statement: Given a number of short reads (substrings of the genome of particular length) assemble the genome sequence using de Bruijn graph.

Sample input: 'ATGC', 'TGCT', 'GCTG', 'CTGC', 'TGCT'

Sample output: ATGCTGCT

Step 1: Start with a collection of reads as input, which are substrings of the reference genome.

          There are also called as k-mers where k is the length of each substring/read.

           ATGC

               TGCT

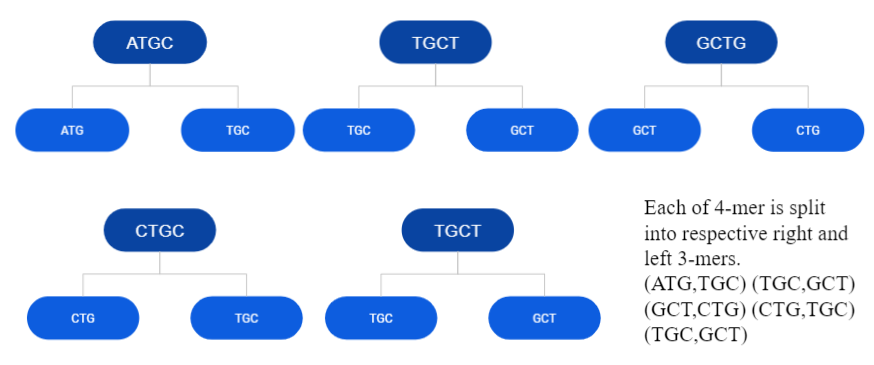
                  GCTG                            here, k =4

                     CTGC

                        TGCT

Step 2: Each k-mer input string is split into 2 overlapping substrings of length k-1. Each of

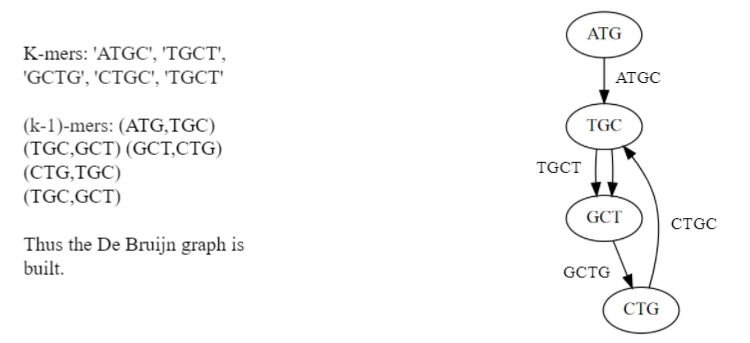
these are called the left (k-1)-mer and the right (k-1)mer.



Step 3: Each (k-1)-mer are acts as the nodes of the graph.(Add k-1 mers as nodes to De

Bruijn graph, if not already there). Draw a directed edge from each left (k-1)-mer to

corresponding right (k-1)-mer. Each edge in this graph is a k-mer.



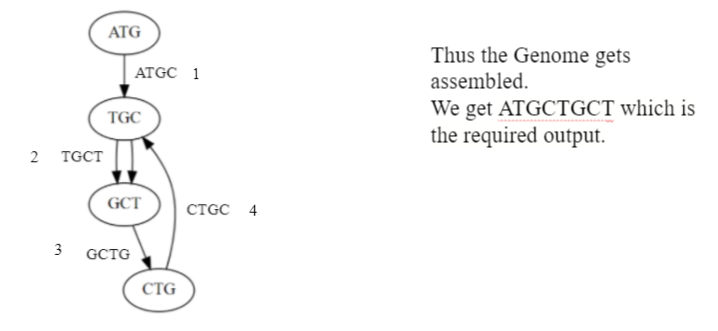
Step 4: Traverse through this graph following the Eulerian path.

1. Count the number of incoming and outgoing edges
2. Node having number of outgoing edges – incoming edges = 1 will be start node
3. Perform depth first search (dfs) recursively i.e travel from one node to another each time subtracting the number of outgoing edges until the number of outgoing edges of all nodes become 0
4. This way the graph is traversed passing through every edge once and only once

