Joseph Ville
CSCI-351-01
Report - Programming Project 2

1. Each of my programs takes in the name of a graph file in Graph File Format at the command line. The graph is an Arxiv High Energy Physics Theory (HEP-TH) collaboration network from the Stanford Large Network Dataset Collection.

   `Regression` performs two linear regressions on the degree distribution of the HEP-TH graph: one fitted to an exponential function, and one fitted to a power function. It then gives me a table of the Degree *d* and Probability *pr(d)*. `Regression` then outputs the respective equations for *log pr(d)* and *pr(d)* and correlation coefficients for both linear regressions.

   `ConnectedComponents` finds all the connected components of the HEP-TH graph. It outputs the total number of connected components, then a table of each component's number and the number of vertices in that component. Next, it prints the number of each component, and the number of each vertex in that component. Finally, it tells me the size of the smallest and largest components, and the percentage of the graph's vertices that are contained in the largest component.

   `TopRank` finds the top-40-ranked vertices of the HEP-TH graph, with respect to both degree centrality and closeness centrality, and outputs the results.

2. `Regression.java` uses classes from the Parallel Java 2 Library, so the classpath must be set accordingly before this code can be compiled.

   Once the files have been compiled, the command lines to run my programs are as follows:

```
java Regression <fileName>
where <filename> = the name of a file in Graph File Format

java ConnectedComponents <fileName>
where <filename> = the name of a file in Graph File Format

java TopRank <fileName>
where <filename> = the name of a file in Graph File Format
```

3.

```
import java.math.BigDecimal;
import java.math.MathContext;
import java.text.DecimalFormat;
import java.util.TreeMap;
```

```
import edu.rit.numeric.ListXYSeries;
import edu.rit.numeric.XYSeries;

/**
 * Perform regressions on a graph, and analyze the results
 * @author Joseph Ville
 *
 * Usage: java Regression <fileName>
 *                    <fileName> = the name of the graph file to be analyzed
 *
 */
public class Regression
{
	private static Collaboration collab;
	private static String fileName;
	private static Vertex[] vertices;
	private static XYSeries.Regression expReg;
	private static XYSeries.Regression powReg;

	/**
	 * Main method for this program
	 * @param args - the command line arguments
	 */
	public static void main(String[] args)
	{
		if(args.length != 1)
		{
			usage();
		}
		fileName = args[0];
		collab = new Collaboration();
		vertices = collab.readFile(fileName);
		TreeMap<Integer, Integer> occurrences = degreeOccurrences(vertices);

		executeRegressions(occurrences);

		MathContext mathContext = new MathContext(5);

		// power
		double cP = Math.exp(powReg.a);
		System.out.print("log pr(d) = " + new BigDecimal(Math.log(cP),
mathContext) + " + " + new BigDecimal(powReg.b, mathContext) + " log d");
```

```
                System.out.println(", corr = " + new BigDecimal(powReg.corr,
mathContext));
                System.out.println("pr(d) = " + new BigDecimal(cP, mathContext) + " d^" +
new BigDecimal(powReg.b, mathContext));

                // exponential
                double cE = Math.exp(expReg.a);
                double d = Math.exp(expReg.b);
                System.out.print("log pr(d) = " + new BigDecimal(Math.log(cE),
mathContext) + " + " + new BigDecimal(Math.log(d), mathContext) + " d");
                System.out.println(", corr = " + new BigDecimal(expReg.corr,
mathContext));
                System.out.println("pr(d) = " + new BigDecimal(cE, mathContext) + " * " +
new BigDecimal(d, mathContext) + "^d");
        }

        /**
         * Performs linear regressions on an exponential function and a power function
         * @param degreeCounts - map of degrees to the number of occurrences
         */
        public static void executeRegressions(TreeMap<Integer, Integer> degreeCounts)
        {
                ListXYSeries expXYSeries = new ListXYSeries();
                ListXYSeries powXYSeries = new ListXYSeries();
                DecimalFormat d = new DecimalFormat("#");
                DecimalFormat d1 = new DecimalFormat("0.00000E00");
                System.out.println("d\tcount\tpr");
                for(Integer key : degreeCounts.keySet())
                {
                        double probD = (double)degreeCounts.get(key) / collab.V();
                        System.out.println(key + "\t" + d.format(degreeCounts.get(key)) +
"\t" + sFormat(d1.format(probD).toString()));
                        expXYSeries.add(key, Math.log(probD));
                        powXYSeries.add(Math.log(key), Math.log(probD));
                }
                expReg = expXYSeries.linearRegression();
                powReg = powXYSeries.linearRegression();
        }

        /**
         * Find the number of occurrences of each degree in an of an
         * array of vertices
         * @param vertices
         */
```

```
	public static TreeMap<Integer, Integer> degreeOccurrences(Vertex[] vertices)
	{
		// Using a TreeMap so it will be ordered automatically by vertex degree
		TreeMap<Integer, Integer> occurrences = new TreeMap<Integer,
Integer>();

		for(Vertex v : vertices)
		{
			if(occurrences.containsKey(v.degree()))
			{
				int value = occurrences.get(v.degree());
				occurrences.put(v.degree(), ++value);
			}
			else
			{
				occurrences.put(v.degree(), 1);
			}
		}
		return occurrences;
	}

	/**
	 * Format the given string so that "E"s are "e"s
	 * @param input - the given string
	 * @return the formatted string
	 */
	public static String sFormat(String input)
	{
		return input.replaceAll("E", "e");
	}

	/**
	 * Print a usage message and exit.
	 */
	public static void usage()
	{
		System.err.println("Usage: java Regression <fileName>\n" +
				"<fileName> = the name of the graph file to be analyzed");
		System.exit(0);
	}
}


import java.text.DecimalFormat;
```

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.LinkedList;
import java.util.TreeMap;

/**
 * Find the connected components of a graph
 * @author Joseph Ville
 *
 * Usage: java ConnectedComponents <fileName>
 *                    <fileName> the name of the graph file to be analyzed
 *
 */
public class ConnectedComponents
{
	private static String fileName;
	private static int smallestCCSize;
	private static int largestCCSize;
	private static int largestCC;
	private boolean printData;
	private static TreeMap<Integer, ArrayList<Integer>> ccIndexes;

	/**
	 * Default constructor
	 */
	public ConnectedComponents()
	{
		smallestCCSize = Integer.MAX_VALUE;
		largestCCSize = Integer.MIN_VALUE;
	}

	/**
	 * Construct an object of this class
	 * @param printResults
	 */
	public ConnectedComponents(boolean printData)
	{
		this.printData = printData;
		smallestCCSize = Integer.MAX_VALUE;
		largestCCSize = Integer.MIN_VALUE;
	}

	/**
	 * Main method for this class
```

```
	 * @param args - the command line arguments
	 */
	public static void main(String[] args)
	{
		if(args.length != 1)
		{
			usage();
		}
		fileName = args[0];

		Collaboration collab = new Collaboration();
		Vertex[] vertices = collab.readFile(fileName);
		ConnectedComponents cc = new ConnectedComponents(true);

		cc.findComponents(vertices);
		DecimalFormat df = new DecimalFormat("0.#####");
		System.out.println("Size of smallest CC = " + smallestCCSize);
		System.out.println("Size of largest CC = " + largestCCSize);
		System.out.println(largestCCSize + "/" + collab.V() + " = " +
df.format((double)largestCCSize / collab.V()));
	}

	/**
	 * Find all connected components of an array of vertices
	 * @param vertices - the array of vertices
	 * @return the total number of connected components in the graph
	 */
	public int findComponents(Vertex[] vertices)
	{
		LinkedList<Integer> unvisited = new LinkedList<Integer>();
		LinkedList<Integer> queue = new LinkedList<Integer>();
		ccIndexes = new TreeMap<Integer, ArrayList<Integer>>();
		ArrayList<Integer> components;

		for(int a = 0; a < vertices.length; a++)
		{
			unvisited.add(a);
		}

		String outString = "connected components\nComp\tSize\n";
		int i = -1, m = -1, n = 0;
		while(unvisited.size() > 0)
		{
			i = unvisited.poll();
```

```
                queue.add(i);

                components = new ArrayList<Integer>();
                components.add(i);
                while(queue.size() > 0)
                {
                        m = queue.poll();

                        for(Integer neighbor : vertices[m].getNeighbors())
                        {
                                if(unvisited.contains(neighbor))
                                {
                                        unvisited.remove(neighbor);
                                        queue.add(neighbor);
                                        components.add(neighbor);
                                }
                        }
                }

                if(components.size() > largestCCSize)
                {
                        largestCC = n;
                        largestCCSize = components.size();
                }
                if(components.size() < smallestCCSize)
                {
                        smallestCCSize = components.size();
                }
                outString = outString.concat(n + "\t" + components.size() + "\n");
                Collections.sort(components);
                ccIndexes.put(n++, components);
        }

        if(printData)
        {
                System.out.print(n + " " + outString);
                printFormat(ccIndexes);
        }
        return n;
}

/**
 * Print the CCs and their associated indexes in a readable format
 * @param ccIndexes - a map of indexes to their CCs
```

```
	 */
	public static void printFormat(TreeMap<Integer, ArrayList<Integer>> ccIndexes)
	{
		for(int i = 0; i < ccIndexes.size(); i++)
		{
			System.out.printf("Comp %d = ", i);
			for(int j = 0; j < ccIndexes.get(i).size(); j++)
			{
				System.out.printf("%d ", ccIndexes.get(i).get(j));
			}
			System.out.println();
		}
	}

	/**
	 * @return the largest connected component
	 */
	public ArrayList<Integer> getLargestComponent()
	{
		return ccIndexes.get(largestCC);
	}

	/**
	 * Print a usage message and exit
	 */
	public static void usage()
	{
		System.err.println("Usage: java ConnectedComponents <fileName>\n" +
				"<fileName> = the name of the graph file to be analyzed");
		System.exit(0);
	}
}// end class ConnectedComponents


import java.util.AbstractMap.SimpleEntry;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.LinkedList;
import java.util.List;

/**
 * Analyze a graph and make conclusions about the top-40-ranked vertices
 * @author Joseph Ville
```

```
 *
 * Usage: java TopRank <fileName>
 *                    <fileName> = the name of the graph file to be analyzed
 */
public class TopRank
{
	private static String fileName;
	private static Vertex[] vertices;

	/**
	 * Main method for this program
	 * @param args - the command line arguments
	 */
	public static void main(String[] args)
	{
		if(args.length != 1)
		{
			usage();
		}
		fileName = args[0];

		Collaboration collab = new Collaboration();
		ConnectedComponents cc = new ConnectedComponents(false);
		vertices = collab.readFile(fileName);
		int n = cc.findComponents(vertices);
		ArrayList<Integer> largestCC = cc.getLargestComponent();

		int vertex;
		List<SimpleEntry<Integer, Integer>> cloCent = degreeCent(largestCC);
		System.out.println("Rank\tVertex\tDegCen");
		for(int i = 0; i < 40; i++)
		{
			vertex = largestCC.get(i);
			System.out.println((i+1) + "\t" + cloCent.get(vertex).getKey() + "\t"
+ cloCent.get(vertex).getValue());
		}

		List<SimpleEntry<Integer, Float>> avgDistances =
closenessCent(largestCC);
		System.out.println("Rank\tVertex\tCloCen");
		for(int i = 0; i < 40; i++)
		{
			vertex = largestCC.get(i);
```

```
                        System.out.println((i+1) + "\t" + avgDistances.get(vertex).getKey()
+ "\t" + avgDistances.get(vertex).getValue());
                }
        }

        /**
         * Compute the closeness centrality of the given connected component
         * @param cc - the connected component
         * @return - an ArrayList<SimpleEntry<Integer, Float>> which stores the vertex
number and its
         *                  associated closeness centrality
         */
        public static List<SimpleEntry<Integer, Float>>
closenessCent(ArrayList<Integer> cc)
        {
                List<SimpleEntry<Integer, Float>> avgDistances = new
ArrayList<SimpleEntry<Integer, Float>>(cc.size());
                int current = 0;
                for(int i = 0; i < cc.size(); i++)
                {
                        current = cc.get(i);
                        avgDistances.add(new SimpleEntry<Integer, Float>(current,
avgDistance(current)));
                }

