CREU 2016 - 2017 Final Report: Prime Labelings of Hypercube Graphs

Cayla McBee, PhD, Providence College
Linda Wilkens, DSc, Providence College
Michelle Cruz, Providence College
Kayla Johnson, Providence College
Giana Solviletti, Providence College

I) Goals and Purpose

This CREU project arose from a collaboration which began at a Research Experiences for Undergraduate Faculty (REUF) workshop in 2012. During the workshop topics suitable for undergraduate research and student faculty collaboration were presented. One such topic included prime labelings of graphs.

A graph labeling is an assignment of values to the vertices or edges of a graph subject to certain constraints. Many different graph labeling problems exist - our work focuses on prime labelings. A *coprime* labeling is a labeling of the vertices by distinct integers in such a way that the labels of any two adjacent vertices are relatively prime. If a graph with n vertices has a coprime labeling by the integers $1, 2, 3, \ldots, n$ then the labeling is said to be *prime*.

The CREU project completed used computing along with knowledge of graph theory to help determine which hypercube graphs, Q_k, are prime.

II) Related Work

Interest in prime labeling problems first began in the early 1980s when Entringer and Tout first conjectured that all trees are prime [2]. Since then, much work has been done trying to determine which classes of graphs are prime [1]. Despite the numerous publications in the area of prime labelings, there are no publications specifically looking at prime labelings of hypercube graphs.

There is a long history of computing being used to help solve difficult problems in graph theory. For example, the Traveling Salesman Problem has given rise to numerous heuristics and exact algorithms. The area of graph theory is rich in problems that are easy to state and may even be easy to solve for small cases, but which become extremely challenging for graphs on a large number of vertices. Prime labeling problems fall into this category. For certain graphs, computing can be used to determine whether a prime labeling exists. It can also be used along with mathematical knowledge to determine whether a particular class of graphs is prime.

III) **Process**

There were three main components to this project. The first component involved students learning the necessary background in the area of graph theory and more specifically learning about the structure and characteristics of hypercube graphs.

The second component of the project involved creating an algorithm that searches for prime labelings of hypercube graphs. The hypercube, Q_k, has 2^k vertices and therefore an algorithm that tests every possible labeling to find a prime labeling has efficiency order $O(2^k!)$. For this reason an exhaustive search for prime labelings is not efficient or practical. A more efficient algorithm needed to be developed to check for prime labelings for larger values of k. We used their knowledge of C++ to implement algorithms that could potentially find a prime labeling. We first started by creating a program that utilized a brute force search method in order to find a prime labeling for graphs up to Q_3. For larger values of k another approach was needed.

One algorithm that we came up with has worked for values of k through $k = 8$ and involves partitioning the labels according to common divisors. The computer program creates sets for the labels $1, \ldots, 2^k$ with the same prime divisors. The first step of the algorithm is to determine the possible adjacencies by determining the hamming distance between each pair of vertices. Next, it finds the prime divisors of each label and stores them. Once that is completed, the sets are created, which groups the labels with the same prime divisors. Conflict sets are those that contain labels that are not relatively prime to those within a set. Candidate sets are those that contain labels that are relatively prime to those within a set. These conflict and candidate sets are stored as well. The program then tries to place a label beginning with label 1. If a label works, it tries another label from that set's candidate sets. If a label does not work, the replacement label cannot come from that set or from that set's conflicting sets. Unfortunately, this program takes several weeks to complete for $k = 8$ so as the hypercube graphs become larger, searching still remains an issue.

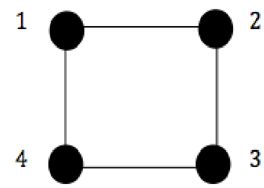
The third component of the project involved using knowledge gained from determining which graphs are prime along with mathematical knowledge to attempt to prove theorems regarding the values of k for which Q_k is prime.

IV) **Results and Discussion**

During our research experience, we were able to determine whether a hypercube graph Q_k is prime for $k = 1, 2, 3 \ldots 8$. The computer algorithm concluded there are no prime labelings for the hypercube graphs Q_4 and Q_6. We were able to verify prime labelings for the hypercube graphs Q_2, Q_3, Q_5, Q_7, and Q_8. The prime labelings pictured below were generated by the computer algorithm described above. Our main results are given below.