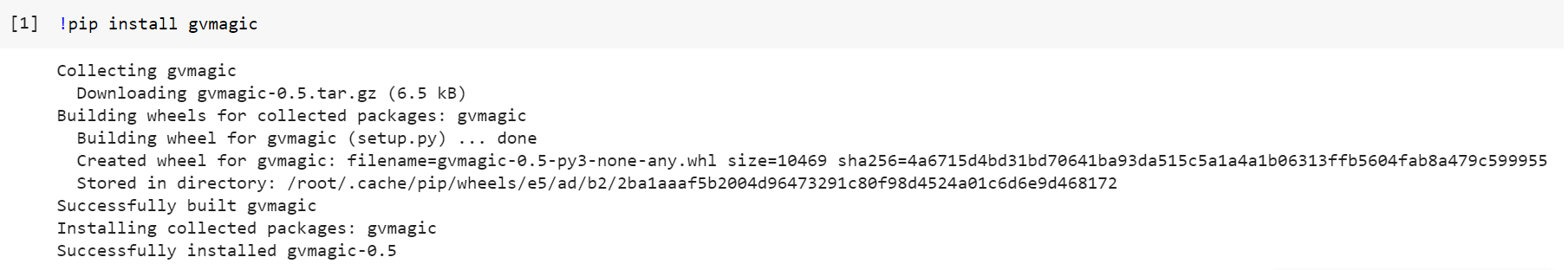
Step 5: Every time we reach a node while traversing the graph following the Eulerian path

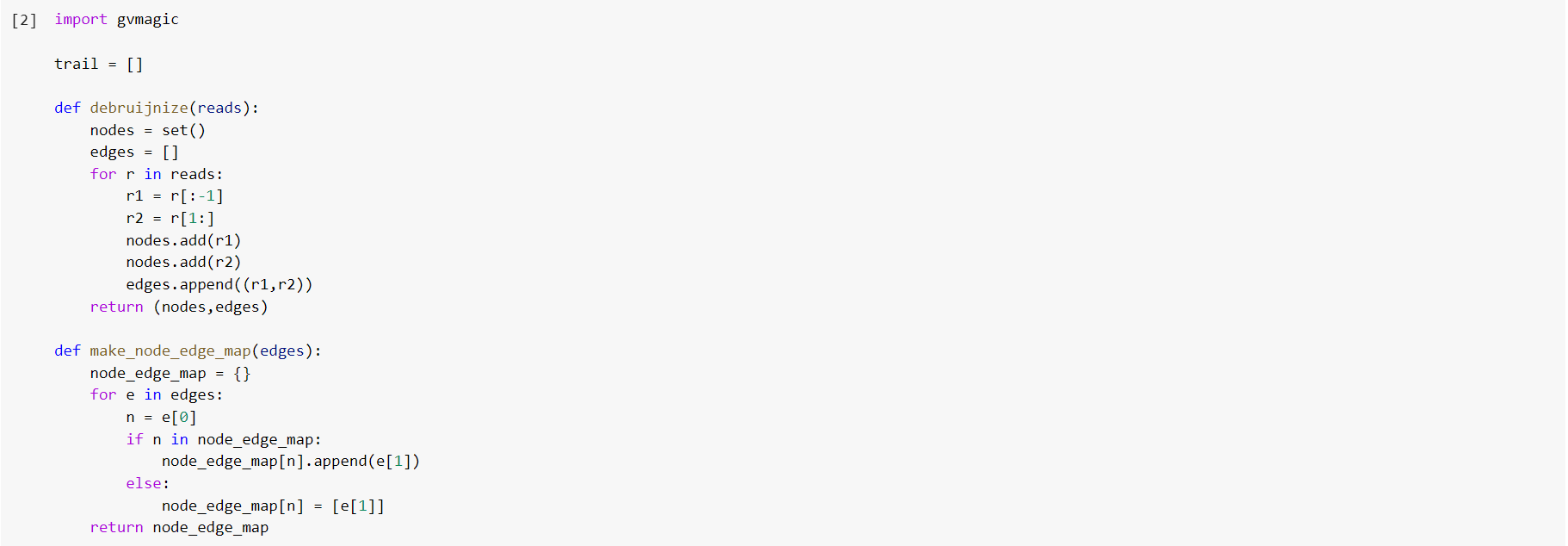
extract the last character and append it to the gene we are constructing. Once the