                // sort the distances
                Collections.sort(avgDistances, new Comparator<SimpleEntry<Integer,
Float>>(){
                                @Override
                                public int compare(SimpleEntry<Integer, Float> arg0,
SimpleEntry<Integer, Float> arg1)
                                {
                                        return
arg0.getValue().compareTo(arg1.getValue());
                                }
                        });
                return avgDistances;
        }

        /**
         * Compute the degree centrality of the given connected component
         * @param cc - the connected component
         * @return an ArrayList<SimpleEntry<Integer, Integer>> which stores the vertex
number and its
```

```
	 *		associated degree centrality
	 */
	public static List<SimpleEntry<Integer, Integer>> degreeCent(ArrayList<Integer>
cc)
	{
		List<SimpleEntry<Integer, Integer>> degrees = new
ArrayList<SimpleEntry<Integer, Integer>>(cc.size());
		int current = 0;
		for(int v = 0; v < cc.size(); v++)
		{
			current = cc.get(v);
			degrees.add(new SimpleEntry<Integer, Integer>(current,
vertices[current].degree()));
		}
		Collections.sort(degrees, new Comparator<SimpleEntry<Integer,
Integer>>(){
			@Override
			public int compare(SimpleEntry<Integer, Integer> arg0,
SimpleEntry<Integer, Integer> arg1)
			{
				return arg1.getValue().compareTo(arg0.getValue());
			}
		});
		return degrees;
	}

	/**
	 * Compute the average of the distances between the given vertex and every
other vertex
	 * @param vertex - the vertex to find distances from
	 * @return the average distance from this vertex to every other vertex
	 */
	public static float avgDistance(int vertex)
	{
		LinkedList<Integer> queue = new LinkedList<Integer>(); // queue of
vertex indices
		boolean[] seen = new boolean[	vertices.length]; // whether this
vertex has been seen
		int[] distances = new int[vertices.length]; // distances from this to all other
vertices

		for(int i = 0; i < vertices.length; i++) // initialize the arrays
		{
			distances[i] = Integer.MAX_VALUE;
```

```
                seen[i] = false;
            }

            seen[vertex] = true;

            queue.add(vertex);
            int a = 0;
            int count = 0;
            distances[vertex] = 0;

            while(queue.size() != 0)
            {
                a = queue.poll(); // remove the head of the queue
                for(Integer b : vertices[a].getNeighbors()) // loop through all
neighbors of current vertex
                {
                    if(!seen[b])
                    {
                        seen[b] = true;
                        queue.add(b); // add b to end of queue
                        count++;
                        distances[b] = distances[a] + 1;
                    } // end if
                } // end for
            } // end while

            int sum = 0;
            for(int i = 0; i < vertices.length; i++)
            {
                if(distances[i] != Integer.MAX_VALUE)
                {
                    sum += distances[i];
                }
            }
            return (float) sum / count;
        }

        /**
         * Print a usage message and exit
         */
        public static void usage()
        {
            System.err.println("Usage: java TopRank <fileName>\n" +
                    "<fileName> = the name of the graph file to be analyzed");
```

```
                System.exit(0);
        }
}


import java.io.FileReader;
import java.io.BufferedReader;

/**
 * Analyzes a Collaboration Graph
 *
 * @author Joseph Ville
 */
public class Collaboration
{
        private int V;
        private int E;
        private Vertex[] vertices;

        /**
         * Construct an object of this class
         */
        public Collaboration()
        {
        }

        /**
         * Reads and processes a file in the Graph File Format
         * @param fileName - the file to process
         * @return vertices - an array of vertices for the graph
         * @throws Exception
         */
        public Vertex[] readFile(String fileName)
        {
                String line = "";

                try
                {
                        BufferedReader buff = new BufferedReader(new
FileReader(fileName));

                        boolean gLine = false;
                        boolean eLine = false;
```

```
                        while((line = buff.readLine()) != null)
                        {
                                String[] lineArr;

                                // ignore any blank line
                                if(!line.equals(null) && !line.isEmpty())
                                {
                                        lineArr = line.split(" ");

                                        /* Required. Edges of the graph.
                                        First field - source vertex #, 0 <= an int <= V-1
                                        Second field - destination vertex #, 0 <= an int <=
V-1
                                        Third field - edge weight, a floating pt #. If 3rd field
omitted,
                                                assume to be 1 by default.
                                        */
                                        if(lineArr[0].equals("e"))
                                        {
                                                eLine = true;
                                                int index = Integer.parseInt(lineArr[1]);
                                                int neighbor = Integer.parseInt(lineArr[2]);

                                                initializeIfNull(vertices, index);
                                                initializeIfNull(vertices, neighbor);
                                                vertices[index].addNeighbor(neighbor);
                                                vertices[neighbor].addNeighbor(index);
                                        }
                                        /* Required. Occurs once, at beginning of file
                                        The parameters of the graph.
                                        First field is # of vertices V, an int >= 0
                                        Second field is # of edges E, an int >= 0
                                        */
                                        else if(lineArr[0].equals("g"))
                                        {
                                                gLine = true;
                                                int numVertices =
Integer.parseInt(lineArr[1]); // V and E are guaranteed to be >= 0
                                                int numEdges = Integer.parseInt(lineArr[2]);

                                                /* ok to do this here, because this will be
executed before
                                                any of the other if statements on this or any
other
```

```
                                        pass of the while loop */
                                        this.V = numVertices;
                                        this.E = numEdges;
                                        vertices = new Vertex[V];
                                }
                                // ignore any lines beginning with v, c, or d
                                else if(lineArr[0].equals("v"))
                                {
                                        continue;
                                }
                                else if(lineArr[0].equals("d"))
                                {
                                        continue;
                                }
                                else if(lineArr[0].equals("c"))
                                {
                                        continue;
                                }
                        }// end if
                }// end while

                buff.close();

                // make sure the file contained the required lines
                if(gLine == false || eLine == false)
                {
                        System.err.println("The file is in an invalid format.");
                        System.exit(0);
                }
        }
        catch(NumberFormatException nfe)
        {
                System.err.println("A number in the file had invalid format");
                System.exit(0);
        }
        catch(Exception ex)
        {
                System.err.println("There was an error reading the file");
                System.exit(0);
        }
        return vertices;
}// end readFile()

/**
```

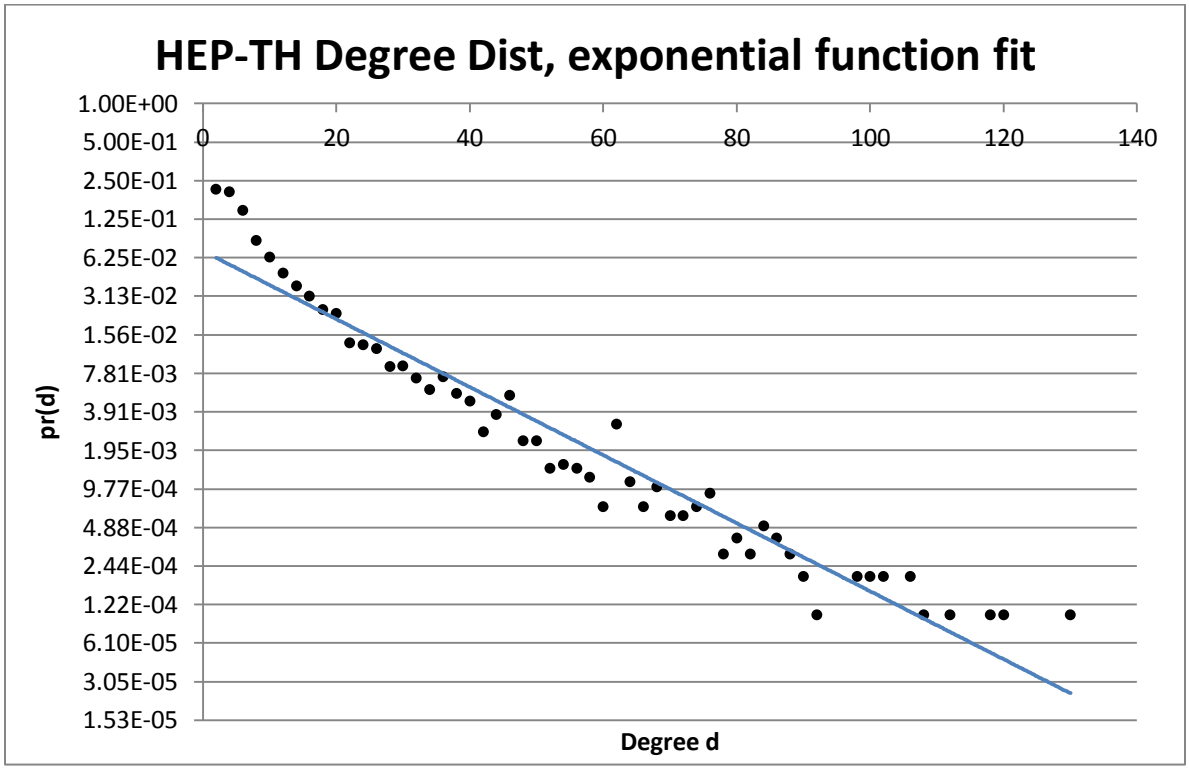
```
	 * @return the number of vertices
	 */
	public int V()
	{
		return V;
	}

	/**
	 * @return the number of edges
	 */
	public int E()
	{
		return E;
	}

	/**
	 * Check if a vertex is null, and if it is, initialize it.
	 * @param v - the vertex to check
	 */
	private void initializeIfNull(Vertex[] vertices, int v)
	{
		if(vertices[v] == null)
		{
			vertices[v] = new Vertex();
		}
	}
}// end class Collaboration


import java.util.ArrayList;

/**
 * A class to store all the attributes of a vertex object
 * @author Joseph Ville
 */
public class Vertex
{
	private ArrayList<Integer> neighbors; // neighbors of the current vertex

	/**
	 * Construct an object of this class, by setting
	 * the class variables to their default values
	 */
	public Vertex()
```

```
	{
		neighbors = new ArrayList<Integer>();
	}

	/**
	 * Find the degree of this vertex. Will only work after
	 * neighbors list has finished populating.
	 * @return the degree of this vertex
	 */
	public int degree()
	{
		return neighbors.size();
	}

	/**
	 * Get the list of vertices that are adjacent to the current vertex
	 * @return the list of neighbors
	 */
	public ArrayList<Integer> getNeighbors()
	{
		return neighbors;
	}