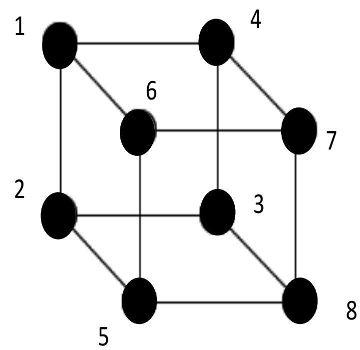
Theorem 1. *Q_2 is prime.*

The below graph shows an example of a possible prime labeling of the hypercube graph Q_2.



Theorem 2. *Q_3 is prime.*

The below graph shows an example of a possible prime labeling of the hypercube graph Q_3.



In order to prove there is no prime labeling for Q_k the following lemma is needed.

lemma 1. *Any two vertices in Q_k have 0 or 2 common neighbors.*

Proof. Let Q_k be a hypercube graph with $2^k = n$ vertices. By Lemma 1, Q_k is bipartite. This implies that Q_k has a vertex set that can be partitioned into two nonempty sets A and B each of order $n/2$ such that each vertex has degree k. Let all the vertices of Q_k be labeled with binary strings of length k. Without loss of generality, let the vertices in partition A be labeled by binary strings with an even number of ones. Then, the vertices in partition B have an odd number of ones in their binary string labelings.

Case 1. *Let x, y be two vertices in Q_k with hamming distance 1. Then x, y are adjacent. Suppose x and y have a common neighbor. Then, Q_k would contain C_3 as a subgraph. Cycles of odd length are not permitted in bipartite graphs. Therefore, x and y have exactly 0 common neighbors.*

Case 2. *Let x, y be vertices with hamming distance 2. Notice that x and y are in the same partition of the bipartite graph. Without loss of generality, let x and y be in partition A. The degree of x is k and the degree of y is k, that is to say x and y each have k possible vertices to be adjacent to in partition B. Let x be labeled with the binary string $v_1 v_2 \dots v_i \dots v_j \dots v_{k-1} v_k$ and let y be labeled with the binary string $v_1 v_2 \dots w_i \dots w_j \dots v_{k-1} v_k$, where i and j are integers between 1 and k and each bit in the sequences for x and y are either 0 or 1. Since x and y have hamming distance 2, x and y differ in the i^{th} and j^{th} positions. The only common neighbors of x and y are $v_1 v_2 \dots v_i \dots w_j \dots v_{k-1} v_k$ and $v_1 v_2 \dots w_i \dots v_j \dots v_{k-1} v_k$ which are produced by changing bit i or j. Therefore, if the hamming distance of two vertices in Q_k is 2, then there are exactly 2 common neighbors.*

Case 3. *Let x, y be vertices with hamming distance greater than* 2*, that is to say the hamming distance is greater than or equal to* 3*. Then, x and y differ in at least three bits. There is no vertex that differs in exactly one bit to x that also differs in one bit to y. Therefore, if the hamming distance of two vertices in Q_k is greater than* 2*, then they have exactly* 0 *common neighbors.*

Therefore, any two vertices in Q_k have 0 or 2 common neighbors. □

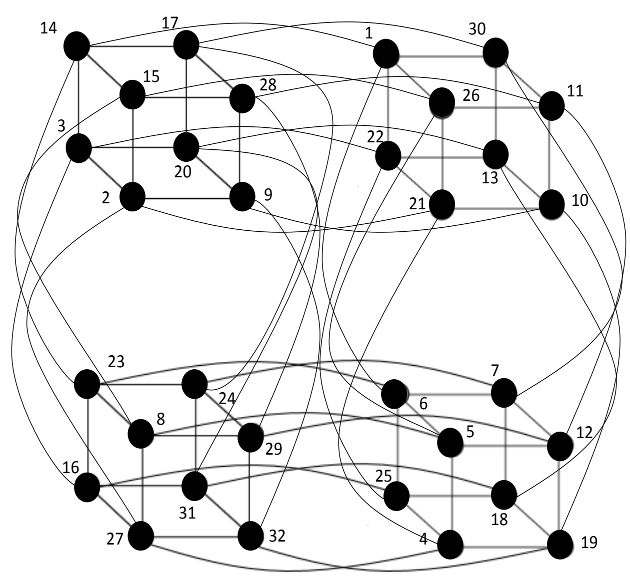
The following proof that Q_4 is not prime utilizes the bipartite structure of hypercube graphs combined with the common neighbor lemma to reach a contradiction when placing the labels, specifically the labels that are multiples of three. This contradiction is a result of the large number of multiples of three that need to be placed.