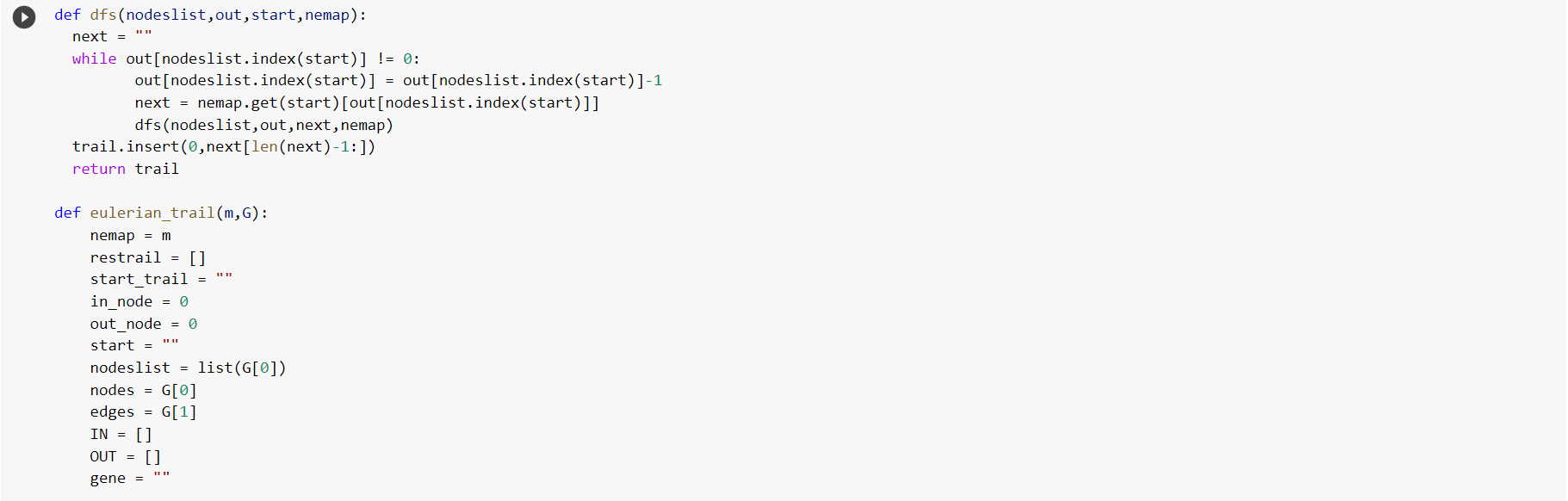
traversal and appending is done we obtain the output gene.