	/**
	 * Appends a vertex number to the end of this vertex's neighbor list
	 * @param n - the neighbor to be added
	 */
	public void addNeighbor(int n)
	{
		this.neighbors.add(n);
	}
}// end class Vertex
```

4. Command line: java Regression CA-HepTh-graph.txt

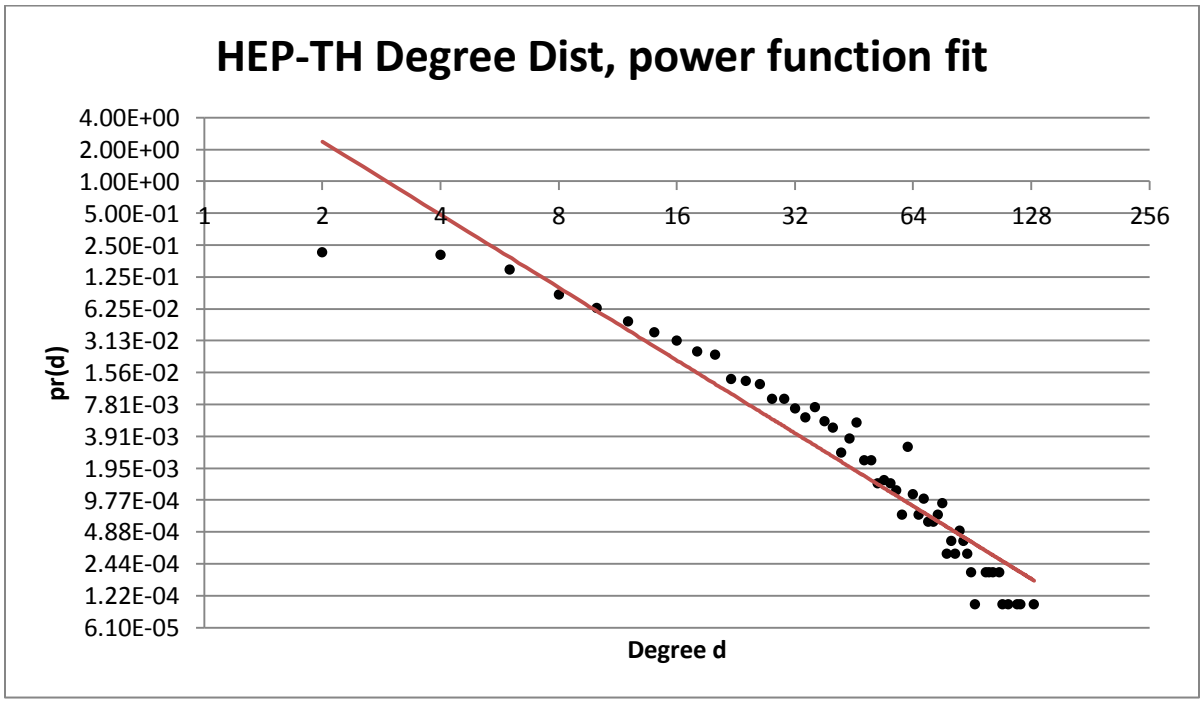
d	count	pr
2	2110	2.13628E-01
4	2015	2.04009E-01
6	1446	1.46401E-01
8	847	8.57548E-02
10	628	6.35821E-02
12	468	4.73828E-02
14	371	3.75620E-02
16	311	3.14873E-02
18	244	2.47039E-02

20	227	2.29827E-02
22	134	1.35669E-02
24	129	1.30606E-02
26	120	1.21494E-02
28	87	8.80834E-03
30	88	8.90959E-03
32	71	7.18842E-03
34	58	5.87223E-03
36	73	7.39091E-03
38	54	5.46725E-03
40	47	4.75853E-03
42	27	2.73362E-03
44	37	3.74608E-03
46	52	5.26476E-03
48	23	2.32864E-03
50	23	2.32864E-03
52	14	1.41743E-03
54	15	1.51868E-03
56	14	1.41743E-03
58	12	1.21494E-03
60	7	7.08717E-04
62	31	3.13860E-03
64	11	1.11370E-03
66	7	7.08717E-04
68	10	1.01245E-03
70	6	6.07472E-04
72	6	6.07472E-04
74	7	7.08717E-04
76	9	9.11208E-04
78	3	3.03736E-04
80	4	4.04981E-04
82	3	3.03736E-04
84	5	5.06227E-04
86	4	4.04981E-04
88	3	3.03736E-04
90	2	2.02491E-04
92	1	1.01245E-04
98	2	2.02491E-04
100	2	2.02491E-04
102	2	2.02491E-04
106	2	2.02491E-04
108	1	1.01245E-04
112	1	1.01245E-04
118	1	1.01245E-04
120	1	1.01245E-04
130	1	1.01245E-04



$\mathrm{pr}(d) = 0.070479 * 0.94064^{d}$

Correlation = -0.96800



$\mathrm{pr}(d) = 11.529\ d^{-2.2857}$

Correlation = -0.95723

The exponential function fit has a stronger correlation coefficient, therefore the HEP-TH graph more closely resemble a small-world graph.

The HepTh graph more closely resembles a small-world graph, because the exponential function has the stronger correlation coeffiecient.

5. Command line: java ConnectedComponents CA-HepTh-graph.txt

429 connected components

Comp	Size
0	8638
1	2
2	2
3	2
4	7
5	2
6	2
7	2
8	2
9	2
10	2
11	4
12	3
13	2
14	2
15	5
16	7
17	2
18	4
19	2
20	2
21	4
22	2
23	3
24	11
25	2
26	3
27	2
28	2
29	5
30	4
31	6
32	7
33	2

34 2
35 3
36 2
37 3
38 8
39 2
40 2
41 2
42 2
43 2
44 7
45 2
46 2
47 6
48 5
49 4
50 2
51 2
52 5
53 4
54 4
55 5
56 2
57 2
58 2
59 2
60 2
61 3
62 2
63 2
64 2
65 2
66 8
67 5
68 5
69 2
70 2
71 5
72 3
73 2
74 4
75 2
76 2
77 8

78 2
79 4
80 5
81 2
82 2
83 2
84 2
85 2
86 2
87 3
88 2
89 3
90 7
91 2
92 2
93 2
94 8
95 2
96 4
97 8
98 2
99 2
100 6
101 2
102 2
103 3
104 2
105 2
106 7
107 2
108 3
109 2
110 2
111 3
112 5
113 6
114 2
115 3
116 3
117 2
118 3
119 3
120 2
121 4

122	2
123	4
124	3
125	2
126	2
127	5
128	2
129	3
130	2
131	4
132	2
133	3
134	2
135	2
136	2
137	5
138	3
139	4
140	2
141	9
142	4
143	5
144	7
145	2
146	2
147	5
148	4
149	2
150	7
151	2
152	2
153	2
154	3
155	2
156	4
157	2
158	3
159	2
160	6
161	2
162	2
163	3
164	2
165	2

166	2
167	2
168	2
169	2
170	2
171	4
172	2
173	3
174	3
175	2
176	2
177	2
178	2
179	2
180	2
181	2
182	3
183	2
184	3
185	2
186	5
187	2
188	7
189	2
190	4
191	2
192	4
193	3
194	2
195	2
196	21
197	2
198	2
199	2
200	2
201	2
202	2
203	2
204	2
205	2
206	2
207	3
208	2
209	2

210	2
211	2
212	5
213	7
214	2
215	4
216	2
217	3
218	2
219	4
220	3
221	2
222	4
223	4
224	2
225	2
226	2
227	2
228	2
229	2
230	3
231	2
232	3
233	2
234	3
235	4
236	3
237	2
238	3
239	2
240	5
241	5
242	2
243	2
244	2
245	2
246	2
247	4
248	2
249	4
250	4
251	2
252	4
253	2

254	2
255	2
256	2
257	2
258	2
259	3
260	2
261	3
262	2
263	3
264	2
265	8
266	4
267	2
268	2
269	3
270	2
271	3
272	7
273	6
274	2
275	3
276	2
277	4
278	2
279	2
280	5
281	2
282	3
283	2
284	3
285	3
286	4
287	2
288	2
289	6
290	2
291	2
292	6
293	2
294	2
295	4
296	3
297	2

298	2
299	4
300	5
301	2
302	2
303	3
304	9
305	2
306	2
307	5
308	2
309	4
310	2
311	4
312	2
313	4
314	2
315	2
316	2
317	4
318	4
319	2
320	2
321	2
322	1
323	2
324	3
325	2
326	2
327	2
328	3
329	2
330	1
331	2
332	3
333	2
334	2
335	2
336	2
337	2
338	2
339	3
340	2
341	2

342	2
343	3
344	3
345	2
346	2
347	2
348	2
349	2
350	2
351	2
352	2
353	2
354	3
355	4
356	2
357	2
358	2
359	3
360	3
361	3
362	2
363	2
364	2
365	2
366	2
367	2
368	2
369	2
370	2
371	2
372	3
373	2
374	4
375	2
376	2
377	2
378	3
379	2
380	2
381	4
382	2
383	2
384	2
385	2

386	2
387	2
388	2
389	2
390	3
391	2
392	2
393	2
394	2
395	2
396	3
397	2
398	2
399	3
400	2
401	3
402	2
403	2
404	2
405	4
406	2
407	3
408	2
409	2
410	2
411	2
412	2
413	2
414	4
415	4
416	2
417	3
418	2
419	2
420	2
421	2
422	3
423	2
424	2
425	2
426	2
427	2
428	3

Comp 0 = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58
59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87
88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132
133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174
175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195
196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237
238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258
259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279
280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321
322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342
343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363
364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384
385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405
406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426
427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447
448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468
469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489
490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510
511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531
532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552
553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573
574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594
595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615
616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636
637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657
658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678
679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699
700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720
721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741
742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762
763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783
784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804
805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825
826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846
847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867
868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888
889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909
910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930
931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951

952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972
973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993
994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010
1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026
1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042
1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058
1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074
1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090
1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106
1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122
1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138
1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154
1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170
1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186
1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202
1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218
1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234
1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250
1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266
1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282
1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298
1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314
1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330
1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346
1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362
1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378
1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394
1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410
1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426
1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442
1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458
1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474
1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490
1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506
1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522
1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538
1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554
1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570
1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586
1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602
1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618
1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634
1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650
1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666

1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682
1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698
1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714
1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730
1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746
1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762
1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778
1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794
1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810
1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826
1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842
1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858
1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874
1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890
1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906
1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922
1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938
1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954
1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970
1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986
1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018
2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034
2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050
2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066
2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2080 2081 2082 2083 2084
2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100
2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116
2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132
2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148
2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164
2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180
2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196
2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212
2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228
2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244
2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260
2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276
2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292
2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308
2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324
2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340
2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356
2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372

2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388
2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404
2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420
2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436
2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452
2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468
2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484
2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500
2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516
2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532
2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548
2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564
2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580
2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596
2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612
2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628
2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644
2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660
2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676
2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692
2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708
2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724
2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740
2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756
2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772
2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788
2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804
2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2820
2821 2822 2823 2824 2825 2826 2827 2828 2829 2830 2831 2832 2833 2834 2835 2836
2837 2838 2839 2840 2841 2842 2843 2844 2845 2846 2847 2848 2849 2850 2851 2852
2853 2854 2855 2856 2857 2858 2859 2860 2861 2862 2863 2864 2865 2866 2867 2868
2869 2870 2871 2872 2873 2874 2875 2876 2877 2878 2879 2880 2881 2882 2883 2884
2885 2886 2889 2890 2891 2892 2893 2894 2895 2896 2897 2898 2899 2900 2901 2902
2903 2904 2905 2906 2907 2908 2909 2910 2911 2912 2913 2914 2915 2916 2917 2918
2919 2920 2921 2922 2923 2924 2925 2926 2927 2928 2929 2930 2931 2932 2933 2934
2935 2936 2937 2938 2939 2940 2941 2942 2943 2944 2945 2946 2947 2948 2949 2950
2951 2952 2953 2954 2955 2956 2957 2958 2959 2960 2961 2962 2963 2964 2965 2966
2967 2968 2969 2970 2971 2972 2973 2974 2975 2976 2977 2978 2979 2980 2981 2982
2983 2984 2985 2986 2987 2988 2989 2990 2991 2992 2993 2994 2995 2996 2997 2998
2999 3000 3001 3002 3003 3004 3005 3006 3007 3008 3009 3010 3011 3012 3013 3014
3015 3016 3017 3018 3019 3020 3021 3022 3023 3024 3025 3026 3027 3028 3029 3030
3031 3032 3033 3034 3035 3036 3037 3038 3039 3040 3041 3042 3043 3044 3045 3046
3047 3048 3049 3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062
3063 3064 3065 3066 3067 3068 3069 3070 3071 3072 3073 3074 3075 3076 3077 3078

3079 3080 3081 3082 3083 3084 3085 3086 3087 3088 3089 3090 3091 3092 3093 3094
3095 3096 3097 3098 3099 3100 3101 3102 3103 3104 3105 3106 3107 3108 3109 3110
3111 3112 3113 3114 3115 3116 3117 3118 3119 3120 3121 3122 3123 3124 3125 3126
3127 3128 3129 3130 3131 3132 3133 3134 3135 3136 3137 3138 3139 3140 3141 3142
3143 3144 3145 3146 3147 3148 3149 3150 3151 3152 3153 3154 3155 3156 3157 3158
3159 3160 3161 3162 3163 3164 3165 3166 3167 3168 3169 3170 3171 3172 3173 3174
3175 3176 3177 3178 3179 3180 3181 3182 3183 3184 3185 3186 3187 3188 3189 3190
3191 3192 3193 3194 3195 3196 3197 3198 3199 3200 3201 3202 3203 3204 3205 3206
3207 3208 3209 3210 3211 3212 3213 3214 3215 3216 3217 3218 3219 3220 3221 3222
3223 3224 3225 3226 3227 3228 3229 3230 3231 3232 3233 3234 3235 3236 3237 3238
3239 3240 3241 3242 3243 3244 3245 3246 3247 3248 3249 3250 3251 3252 3253 3254
3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267 3268 3269 3270
3271 3272 3273 3274 3275 3276 3277 3278 3279 3280 3281 3282 3283 3284 3285 3286
3287 3288 3289 3290 3291 3292 3293 3294 3295 3296 3297 3298 3299 3300 3301 3302
3303 3304 3305 3306 3307 3308 3309 3310 3311 3312 3313 3314 3315 3316 3317 3318
3319 3320 3321 3322 3323 3324 3325 3326 3327 3328 3329 3330 3331 3332 3333 3334
3335 3336 3337 3338 3339 3340 3341 3342 3343 3344 3345 3346 3347 3348 3349 3350
3351 3352 3353 3354 3355 3356 3357 3358 3359 3360 3361 3362 3363 3364 3365 3366
3367 3368 3369 3370 3371 3372 3373 3374 3375 3376 3377 3378 3379 3380 3381 3382
3383 3384 3385 3386 3387 3388 3389 3390 3391 3392 3393 3394 3395 3396 3397 3398
3399 3400 3401 3402 3403 3404 3405 3406 3407 3408 3409 3410 3411 3412 3413 3414
3415 3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 3428 3429 3430
3431 3432 3433 3434 3435 3436 3437 3438 3439 3440 3441 3442 3443 3444 3445 3446
3447 3448 3449 3450 3451 3452 3453 3454 3455 3456 3457 3458 3459 3460 3461 3462
3463 3464 3465 3466 3467 3468 3469 3470 3471 3472 3473 3474 3475 3476 3477 3478
3479 3480 3481 3482 3483 3484 3485 3486 3487 3488 3489 3490 3491 3492 3493 3494
3495 3496 3497 3498 3499 3500 3501 3502 3503 3504 3505 3506 3507 3508 3509 3510
3511 3512 3513 3514 3515 3516 3517 3518 3519 3520 3521 3522 3523 3524 3525 3526
3527 3528 3529 3530 3531 3532 3533 3534 3535 3536 3537 3538 3539 3540 3541 3542
3543 3544 3545 3546 3547 3548 3549 3550 3551 3552 3553 3554 3555 3556 3557 3558
3559 3560 3561 3562 3563 3564 3565 3566 3567 3568 3569 3570 3571 3572 3573 3574
3575 3576 3577 3578 3579 3580 3581 3582 3583 3584 3585 3586 3587 3588 3589 3590
3591 3592 3593 3594 3595 3596 3597 3598 3599 3600 3601 3602 3603 3604 3605 3606
3607 3608 3609 3610 3611 3612 3613 3614 3615 3616 3617 3618 3619 3620 3621 3622
3623 3624 3625 3626 3627 3628 3629 3630 3631 3632 3633 3634 3635 3636 3637 3638
3639 3640 3641 3642 3643 3644 3645 3646 3647 3648 3649 3650 3651 3652 3653 3654
3655 3656 3657 3658 3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670
3671 3672 3673 3674 3675 3676 3677 3678 3679 3680 3681 3682 3683 3684 3685 3686
3687 3688 3689 3690 3691 3692 3693 3694 3695 3696 3697 3698 3699 3700 3701 3702
3703 3704 3705 3706 3707 3708 3709 3710 3711 3712 3713 3714 3715 3716 3717 3718
3719 3720 3721 3722 3723 3724 3725 3726 3727 3728 3729 3730 3731 3732 3733 3734
3735 3736 3737 3738 3739 3740 3741 3742 3743 3744 3745 3746 3747 3748 3749 3750
3751 3752 3753 3754 3755 3756 3757 3758 3759 3760 3761 3762 3763 3764 3765 3766
3767 3768 3769 3770 3771 3772 3773 3774 3775 3776 3777 3778 3779 3780 3781 3782

3783 3786 3787 3788 3789 3790 3791 3792 3793 3794 3795 3796 3797 3798 3799 3800
3801 3802 3803 3804 3805 3806 3807 3808 3809 3810 3811 3812 3813 3814 3815 3816
3817 3818 3819 3820 3821 3822 3823 3824 3825 3826 3827 3828 3829 3830 3831 3832
3833 3834 3835 3836 3837 3838 3839 3840 3841 3842 3843 3844 3845 3846 3847 3848
3849 3850 3851 3852 3853 3854 3855 3856 3857 3858 3859 3860 3861 3862 3863 3864
3865 3866 3867 3868 3869 3870 3871 3872 3873 3874 3875 3876 3877 3878 3879 3880
3881 3882 3883 3884 3885 3886 3887 3888 3889 3890 3891 3892 3893 3894 3895 3896
3897 3898 3899 3900 3901 3902 3903 3904 3905 3906 3907 3908 3909 3910 3911 3912
3913 3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928
3929 3930 3931 3932 3933 3934 3935 3936 3937 3938 3939 3940 3941 3942 3943 3944