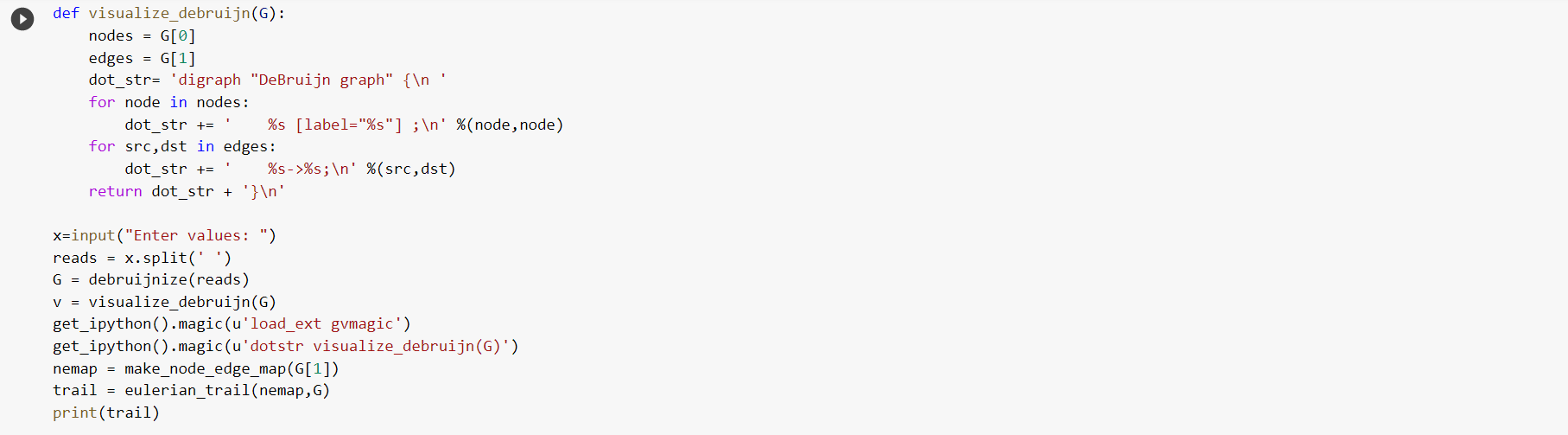
Part B: Code in python





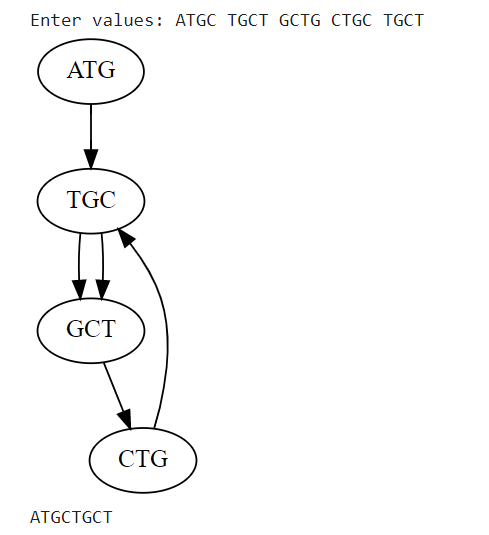
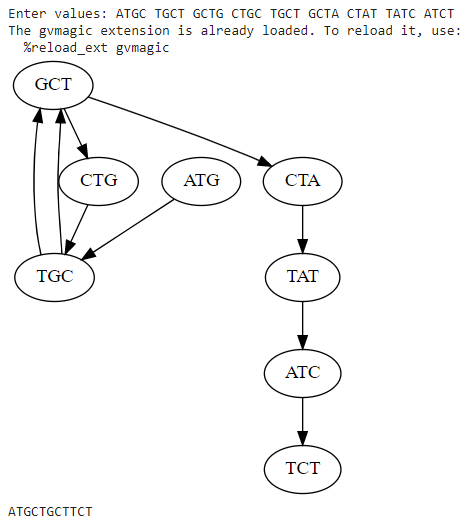






Part C: Sample input and output

Sample 1: Sample 2: Sample 3:

**Results and Discussion**

Graph theory is found to be a vital tool for solving biological problems. De Bruijn Graphs and Euler’s algorithm are the backbone of almost every technique used while assembling a genome. De Bruijn Graphs are found to be faster compared to other techniques and can also handle large inputs. As the De Bruijn graph represents a collection of read sequences by millions of vertices and edges, it requires large amounts of memory and computational power to store and process. It is one of the major drawbacks of de Bruijn graphs. Other drawbacks are that the reads are error-prone, and the Distance between reads within the read pairs is inexact.

De Bruijn graphs are not a cure-all. Yet for every apparent complication to sequence assembly, it has proven fruitful to apply some cousin of de Bruijn graphs to transform a question involving Eulerian cycles. Moreover, analogs of de Bruijn graphs have been useful in many other bioinformatics problems, including antibody sequencing, synteny block reconstruction, and RNA assembly11.

In each of these applications, the de Bruijn graph represents the experimental data in a manner that leads to a tractable computational problem. As new sequencing technologies emerge, the best computational strategies for assembling genomes from reads may change. The factors that influence the choice of algorithms include the quantity of data (measured by read length and coverage); quality of data (including error rates); and genome structure (e.g., GC content and the number and size of repeated regions). Short-read sequencing technologies produce very large numbers of reads, which currently favour the use of de Bruijn graphs.

De Bruijn graphs are also well suited to representing genomes with repeats, whereas overlap methods need to mask repeats that are longer than the read length. However, if a future sequencing technology produces high-quality reads with tens of thousands of bases, a smaller number of reads would be needed, and the pendulum could swing back toward favouring overlap-based approaches for assembly. Yet few things like high accuracy, less memory consumption and less time consumption in the assembly are few required aspects that are primarily needed for assembling a genome. In the case of large genomes like humans, it has become even more needed as time and memory are consumed a lot.

In future genomes, tens of thousands of species will be sequenced. There are enormous efforts to sequence cancer genomes and understand what changes in the genome lead to cancer.

Thousands of cancer genomes have already been sequenced, and genome sequencing will soon become a routine technique in medicine.

**References**

1. A. Kumar and V. Verma, "Genome Assembly: A Review," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 494-498, doi: 10.1109/ICACCS51430.2021.9441878.
2. R. Singh, J. A. Graves, S. Lee, S. R. Sukumar and M. Shankar, "Enabling graph appliance for genome assembly," 2015 IEEE International Conference on Big Data (Big Data), 2015, pp. 2583-2590, doi: 10.1109/BigData.2015.7364056.
3. X. Kang, S. Tang, Y. Ma, R. Liu and Y. Wang, "De Bruijn Graph-Based Whole-Genomic Sequence Assembly Algorithms and Applications," 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 2013, pp. 2094-2097, doi: 10.1109/GreenCom-iThings-CPSCom.2013.393.
4. K. Zhao, W. Liu, G. Voss and W. Mueller-Wittig, "Accelerating De Bruijn Graph-Based Genome Assembly for High-Throughput Short Read Data," 2013 International Conference on Parallel and Distributed Systems, 2013, pp. 426-427, doi: 10.1109/ICPADS.2013.68.
5. <https://www.geeksforgeeks.org/eulerian-path-and-circuit/>
6. <https://en.wikipedia.org/wiki/Eulerian_path>
7. <https://www.geeksforgeeks.org/hamiltonian-cycle-backtracking-6/>
8. <https://www.bioinformaticsalgorithms.org/bioinformatics-chapter-3>
9. <https://slaystudy.com/hierholzers-algorithm/>
10. <https://www.cs.jhu.edu/~langmea/resources/lecture_notes/assembly_dbg.pdf>