3945 3946 3947 3948 3949 3950 3951 3952 3953 3954 3955 3956 3957 3958 3959 3960
3961 3962 3963 3964 3965 3966 3967 3968 3969 3970 3971 3972 3973 3974 3975 3976
3977 3978 3979 3980 3981 3982 3983 3984 3985 3986 3987 3988 3989 3990 3991 3992
3993 3994 3995 3996 3997 3998 3999 4000 4001 4002 4003 4004 4005 4006 4007 4008
4009 4010 4011 4012 4013 4014 4015 4016 4017 4018 4019 4020 4021 4022 4023 4024
4025 4026 4027 4028 4029 4030 4031 4032 4033 4034 4035 4036 4037 4038 4039 4040
4041 4042 4043 4044 4045 4046 4047 4048 4049 4050 4051 4052 4053 4054 4055 4056
4057 4058 4059 4060 4061 4062 4063 4064 4065 4066 4067 4068 4069 4070 4071 4072
4073 4074 4075 4076 4077 4078 4079 4080 4081 4082 4083 4084 4085 4086 4087 4088
4089 4090 4091 4092 4093 4094 4095 4096 4097 4098 4099 4100 4101 4102 4103 4104
4105 4106 4107 4108 4109 4110 4111 4112 4113 4114 4115 4116 4117 4118 4119 4120
4121 4122 4123 4124 4125 4126 4127 4128 4129 4130 4131 4132 4133 4134 4135 4136
4137 4138 4139 4140 4141 4142 4143 4144 4145 4146 4147 4148 4149 4150 4151 4152
4153 4154 4155 4156 4157 4158 4159 4160 4161 4162 4163 4164 4165 4166 4167 4168
4169 4170 4171 4172 4173 4174 4175 4176 4177 4178 4179 4180 4181 4182 4183 4184
4185 4186 4187 4188 4189 4190 4191 4192 4193 4194 4195 4196 4197 4198 4199 4200
4201 4202 4203 4204 4205 4206 4207 4208 4209 4210 4211 4212 4213 4214 4215 4216
4217 4218 4219 4220 4221 4222 4223 4224 4225 4226 4227 4228 4229 4230 4231 4232
4233 4234 4235 4236 4237 4238 4239 4240 4241 4242 4243 4244 4245 4246 4247 4248
4249 4250 4251 4252 4253 4254 4255 4256 4257 4258 4259 4260 4261 4262 4263 4264
4265 4266 4267 4268 4269 4270 4271 4272 4273 4281 4284 4285 4286 4287 4288 4289
4290 4291 4292 4293 4294 4295 4296 4297 4298 4299 4300 4301 4302 4303 4304 4305
4306 4307 4308 4309 4310 4311 4312 4313 4314 4315 4316 4317 4318 4319 4320 4321
4322 4323 4324 4325 4326 4327 4328 4329 4330 4331 4332 4333 4334 4335 4336 4337
4338 4339 4340 4341 4342 4343 4344 4345 4346 4347 4348 4349 4350 4351 4352 4353
4354 4355 4356 4357 4358 4359 4360 4361 4362 4363 4364 4365 4366 4367 4368 4369
4370 4371 4372 4373 4374 4375 4376 4377 4378 4379 4380 4381 4382 4383 4384 4385
4386 4387 4388 4389 4390 4391 4392 4393 4394 4395 4396 4397 4398 4399 4400 4401
4402 4403 4404 4405 4406 4407 4408 4409 4410 4411 4412 4413 4414 4415 4416 4417
4418 4419 4420 4421 4422 4423 4424 4425 4426 4427 4428 4429 4430 4431 4432 4433
4434 4435 4436 4437 4438 4439 4440 4441 4442 4443 4444 4445 4446 4447 4448 4449
4450 4451 4452 4453 4454 4455 4456 4457 4458 4459 4460 4461 4462 4463 4464 4465
4466 4467 4468 4469 4470 4471 4472 4473 4474 4475 4476 4477 4478 4479 4480 4481
4482 4483 4484 4485 4486 4487 4488 4489 4490 4491 4492 4493 4494 4495 4496 4497

4498 4499 4500 4501 4502 4503 4504 4505 4506 4507 4508 4509 4510 4511 4512 4513
4514 4515 4516 4517 4518 4519 4520 4521 4522 4523 4524 4525 4526 4527 4530 4531
4532 4533 4534 4535 4536 4537 4538 4539 4540 4541 4542 4543 4544 4545 4546 4547
4548 4549 4550 4551 4552 4553 4554 4555 4556 4557 4558 4559 4560 4561 4562 4563
4564 4565 4566 4567 4568 4569 4570 4571 4572 4573 4574 4575 4578 4579 4580 4581
4582 4583 4584 4585 4586 4587 4588 4589 4590 4591 4592 4593 4594 4595 4596 4597
4598 4599 4600 4601 4602 4603 4604 4605 4606 4607 4608 4609 4610 4611 4614 4615
4616 4617 4618 4619 4620 4621 4622 4623 4624 4625 4626 4627 4628 4629 4630 4631
4632 4633 4634 4637 4638 4639 4640 4641 4642 4643 4644 4645 4646 4647 4650 4651
4652 4653 4654 4655 4656 4657 4658 4659 4660 4661 4662 4663 4664 4665 4666 4667
4668 4669 4670 4671 4672 4673 4674 4675 4676 4677 4678 4679 4680 4681 4682 4683
4684 4685 4686 4687 4688 4689 4690 4691 4692 4693 4694 4695 4696 4697 4698 4699
4700 4701 4702 4703 4704 4705 4706 4707 4708 4709 4710 4711 4712 4713 4714 4715
4716 4717 4718 4719 4720 4721 4722 4723 4724 4725 4726 4727 4728 4729 4730 4731
4732 4733 4734 4735 4736 4737 4738 4739 4740 4741 4742 4743 4744 4745 4746 4747
4748 4749 4750 4751 4752 4753 4754 4755 4756 4757 4758 4759 4760 4761 4762 4763
4764 4765 4766 4767 4768 4769 4770 4771 4772 4773 4774 4775 4776 4777 4778 4779
4780 4781 4782 4783 4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4794 4795
4796 4797 4798 4799 4800 4801 4802 4803 4804 4805 4806 4807 4808 4809 4810 4811
4812 4813 4814 4819 4820 4821 4822 4823 4824 4825 4826 4827 4828 4829 4830 4831
4832 4833 4834 4835 4836 4837 4838 4839 4840 4841 4842 4843 4844 4845 4846 4847
4848 4849 4850 4851 4852 4856 4857 4858 4859 4860 4861 4862 4863 4864 4865 4866
4867 4868 4869 4870 4871 4872 4873 4874 4875 4876 4877 4878 4879 4880 4881 4882
4883 4884 4885 4886 4887 4888 4889 4890 4891 4892 4893 4894 4895 4896 4897 4898
4899 4900 4901 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916
4917 4918 4919 4920 4921 4922 4923 4924 4925 4926 4927 4928 4929 4930 4931 4932
4933 4934 4935 4936 4937 4938 4939 4940 4941 4942 4943 4944 4945 4946 4947 4948
4949 4950 4951 4952 4953 4954 4955 4956 4957 4958 4959 4960 4961 4962 4963 4964
4965 4966 4967 4970 4971 4972 4973 4974 4975 4976 4977 4978 4979 4980 4981 4982
4983 4984 4985 4986 4987 4988 4989 4990 4991 4992 4993 4994 4995 4996 4997 4998
4999 5000 5001 5002 5003 5004 5005 5006 5007 5008 5009 5015 5016 5017 5018 5019
5020 5021 5022 5023 5024 5025 5026 5027 5028 5029 5030 5031 5032 5033 5034 5035
5036 5037 5038 5039 5040 5041 5042 5043 5044 5045 5046 5047 5048 5049 5050 5051
5052 5053 5054 5055 5056 5057 5058 5059 5060 5061 5062 5063 5064 5065 5066 5067
5068 5069 5070 5071 5072 5073 5074 5075 5076 5077 5078 5079 5080 5081 5082 5083
5084 5085 5086 5087 5088 5089 5090 5091 5092 5093 5094 5095 5096 5097 5098 5099
5100 5107 5108 5109 5110 5111 5112 5113 5114 5115 5116 5117 5118 5119 5120 5121
5122 5123 5124 5125 5126 5127 5128 5129 5130 5131 5132 5133 5134 5135 5136 5137
5138 5139 5140 5141 5142 5143 5144 5145 5146 5147 5148 5149 5150 5151 5154 5155
5156 5157 5158 5159 5160 5161 5162 5163 5164 5165 5166 5167 5168 5169 5170 5171
5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183 5184 5185 5186 5187
5188 5189 5190 5191 5192 5193 5194 5195 5196 5197 5198 5199 5200 5201 5202 5203
5204 5205 5206 5207 5208 5209 5210 5211 5212 5213 5214 5215 5216 5217 5218 5219
5220 5221 5222 5223 5224 5225 5226 5227 5228 5229 5230 5231 5232 5233 5234 5235

5236 5237 5238 5239 5240 5241 5242 5243 5244 5245 5246 5247 5248 5249 5250 5251
5252 5253 5254 5255 5256 5257 5258 5259 5260 5261 5262 5263 5264 5265 5266 5267
5268 5269 5270 5271 5272 5273 5274 5275 5276 5277 5278 5279 5280 5281 5282 5283
5284 5285 5286 5287 5288 5289 5290 5291 5292 5293 5294 5295 5296 5297 5298 5299
5300 5301 5302 5303 5304 5305 5306 5307 5308 5309 5310 5311 5312 5313 5314 5315
5316 5317 5318 5319 5320 5321 5322 5323 5324 5325 5326 5327 5328 5329 5330 5331
5332 5333 5334 5335 5336 5337 5338 5339 5340 5345 5346 5347 5348 5349 5350 5351
5352 5353 5354 5355 5356 5357 5358 5359 5360 5361 5362 5363 5364 5365 5366 5367
5368 5369 5370 5371 5374 5375 5376 5377 5378 5379 5380 5381 5382 5383 5384 5385
5386 5387 5388 5389 5390 5391 5392 5393 5394 5395 5396 5397 5398 5399 5402 5403
5404 5405 5406 5407 5408 5409 5410 5411 5412 5413 5414 5415 5416 5417 5418 5419
5420 5421 5422 5423 5424 5425 5426 5427 5428 5429 5430 5431 5432 5433 5434 5435
5436 5437 5438 5439 5440 5441 5442 5443 5444 5445 5446 5447 5448 5449 5450 5451
5452 5453 5454 5455 5456 5457 5458 5459 5460 5461 5462 5463 5464 5465 5466 5467
5468 5469 5470 5471 5472 5473 5474 5475 5476 5477 5478 5479 5480 5481 5482 5483
5484 5485 5486 5487 5488 5489 5490 5491 5492 5493 5494 5495 5496 5497 5498 5499
5500 5501 5502 5503 5504 5505 5506 5507 5508 5509 5510 5511 5512 5513 5514 5515
5516 5517 5518 5519 5520 5521 5522 5523 5524 5525 5526 5527 5528 5529 5530 5531
5532 5533 5534 5535 5536 5537 5538 5539 5540 5541 5542 5543 5544 5545 5546 5547
5548 5549 5550 5551 5552 5553 5554 5555 5556 5557 5558 5559 5560 5565 5566 5567
5568 5569 5570 5571 5572 5573 5574 5575 5576 5577 5578 5579 5580 5581 5582 5583
5584 5585 5586 5587 5588 5591 5592 5593 5594 5595 5596 5597 5598 5599 5600 5601
5602 5603 5604 5605 5606 5607 5608 5609 5610 5611 5612 5613 5614 5615 5616 5617
5618 5619 5620 5621 5622 5623 5624 5625 5626 5627 5628 5629 5630 5631 5632 5633
5634 5635 5636 5637 5638 5639 5640 5641 5642 5646 5647 5648 5649 5650 5651 5652
5653 5654 5655 5656 5657 5658 5659 5660 5661 5662 5663 5664 5665 5666 5667 5668
5669 5670 5671 5672 5673 5674 5675 5676 5677 5678 5679 5680 5681 5682 5683 5684
5685 5686 5687 5688 5689 5690 5698 5699 5700 5701 5702 5705 5706 5707 5708 5709
5710 5711 5712 5713 5714 5715 5716 5717 5718 5719 5720 5721 5722 5723 5724 5725
5726 5727 5728 5729 5730 5731 5732 5733 5734 5735 5736 5737 5738 5739 5740 5741
5742 5743 5744 5745 5746 5747 5748 5749 5750 5754 5755 5756 5757 5758 5759 5760
5761 5762 5763 5764 5765 5766 5767 5768 5769 5770 5771 5772 5773 5774 5775 5776
5777 5778 5779 5780 5781 5782 5785 5786 5787 5788 5791 5792 5793 5794 5795 5796
5797 5798 5799 5800 5801 5802 5803 5804 5805 5806 5807 5808 5809 5810 5811 5812
5813 5814 5815 5816 5817 5818 5819 5820 5821 5822 5823 5824 5825 5826 5827 5828
5829 5830 5831 5832 5833 5834 5835 5836 5837 5838 5839 5840 5841 5842 5843 5844
5845 5846 5847 5848 5849 5850 5851 5852 5853 5854 5855 5856 5857 5858 5859 5860
5861 5862 5863 5864 5865 5866 5867 5868 5869 5870 5871 5872 5873 5874 5875 5876
5877 5878 5879 5880 5881 5882 5883 5884 5885 5886 5887 5888 5889 5890 5891 5892
5893 5894 5895 5896 5897 5898 5899 5900 5901 5902 5903 5904 5905 5906 5907 5908
5909 5910 5911 5912 5913 5914 5915 5916 5917 5918 5919 5920 5921 5922 5923 5924
5925 5926 5927 5928 5929 5930 5931 5937 5938 5939 5940 5941 5942 5943 5944 5945
5946 5947 5948 5949 5950 5951 5952 5953 5954 5955 5956 5957 5958 5959 5960 5961
5962 5963 5964 5965 5966 5967 5968 5969 5970 5971 5972 5973 5974 5975 5976 5977

5978 5979 5980 5981 5982 5983 5984 5985 5986 5987 5988 5989 5990 5991 5992 5993
5994 5995 5996 5997 5998 5999 6000 6001 6002 6003 6004 6005 6006 6007 6008 6009
6010 6011 6012 6013 6018 6019 6020 6021 6022 6023 6024 6025 6026 6027 6028 6029
6030 6031 6032 6033 6034 6035 6036 6037 6038 6039 6040 6041 6042 6043 6044 6045
6046 6047 6048 6049 6050 6051 6052 6053 6054 6055 6056 6057 6058 6059 6060 6061
6062 6063 6064 6065 6078 6079 6080 6081 6082 6083 6084 6085 6086 6087 6088 6089
6090 6091 6092 6093 6094 6095 6096 6097 6098 6099 6100 6101 6102 6103 6104 6105
6106 6107 6108 6109 6110 6111 6112 6113 6114 6115 6116 6117 6118 6119 6120 6121
6122 6123 6124 6125 6126 6127 6128 6129 6130 6131 6132 6133 6134 6135 6136 6137
6138 6139 6140 6141 6142 6143 6146 6147 6148 6149 6150 6151 6152 6153 6154 6155
6156 6157 6158 6159 6160 6161 6162 6163 6164 6165 6166 6167 6168 6169 6170 6171
6172 6173 6174 6175 6176 6177 6178 6179 6180 6181 6182 6183 6184 6185 6186 6189
6190 6191 6192 6193 6194 6195 6197 6198 6199 6200 6201 6202 6203 6204 6205 6206
6207 6208 6209 6210 6211 6212 6213 6214 6215 6216 6217 6218 6219 6220 6221 6222
6223 6227 6228 6229 6230 6231 6232 6233 6234 6235 6236 6237 6238 6239 6240 6241
6242 6243 6244 6245 6246 6247 6248 6249 6250 6251 6252 6253 6254 6255 6256 6257
6258 6259 6260 6261 6262 6263 6264 6265 6266 6267 6268 6269 6270 6271 6272 6273
6274 6275 6276 6277 6280 6281 6282 6283 6284 6285 6286 6287 6288 6289 6290 6291
6292 6293 6294 6295 6296 6297 6298 6299 6300 6301 6302 6303 6304 6305 6306 6307
6308 6318 6319 6320 6321 6322 6323 6324 6325 6326 6327 6328 6329 6330 6331 6332
6333 6334 6335 6336 6337 6338 6339 6340 6341 6342 6343 6344 6345 6346 6347 6348
6349 6350 6351 6352 6353 6354 6355 6356 6357 6358 6359 6360 6361 6362 6363 6364
6365 6366 6367 6368 6369 6370 6371 6372 6373 6376 6377 6378 6379 6380 6381 6382
6383 6384 6385 6386 6387 6388 6389 6390 6391 6392 6393 6394 6395 6396 6397 6398
6399 6400 6401 6402 6403 6404 6405 6406 6407 6408 6409 6410 6411 6412 6413 6416
6417 6418 6419 6420 6421 6424 6425 6426 6427 6428 6429 6430 6431 6432 6433 6434
6435 6436 6437 6438 6439 6440 6441 6442 6443 6444 6445 6446 6447 6448 6449 6450
6453 6454 6455 6456 6457 6458 6459 6460 6461 6462 6463 6464 6465 6466 6467 6468
6469 6470 6471 6472 6473 6474 6475 6476 6477 6478 6479 6480 6481 6482 6483 6484
6485 6486 6487 6488 6489 6490 6491 6492 6496 6497 6498 6499 6500 6501 6502 6503
6504 6505 6506 6507 6508 6509 6510 6511 6512 6513 6514 6515 6516 6517 6518 6519
6520 6521 6522 6523 6524 6525 6526 6527 6528 6529 6530 6533 6534 6535 6536 6537
6538 6539 6540 6541 6542 6543 6544 6552 6553 6554 6555 6560 6561 6562 6563 6564
6565 6566 6567 6568 6569 6570 6571 6572 6573 6574 6575 6576 6577 6581 6582 6583
6584 6585 6586 6587 6588 6594 6595 6596 6597 6598 6599 6600 6601 6602 6603 6604
6605 6606 6607 6608 6609 6610 6611 6612 6613 6614 6615 6616 6617 6618 6619 6620
6621 6622 6623 6624 6625 6626 6627 6628 6629 6630 6631 6632 6633 6634 6635 6636
6637 6638 6639 6640 6641 6642 6643 6644 6645 6646 6647 6648 6649 6650 6651 6652
6653 6654 6655 6656 6657 6658 6659 6660 6661 6662 6663 6668 6669 6670 6671 6672
6673 6674 6675 6676 6677 6678 6679 6680 6681 6682 6683 6684 6685 6686 6687 6688
6689 6690 6691 6692 6693 6694 6695 6696 6697 6698 6699 6702 6703 6704 6705 6706
6707 6708 6709 6710 6711 6712 6713 6714 6715 6716 6717 6718 6719 6720 6721 6722
6723 6724 6725 6726 6727 6728 6729 6730 6731 6732 6733 6734 6735 6736 6737 6738
6739 6740 6741 6742 6743 6744 6745 6746 6747 6748 6749 6750 6751 6752 6753 6754

6755 6756 6757 6758 6759 6760 6761 6762 6763 6764 6765 6766 6767 6768 6769 6770
6771 6772 6773 6776 6777 6778 6779 6780 6781 6782 6783 6784 6785 6786 6787 6788
6789 6790 6791 6792 6793 6794 6795 6796 6797 6798 6799 6800 6801 6802 6803 6804
6805 6806 6807 6808 6818 6819 6820 6821 6822 6823 6824 6825 6826 6827 6828 6829
6830 6831 6832 6833 6834 6835 6836 6837 6838 6848 6849 6850 6851 6852 6853 6854
6855 6856 6857 6858 6859 6860 6861 6862 6863 6864 6865 6866 6867 6868 6869 6870
6871 6872 6873 6876 6877 6878 6879 6880 6881 6882 6883 6884 6885 6886 6887 6888
6889 6890 6891 6892 6893 6894 6895 6896 6897 6898 6899 6900 6901 6902 6905 6906
6907 6908 6909 6910 6911 6912 6913 6914 6917 6918 6919 6920 6921 6922 6926 6927
6928 6929 6930 6931 6932 6933 6934 6935 6936 6937 6938 6939 6940 6941 6942 6943
6946 6947 6948 6949 6950 6951 6952 6953 6956 6957 6958 6959 6960 6961 6962 6963
6964 6965 6966 6967 6968 6969 6970 6971 6972 6973 6974 6975 6979 6980 6981 6982
6983 6984 6985 6986 6987 6988 6989 6990 6991 6992 6993 6994 6995 6996 6997 6998
6999 7000 7001 7002 7003 7004 7005 7006 7007 7008 7009 7010 7011 7012 7013 7014
7015 7016 7017 7018 7019 7020 7021 7022 7023 7024 7025 7026 7027 7028 7029 7030
7031 7032 7033 7034 7035 7036 7037 7038 7039 7040 7041 7042 7043 7044 7045 7046
7047 7048 7049 7050 7051 7052 7053 7054 7055 7056 7057 7058 7059 7060 7061 7062
7063 7064 7065 7066 7067 7068 7069 7070 7071 7072 7073 7074 7075 7076 7077 7078
7079 7082 7083 7084 7085 7086 7087 7088 7089 7090 7091 7094 7095 7098 7099 7100
7101 7102 7103 7104 7105 7106 7107 7108 7109 7110 7111 7112 7113 7114 7115 7116
7117 7118 7119 7120 7121 7122 7123 7124 7125 7128 7129 7130 7131 7132 7133 7134
7135 7136 7145 7146 7147 7148 7149 7150 7151 7152 7153 7154 7160 7161 7162 7163
7164 7165 7166 7167 7168 7169 7170 7176 7177 7178 7179 7180 7181 7182 7183 7184
7185 7186 7187 7188 7189 7192 7193 7194 7195 7196 7197 7198 7199 7200 7201 7202
7203 7204 7205 7206 7207 7208 7209 7210 7211 7212 7213 7214 7222 7223 7224 7225
7226 7227 7228 7229 7230 7231 7232 7233 7237 7238 7239 7240 7241 7242 7243 7244
7245 7246 7247 7248 7249 7250 7251 7252 7253 7254 7255 7256 7257 7258 7259 7262
7263 7264 7265 7266 7267 7268 7269 7270 7271 7272 7273 7274 7279 7282 7283 7284
7285 7286 7287 7288 7289 7290 7291 7292 7293 7294 7295 7296 7297 7298 7301 7302
7311 7312 7313 7314 7315 7316 7317 7318 7319 7320 7321 7322 7323 7326 7327 7328
7329 7330 7331 7332 7333 7334 7335 7336 7337 7342 7343 7344 7345 7346 7347 7348
7349 7350 7351 7352 7353 7354 7355 7363 7364 7365 7366 7367 7370 7371 7372 7373
7376 7377 7378 7379 7380 7381 7382 7383 7384 7385 7386 7387 7388 7389 7390 7391
7392 7393 7394 7395 7396 7397 7398 7399 7400 7401 7402 7403 7404 7407 7408 7409
7410 7411 7412 7413 7414 7415 7416 7417 7418 7419 7420 7421 7422 7423 7424 7425
7426 7427 7428 7429 7430 7431 7432 7433 7434 7435 7436 7437 7438 7439 7440 7441
7442 7443 7444 7445 7446 7447 7448 7449 7450 7451 7452 7453 7456 7457 7458 7459
7460 7461 7462 7465 7466 7467 7468 7469 7470 7471 7472 7473 7474 7475 7476 7477
7478 7479 7480 7481 7482 7483 7484 7485 7486 7487 7488 7489 7490 7494 7495 7496
7497 7498 7499 7500 7501 7502 7503 7506 7507 7508 7509 7510 7511 7512 7513 7514
7515 7516 7517 7518 7519 7520 7524 7525 7526 7527 7528 7529 7530 7531 7532 7533
7534 7542 7543 7544 7547 7548 7549 7550 7551 7552 7553 7554 7555 7556 7557 7558
7559 7560 7561 7562 7563 7564 7565 7566 7567 7568 7569 7570 7571 7572 7573 7574
7590 7591 7592 7593 7594 7595 7596 7597 7598 7599 7600 7601 7602 7603 7604 7605

7606 7607 7608 7609 7610 7611 7612 7613 7614 7615 7616 7617 7618 7619 7620 7626
7627 7628 7629 7630 7631 7632 7633 7634 7635 7636 7637 7638 7639 7640 7641 7642
7643 7644 7645 7646 7647 7648 7649 7650 7651 7652 7653 7656 7657 7658 7659 7660
7661 7662 7663 7664 7665 7666 7669 7670 7671 7672 7673 7674 7675 7676 7677 7678
7679 7680 7681 7688 7689 7690 7691 7692 7693 7694 7695 7696 7697 7698 7699 7702
7703 7704 7705 7706 7707 7708 7709 7710 7711 7712 7713 7716 7717 7718 7719 7720
7721 7722 7723 7724 7725 7726 7727 7731 7732 7733 7734 7735 7736 7737 7738 7739
7740 7741 7742 7743 7744 7745 7746 7747 7748 7749 7754 7755 7765 7766 7767 7768
7772 7773 7774 7775 7776 7777 7778 7779 7780 7781 7782 7783 7784 7785 7786 7787
7788 7789 7790 7791 7792 7793 7794 7795 7796 7797 7798 7799 7800 7801 7802 7803
7804 7805 7806 7807 7808 7809 7810 7811 7812 7813 7814 7815 7816 7817 7818 7819
7820 7821 7822 7825 7826 7827 7828 7829 7830 7831 7832 7833 7836 7837 7838 7839
7840 7841 7842 7843 7844 7848 7849 7850 7851 7852 7853 7854 7855 7856 7857 7858
7859 7860 7861 7862 7863 7864 7865 7866 7867 7868 7869 7870 7871 7872 7873 7874
7875 7876 7877 7878 7879 7880 7881 7882 7883 7884 7885 7886 7887 7888 7889 7890
7891 7892 7893 7894 7895 7896 7897 7903 7904 7905 7906 7907 7908 7909 7910 7911
7912 7913 7914 7915 7916 7917 7924 7925 7926 7927 7928 7929 7930 7931 7932 7935
7939 7940 7941 7942 7943 7944 7945 7946 7947 7948 7949 7950 7951 7952 7953 7954
7955 7956 7957 7958 7959 7960 7961 7962 7963 7964 7965 7966 7967 7968 7969 7970
7971 7972 7976 7979 7980 7981 7982 7983 7984 7985 7986 7987 7988 7989 7990 7991
7992 7993 7994 7995 7996 7997 7998 7999 8000 8001 8002 8003 8004 8005 8006 8007
8008 8009 8013 8014 8015 8016 8017 8018 8019 8020 8021 8022 8023 8024 8028 8029
8030 8031 8032 8033 8034 8038 8039 8040 8041 8042 8043 8044 8045 8046 8047 8048
8049 8050 8051 8052 8053 8056 8063 8064 8065 8066 8067 8068 8069 8070 8071 8072
8073 8074 8075 8076 8077 8078 8079 8080 8081 8082 8083 8084 8085 8086 8087 8088
8089 8090 8095 8096 8097 8098 8099 8103 8104 8107 8108 8109 8110 8111 8112 8113
8114 8115 8116 8117 8118 8119 8120 8121 8131 8132 8133 8137 8138 8141 8142 8143
8144 8145 8146 8147 8152 8153 8154 8155 8156 8157 8163 8164 8165 8166 8167 8168
8169 8170 8171 8174 8175 8176 8177 8178 8179 8180 8185 8186 8187 8188 8189 8190
8191 8192 8193 8206 8207 8208 8209 8210 8213 8214 8215 8216 8217 8218 8219 8220
8221 8231 8232 8233 8238 8239 8240 8241 8242 8243 8244 8259 8265 8266 8271 8272
8273 8274 8275 8276 8277 8278 8279 8280 8281 8282 8283 8284 8285 8286 8287 8288
8289 8290 8291 8292 8293 8294 8297 8298 8299 8300 8301 8302 8303 8304 8311 8312
8313 8314 8315 8316 8317 8320 8321 8322 8323 8324 8325 8326 8327 8328 8329 8330
8331 8332 8333 8334 8335 8336 8337 8338 8339 8340 8341 8342 8343 8346 8347 8348
8349 8350 8351 8352 8353 8354 8355 8356 8357 8358 8361 8362 8363 8364 8365 8366
8370 8371 8372 8373 8374 8377 8382 8383 8384 8385 8386 8387 8388 8391 8392 8393
8394 8395 8396 8397 8398 8399 8400 8404 8407 8408 8415 8416 8425 8426 8427 8428
8429 8430 8431 8432 8441 8442 8443 8444 8445 8446 8447 8448 8449 8450 8451 8452
8453 8454 8455 8456 8457 8458 8459 8460 8461 8462 8463 8464 8465 8466 8467 8468
8469 8470 8475 8476 8477 8478 8479 8480 8481 8482 8483 8484 8485 8486 8487 8488
8489 8490 8491 8492 8493 8494 8495 8496 8497 8498 8499 8500 8501 8502 8505 8506
8511 8512 8515 8516 8517 8518 8519 8520 8521 8522 8523 8527 8528 8529 8530 8531
8532 8533 8534 8535 8536 8537 8538 8539 8540 8544 8545 8546 8547 8548 8549 8550

8551 8552 8553 8554 8555 8556 8557 8558 8559 8560 8561 8562 8563 8564 8565 8566
8567 8568 8569 8570 8571 8572 8573 8574 8575 8576 8577 8578 8579 8580 8581 8582
8583 8584 8585 8586 8587 8588 8589 8590 8591 8592 8593 8594 8595 8596 8597 8598
8599 8600 8601 8602 8603 8610 8611 8612 8615 8622 8623 8624 8625 8626 8627 8628
8629 8630 8631 8632 8636 8637 8638 8639 8640 8641 8642 8643 8644 8645 8646 8647
8648 8649 8650 8651 8652 8653 8654 8655 8656 8657 8658 8659 8660 8663 8664 8665
8666 8667 8668 8669 8670 8671 8672 8673 8674 8675 8676 8677 8681 8682 8683 8684
8685 8686 8687 8688 8689 8690 8691 8694 8695 8696 8697 8698 8699 8700 8701 8702
8703 8704 8705 8706 8707 8708 8709 8710 8711 8712 8713 8714 8715 8716 8722 8723
8724 8725 8726 8728 8729 8732 8740 8741 8742 8743 8744 8745 8746 8747 8752 8759
8760 8761 8762 8763 8764 8765 8766 8767 8773 8776 8777 8791 8794 8795 8796 8797
8798 8799 8800 8801 8802 8803 8804 8805 8806 8807 8808 8809 8810 8811 8812 8813
8814 8815 8816 8817 8818 8819 8820 8821 8822 8823 8824 8825 8826 8827 8828 8829
8830 8831 8834 8837 8838 8839 8840 8841 8844 8845 8846 8847 8848 8849 8850 8851
8852 8853 8854 8855 8856 8857 8858 8859 8860 8861 8866 8867 8868 8869 8870 8871
8874 8875 8876 8877 8882 8883 8886 8887 8888 8889 8892 8893 8894 8895 8896 8897
8898 8899 8900 8901 8902 8903 8904 8907 8914 8915 8916 8917 8918 8919 8920 8921
8926 8927 8928 8929 8930 8941 8942 8950 8956 8961 8962 8963 8964 8965 8966 8967
8968 8969 8970 8971 8972 8973 8976 8979 8980 8981 8982 8983 8984 8985 8986 8987
8993 8994 8995 8996 8997 8998 8999 9000 9005 9006 9007 9011 9012 9013 9014 9017
9018 9019 9020 9021 9022 9023 9028 9029 9030 9035 9038 9041 9042 9045 9046 9047
9048 9049 9050 9051 9052 9057 9058 9061 9062 9063 9064 9065 9066 9067 9068 9069
9070 9071 9072 9073 9077 9078 9079 9080 9081 9082 9083 9084 9087 9088 9092 9102
9103 9112 9113 9114 9115 9116 9129 9130 9133 9134 9135 9136 9137 9138 9139 9142
9149 9150 9151 9152 9153 9154 9155 9160 9161 9162 9163 9164 9165 9166 9167 9168
9169 9170 9171 9172 9173 9174 9175 9178 9179 9180 9181 9190 9191 9192 9193 9200
9201 9204 9205 9206 9207 9210 9211 9212 9215 9216 9217 9220 9225 9229 9230 9231
9234 9235 9241 9242 9243 9244 9245 9249 9250 9251 9252 9253 9254 9255 9256 9257
9258 9259 9260 9261 9262 9263 9264 9265 9266 9267 9268 9269 9270 9271 9272 9273
9274 9275 9290 9291 9292 9295 9296 9299 9303 9309 9323 9324 9325 9326 9329 9335
9336 9337 9346 9347 9348 9349 9350 9351 9357 9360 9361 9362 9363 9364 9365 9369
9372 9373 9374 9375 9376 9377 9381 9382 9386 9387 9388 9393 9398 9399 9400 9407
9410 9411 9412 9421 9422 9423 9424 9425 9426 9427 9430 9431 9432 9433 9434 9439
9445 9459 9476 9484 9485 9486 9489 9490 9491 9492 9493 9504 9505 9508 9509 9510
9511 9512 9513 9514 9523 9524 9525 9526 9533 9538 9541 9544 9547 9548 9550 9551
9552 9553 9556 9557 9558 9559 9560 9561 9562 9563 9564 9565 9566 9567 9568 9569
9575 9576 9577 9578 9579 9580 9588 9589 9592 9593 9594 9595 9596 9597 9598 9599
9601 9619 9623 9624 9629 9630 9631 9632 9635 9644 9645 9648 9649 9650 9651 9656
9657 9662 9665 9666 9677 9678 9679 9684 9699 9700 9703 9713 9714 9717 9726 9734
9745 9746 9776 9780 9781 9782 9800 9801 9829 9830 9833
Comp 1 = 2078 2079
Comp 2 = 2887 2888
Comp 3 = 3784 3785
Comp 4 = 4274 4275 4276 4277 4278 4279 4280

Comp 5 = 4282 4283
Comp 6 = 4528 4529
Comp 7 = 4576 4577
Comp 8 = 4612 4613
Comp 9 = 4635 4636
Comp 10 = 4648 4649
Comp 11 = 4815 4816 4817 4818
Comp 12 = 4853 4854 4855
Comp 13 = 4902 4903
Comp 14 = 4968 4969
Comp 15 = 5010 5011 5012 5013 5014
Comp 16 = 5101 5102 5103 5104 5105 5106 6196
Comp 17 = 5152 5153
Comp 18 = 5341 5342 5343 5344
Comp 19 = 5372 5373
Comp 20 = 5400 5401
Comp 21 = 5561 5562 5563 5564
Comp 22 = 5589 5590
Comp 23 = 5643 5644 5645
Comp 24 = 5691 5692 5693 5694 5695 5696 5697 6493 6494 6495 8727
Comp 25 = 5703 5704
Comp 26 = 5751 5752 5753
Comp 27 = 5783 5784
Comp 28 = 5789 5790
Comp 29 = 5932 5933 5934 5935 5936
Comp 30 = 6014 6015 6016 6017
Comp 31 = 6066 6067 6068 6069 6070 6071
Comp 32 = 6072 6073 6074 6075 6076 6077 9676
Comp 33 = 6144 6145
Comp 34 = 6187 6188
Comp 35 = 6224 6225 6226
Comp 36 = 6278 6279
Comp 37 = 6309 6310 6311
Comp 38 = 6312 6313 6314 6315 6316 6317 9694 9695
Comp 39 = 6374 6375
Comp 40 = 6414 6415
Comp 41 = 6422 6423
Comp 42 = 6451 6452
Comp 43 = 6531 6532
Comp 44 = 6545 6546 6547 6548 6549 6550 6551
Comp 45 = 6556 6557
Comp 46 = 6558 6559
Comp 47 = 6578 6579 6580 6923 6924 6925
Comp 48 = 6589 6590 6591 6592 6593

Comp 49 = 6664 6665 6666 6667
Comp 50 = 6700 6701
Comp 51 = 6774 6775
Comp 52 = 6809 6810 6811 6812 6813
Comp 53 = 6814 6815 6816 6817
Comp 54 = 6839 6840 6841 6842
Comp 55 = 6843 6844 6845 6846 6847
Comp 56 = 6874 6875
Comp 57 = 6903 6904
Comp 58 = 6915 6916
Comp 59 = 6944 6945
Comp 60 = 6954 6955
Comp 61 = 6976 6977 6978
Comp 62 = 7080 7081
Comp 63 = 7092 7093
Comp 64 = 7096 7097
Comp 65 = 7126 7127
Comp 66 = 7137 7138 7139 7140 7141 7142 7143 7144
Comp 67 = 7155 7156 7157 7158 7159
Comp 68 = 7171 7172 7173 7174 7175
Comp 69 = 7190 7191
Comp 70 = 7215 7216
Comp 71 = 7217 7218 7219 7220 7221
Comp 72 = 7234 7235 7236
Comp 73 = 7260 7261
Comp 74 = 7275 7276 7277 7278
Comp 75 = 7280 7281
Comp 76 = 7299 7300
Comp 77 = 7303 7304 7305 7306 7307 7308 7309 7310
Comp 78 = 7324 7325
Comp 79 = 7338 7339 7340 7341
Comp 80 = 7356 7357 7358 7359 7360
Comp 81 = 7361 7362
Comp 82 = 7368 7369
Comp 83 = 7374 7375
Comp 84 = 7405 7406
Comp 85 = 7454 7455
Comp 86 = 7463 7464
Comp 87 = 7491 7492 7493
Comp 88 = 7504 7505
Comp 89 = 7521 7522 7523
Comp 90 = 7535 7536 7537 7538 7539 7540 7541
Comp 91 = 7545 7546
Comp 92 = 7575 7576

Comp 93 = 7577 7578
Comp 94 = 7579 7580 7581 7582 7583 8010 8011 8012
Comp 95 = 7584 7585
Comp 96 = 7586 7587 7588 7589
Comp 97 = 7621 7622 7623 7624 7625 8953 8954 8955
Comp 98 = 7654 7655
Comp 99 = 7667 7668
Comp 100 = 7682 7683 7684 7685 7686 7687
Comp 101 = 7700 7701
Comp 102 = 7714 7715
Comp 103 = 7728 7729 7730
Comp 104 = 7750 7751
Comp 105 = 7752 7753
Comp 106 = 7756 7757 7758 7759 7760 7761 7762
Comp 107 = 7763 7764
Comp 108 = 7769 7770 7771
Comp 109 = 7823 7824
Comp 110 = 7834 7835
Comp 111 = 7845 7846 7847
Comp 112 = 7898 7899 7900 7901 7902
Comp 113 = 7918 7919 7920 7921 7922 7923
Comp 114 = 7933 7934
Comp 115 = 7936 7937 7938
Comp 116 = 7973 7974 7975
Comp 117 = 7977 7978
Comp 118 = 8025 8026 8027
Comp 119 = 8035 8036 8037
Comp 120 = 8054 8055
Comp 121 = 8057 8058 8059 8060
Comp 122 = 8061 8062
Comp 123 = 8091 8092 8093 8094
Comp 124 = 8100 8101 8102
Comp 125 = 8105 8106
Comp 126 = 8122 8123
Comp 127 = 8124 8125 8126 8127 8128
Comp 128 = 8129 8130
Comp 129 = 8134 8135 8136
Comp 130 = 8139 8140
Comp 131 = 8148 8149 8150 8151
Comp 132 = 8158 8159
Comp 133 = 8160 8161 8162
Comp 134 = 8172 8173
Comp 135 = 8181 8182
Comp 136 = 8183 8184

Comp 137 = 8194 8195 8196 8197 8198
Comp 138 = 8199 8200 8201
Comp 139 = 8202 8203 8204 8205
Comp 140 = 8211 8212
Comp 141 = 8222 8223 8224 8225 8226 8227 8228 8229 8230
Comp 142 = 8234 8235 8236 8237
Comp 143 = 8245 8246 8247 8977 8978
Comp 144 = 8248 8249 8250 8251 8252 8253 8254
Comp 145 = 8255 8256
Comp 146 = 8257 8258
Comp 147 = 8260 8261 8262 8263 8264
Comp 148 = 8267 8268 8269 8270
Comp 149 = 8295 8296
Comp 150 = 8305 8306 8307 8308 8309 8310 8417
Comp 151 = 8318 8319
Comp 152 = 8344 8345
Comp 153 = 8359 8360
Comp 154 = 8367 8368 8369
Comp 155 = 8375 8376
Comp 156 = 8378 8379 8380 8381
Comp 157 = 8389 8390
Comp 158 = 8401 8402 8403
Comp 159 = 8405 8406
Comp 160 = 8409 8410 8411 8412 8413 8414
Comp 161 = 8418 8419
Comp 162 = 8420 8421
Comp 163 = 8422 8423 8424
Comp 164 = 8433 8434
Comp 165 = 8435 8436
Comp 166 = 8437 8438
Comp 167 = 8439 8440
Comp 168 = 8471 8472
Comp 169 = 8473 8474
Comp 170 = 8503 8504
Comp 171 = 8507 8508 8509 8510
Comp 172 = 8513 8514
Comp 173 = 8524 8525 8526
Comp 174 = 8541 8542 8543
Comp 175 = 8604 8605
Comp 176 = 8606 8607
Comp 177 = 8608 8609
Comp 178 = 8613 8614
Comp 179 = 8616 8617
Comp 180 = 8618 8619

Comp 181 = 8620 8621
Comp 182 = 8633 8634 8635
Comp 183 = 8661 8662
Comp 184 = 8678 8679 8680
Comp 185 = 8692 8693
Comp 186 = 8717 8718 8719 8720 8721
Comp 187 = 8730 8731
Comp 188 = 8733 8734 8735 8736 8737 9669 9670
Comp 189 = 8738 8739
Comp 190 = 8748 8749 8750 8751
Comp 191 = 8753 8754
Comp 192 = 8755 8756 8757 8758
Comp 193 = 8768 8769 8770
Comp 194 = 8771 8772
Comp 195 = 8774 8775
Comp 196 = 8778 8779 8780 8781 8782 8783 8784 8785 8786 8787 8788 8789 8790 8878 8879 8880 8881 8908 8938 8939 8940
Comp 197 = 8792 8793
Comp 198 = 8832 8833
Comp 199 = 8835 8836
Comp 200 = 8842 8843
Comp 201 = 8862 8863
Comp 202 = 8864 8865
Comp 203 = 8872 8873
Comp 204 = 8884 8885
Comp 205 = 8890 8891
Comp 206 = 8905 8906
Comp 207 = 8909 8910 8911
Comp 208 = 8912 8913
Comp 209 = 8922 8923
Comp 210 = 8924 8925
Comp 211 = 8931 8932
Comp 212 = 8933 8934 8935 8936 8937
Comp 213 = 8943 8944 8945 8946 8947 8948 8949
Comp 214 = 8951 8952
Comp 215 = 8957 8958 8959 8960
Comp 216 = 8974 8975
Comp 217 = 8988 8989 8990
Comp 218 = 8991 8992
Comp 219 = 9001 9002 9003 9004
Comp 220 = 9008 9009 9010
Comp 221 = 9015 9016
Comp 222 = 9024 9025 9026 9027
Comp 223 = 9031 9032 9033 9034

Comp 224 = 9036 9037
Comp 225 = 9039 9040
Comp 226 = 9043 9044
Comp 227 = 9053 9054
Comp 228 = 9055 9056
Comp 229 = 9059 9060
Comp 230 = 9074 9075 9076
Comp 231 = 9085 9086
Comp 232 = 9089 9090 9091
Comp 233 = 9093 9094
Comp 234 = 9095 9096 9097
Comp 235 = 9098 9099 9100 9101
Comp 236 = 9104 9105 9106
Comp 237 = 9107 9108
Comp 238 = 9109 9110 9111
Comp 239 = 9117 9118
Comp 240 = 9119 9120 9121 9122 9123
Comp 241 = 9124 9125 9126 9127 9128
Comp 242 = 9131 9132
Comp 243 = 9140 9141
Comp 244 = 9143 9144
Comp 245 = 9145 9146
Comp 246 = 9147 9148
Comp 247 = 9156 9157 9158 9159
Comp 248 = 9176 9177
Comp 249 = 9182 9183 9184 9185
Comp 250 = 9186 9187 9188 9189
Comp 251 = 9194 9195
Comp 252 = 9196 9197 9198 9199
Comp 253 = 9202 9203
Comp 254 = 9208 9209
Comp 255 = 9213 9214
Comp 256 = 9218 9219
Comp 257 = 9221 9222
Comp 258 = 9223 9224
Comp 259 = 9226 9227 9228
Comp 260 = 9232 9233
Comp 261 = 9236 9237 9238
Comp 262 = 9239 9240
Comp 263 = 9246 9247 9248
Comp 264 = 9276 9277
Comp 265 = 9278 9279 9280 9281 9282 9283 9284 9285
Comp 266 = 9286 9287 9288 9289
Comp 267 = 9293 9294

Comp 268 = 9297 9298
Comp 269 = 9300 9301 9302
Comp 270 = 9304 9305
Comp 271 = 9306 9307 9308
Comp 272 = 9310 9311 9312 9313 9314 9315 9316
Comp 273 = 9317 9318 9319 9320 9321 9322
Comp 274 = 9327 9328
Comp 275 = 9330 9331 9332
Comp 276 = 9333 9334
Comp 277 = 9338 9339 9340 9341
Comp 278 = 9342 9343
Comp 279 = 9344 9345
Comp 280 = 9352 9353 9354 9355 9356
Comp 281 = 9358 9359
Comp 282 = 9366 9367 9368
Comp 283 = 9370 9371
Comp 284 = 9378 9379 9380
Comp 285 = 9383 9384 9385
Comp 286 = 9389 9390 9391 9392
Comp 287 = 9394 9395
Comp 288 = 9396 9397
Comp 289 = 9401 9402 9403 9404 9701 9702
Comp 290 = 9405 9406
Comp 291 = 9408 9409
Comp 292 = 9413 9414 9415 9416 9417 9418
Comp 293 = 9419 9420
Comp 294 = 9428 9429
Comp 295 = 9435 9436 9437 9438
Comp 296 = 9440 9441 9442
Comp 297 = 9443 9444
Comp 298 = 9446 9447
Comp 299 = 9448 9449 9450 9451
Comp 300 = 9452 9453 9454 9455 9456
Comp 301 = 9457 9458
Comp 302 = 9460 9461
Comp 303 = 9462 9463 9464
Comp 304 = 9465 9466 9467 9468 9469 9470 9471 9472 9473
Comp 305 = 9474 9475
Comp 306 = 9477 9478
Comp 307 = 9479 9480 9481 9482 9483
Comp 308 = 9487 9488
Comp 309 = 9494 9495 9496 9497
Comp 310 = 9498 9499
Comp 311 = 9500 9501 9502 9503

Comp 312 = 9506 9507
Comp 313 = 9515 9516 9517 9518
Comp 314 = 9519 9520
Comp 315 = 9521 9522
Comp 316 = 9527 9528
Comp 317 = 9529 9530 9531 9532
Comp 318 = 9534 9535 9536 9537
Comp 319 = 9539 9540
Comp 320 = 9542 9543
Comp 321 = 9545 9546
Comp 322 = 9549
Comp 323 = 9554 9555
Comp 324 = 9570 9571 9572
Comp 325 = 9573 9574
Comp 326 = 9581 9582
Comp 327 = 9583 9584
Comp 328 = 9585 9586 9587
Comp 329 = 9590 9591
Comp 330 = 9600
Comp 331 = 9602 9603
Comp 332 = 9604 9605 9606
Comp 333 = 9607 9608
Comp 334 = 9609 9610
Comp 335 = 9611 9612
Comp 336 = 9613 9614
Comp 337 = 9615 9616
Comp 338 = 9617 9618
Comp 339 = 9620 9621 9622
Comp 340 = 9625 9626
Comp 341 = 9627 9628
Comp 342 = 9633 9634
Comp 343 = 9636 9637 9638
Comp 344 = 9639 9640 9641
Comp 345 = 9642 9643
Comp 346 = 9646 9647
Comp 347 = 9652 9653
Comp 348 = 9654 9655
Comp 349 = 9658 9659
Comp 350 = 9660 9661
Comp 351 = 9663 9664
Comp 352 = 9667 9668
Comp 353 = 9671 9672
Comp 354 = 9673 9674 9675
Comp 355 = 9680 9681 9682 9683

Comp 356 = 9685 9686
Comp 357 = 9687 9688
Comp 358 = 9689 9690
Comp 359 = 9691 9692 9693
Comp 360 = 9696 9697 9698
Comp 361 = 9704 9705 9706
Comp 362 = 9707 9708
Comp 363 = 9709 9710
Comp 364 = 9711 9712
Comp 365 = 9715 9716
Comp 366 = 9718 9719
Comp 367 = 9720 9721
Comp 368 = 9722 9723
Comp 369 = 9724 9725
Comp 370 = 9727 9728
Comp 371 = 9729 9730
Comp 372 = 9731 9732 9733
Comp 373 = 9735 9736
Comp 374 = 9737 9738 9739 9740
Comp 375 = 9741 9742
Comp 376 = 9743 9744
Comp 377 = 9747 9748
Comp 378 = 9749 9750 9751
Comp 379 = 9752 9753
Comp 380 = 9754 9755
Comp 381 = 9756 9757 9758 9759
Comp 382 = 9760 9761
Comp 383 = 9762 9763
Comp 384 = 9764 9765
Comp 385 = 9766 9767
Comp 386 = 9768 9769
Comp 387 = 9770 9771
Comp 388 = 9772 9773
Comp 389 = 9774 9775
Comp 390 = 9777 9778 9779
Comp 391 = 9783 9784
Comp 392 = 9785 9786
Comp 393 = 9787 9788
Comp 394 = 9789 9790
Comp 395 = 9791 9792
Comp 396 = 9793 9794 9795
Comp 397 = 9796 9797
Comp 398 = 9798 9799
Comp 399 = 9802 9803 9804

Comp 400 = 9805 9806
Comp 401 = 9807 9808 9809
Comp 402 = 9810 9811
Comp 403 = 9812 9813
Comp 404 = 9814 9815
Comp 405 = 9816 9817 9818 9819
Comp 406 = 9820 9821
Comp 407 = 9822 9823 9824
Comp 408 = 9825 9826
Comp 409 = 9827 9828
Comp 410 = 9831 9832
Comp 411 = 9834 9835
Comp 412 = 9836 9837
Comp 413 = 9838 9839
Comp 414 = 9840 9841 9842 9843
Comp 415 = 9844 9845 9846 9847
Comp 416 = 9848 9849
Comp 417 = 9850 9851 9852
Comp 418 = 9853 9854
Comp 419 = 9855 9856
Comp 420 = 9857 9858
Comp 421 = 9859 9860
Comp 422 = 9861 9862 9863
Comp 423 = 9864 9865
Comp 424 = 9866 9867
Comp 425 = 9868 9869
Comp 426 = 9870 9871
Comp 427 = 9872 9873
Comp 428 = 9874 9875 9876
Size of smallest CC = 1
Size of largest CC = 8638
8638/9877 = 0.87456

The HepTh graph contains 429 connected components. The smallest connected component contains one vertex, and largest component contains 8638 vertices.

The largest component contains 8638 of the 9877 total vertices in the graph. 8338/9877 = 0.87456, so the largest component contains 87.456% of the graph's vertices. Thus, the graph does contain a giant component.

6. Command line: java TopRank CA-HepTh-grap.txt

Rank	Vertex	DegCen
1	86	130
2	15	120
3	54	118
4	920	112
5	163	108
6	35	106
7	1546	106
8	1420	102
9	1873	102
10	8	100
11	38	100
12	2	98
13	277	98
14	67	92
15	40	90
16	1907	90
17	220	88
18	252	88
19	1305	88
20	32	86
21	386	86
22	442	86
23	590	86
24	1	84
25	62	84
26	141	84
27	1094	84
28	1332	84
29	26	82
30	76	82
31	1904	82
32	13	80
33	527	80
34	1461	80
35	1687	80
36	582	78
37	851	78
38	1231	78
39	36	76
40	173	76

7. Command line: java TopRank CA-HepTh-graph.txt

Rank	Vertex	CloCen
1	15	4.045849
2	920	4.074215
3	40	4.095751
4	38	4.121107
5	1	4.147042
6	1461	4.154799
7	1094	4.163946
8	54	4.168924
9	1332	4.169851
10	442	4.183281
11	13	4.190228
12	8	4.19486
13	163	4.195323
14	30	4.211879
15	17	4.213268
16	62	4.221605
17	35	4.222184
18	277	4.222415
19	1420	4.228551
20	429	4.229015
21	14	4.240245
22	67	4.242446
23	1873	4.247771
24	582	4.248582
25	386	4.249508
26	31	4.250434
27	26	4.253676
28	252	4.259697
29	87	4.264675
30	1443	4.264675
31	141	4.269191
32	851	4.272548
33	171	4.274517
34	618	4.278685
35	76	4.2832
36	173	4.291884
37	413	4.296515
38	590	4.298483
39	1546	4.300104
40	24	4.310293

8. As stated in my answer to question 4, the HEP-TH graph is more closely related to a small-world graph than a scale-free graph. This is because the correlation coefficient of the exponential function fit of the degree distribution was stronger than that of the power function fit. However, the graph appears to exhibit properties of a scale-free graph as well a small-world graph.

   My results from question 5 show that most of the vertices are indeed connected to just a few nearby vertices, but that a few are connected to distant random vertices. This is to be expected in a small-world graph.

   My answer to question 4 shows that there are many vertices of low degree and a few vertices of high degree, known as hubs. This is a characteristic of a scale-free graph. This characteristic is not as strong as the properties of the small-world graph, however. This accounts for the fact that the correlation coefficient is stronger towards a small-world graph.

   Based on my results, it appears that many high energy physics theory researchers collaborated on a few papers with other nearby researchers, and a few of them collaborated on a smaller number of papers with researchers to whom they were distantly connected.

9. In this project I learned a lot about programmatic analysis of graphs. I found it beneficial to learn how it is useful to fit a dataset to logarithmic functions. I thought it was fascinating to see how so much information can be drawn from the same set of data by looking at the data in a few different ways. I also learned better how program design so greatly affects the ability of the program to process large amounts of data. Originally, I was concerned with running out of time and memory, but then I redesigned some parts of the programs to more efficiently handle the volume of data.