Theorem 3. *Q_4 is not prime.*

Proof. Let Q_4 be a hypercube graph with $2^4 = 16$ vertices. A prime labeling of Q_4 is a labeling of the vertices by 1 through 16 such that the labels of two adjacent vertices are relatively prime. Suppose there exists a prime labeling of the vertices by the numbers 1 through 16. Notice that 6 and 12 can only be adjacent to the vertices labeled $1, 5, 7, 11$, and 13. By Lemma 2, two vertices either have 0 or 2 common neighbors. If 6 and 12 have two common neighbors, then the vertices labeled 6 and 12 have six distinct vertices that are either adjacent to the vertex with label 6 or the vertex with label 12. If 6 and 12 have zero common neighbors, vertices labeled 6 and 12 have eight distinct vertices that are either adjacent to the vertex with label 6 or the vertex with label 12. There are not enough possible labels for the vertices adjacent to labels 6 and 12. Therefore, Q_4 is not prime. □

Theorem 4. *Q_5 is prime.*

The graph below gives a prime labeling of the hypercube graph Q_5.



In order to prove that Q_6 is not prime a few preliminary results are necessary. First recall that in a hypercube graph Q_k where the vertices are assigned unique binary sequences of length k, two vertices are adjacent if and only if their binary sequences differ by exactly one bit. Therefore, none of the vertices whose binary sequences contain an even (similarly, odd) number of 1s will be adjacent. In a prime labeling, none of the even labels can be adjacent. As a result, we are choosing to label vertices corresponding to binary sequences with an even (similarly, odd) number of 1s with even (odd) labels.

lemma 2. *In a hypercube Q_n, if vertices v_1 and v_2 each have exactly k 1s and differ by exactly two bits, then both v_1 and v_2 are adjacent to two and only two vertices u_1 and u_2, where exactly one of the vertices u_1 or u_2 has $k-1$ 1s and the other has $k+1$ 1s for $1 \leq k \leq n-1$.*

Proof. Let $v_1 = a_1, a_2, \ldots, a_i \ldots, a_j, \ldots, a_{n-1}, a_n$ and $v_2 = a_1, a_2, \ldots, \bar{a}_i, \ldots, \bar{a}_j, \ldots, a_{n-1}, a_n$, where $1 \leq i \leq n, 1 \leq j \leq n, i \neq j$, the a terms represent a binary value of 0 or 1, and the bar signifies one bit change from 0 to 1 or vice versa. Two vertices are adjacent if and only if their binary sequences differ by exactly one bit, therefore $u_1 = a_1, a_2, \ldots, \bar{a}_i, \ldots, a_j, \ldots, a_{n-1}, a_n$ and $u_2 = a_1, a_2, a_i, \ldots, \bar{a}_j, \ldots, a_{n-1}, a_n$. Then, there are two possible cases for the possible number of 1s that u_1 and u_2 can contain, which are evident in the table below.

a_i	a_j	$\bar{a}_i$	$\bar{a}_j$	**# 1s in** u_1	**# 1s in** u_2
0	1	1	0	k+1	k-1
1	0	0	1	k-1	k+1

Therefore, either u_1 or u_2 has $k-1$ 1s and the other u term has $k+1$ 1s.

□

lemma 3. *In Q_6, any set of 10 vertices in one bipartite set must be adjacent to at least 22 vertices in the other bipartite set.*

Proof. Let the 64 vertices be partitioned into two bipartite sets called U and V, each containing 32 vertices. Let U be the set of vertices whose binary sequences contain an odd number of 1s and let V be the vertices in Q_6 whose binary sequences contain an even number of 1s. Let U' and V' be subsets of U and V, respectively. Without loss of generality, consider the vertex labeled $000000 \in V$ and add it to V'. 000000 must be adjacent to the six vertices in U that have exactly one 1 in their binary sequences. Add those six vertices in U to U'. Those six vertices that were just added to U' must then be adjacent to the vertices in V that contain exactly two 1s in their binary sequences. Since each of the binary sequences in Q_6 consists of six digits from which to choose two 1s, there are $\binom{6}{2}$ = 15 vertices whose sequences have exactly two 1s. Add these fifteen vertices to the set V'. Those same six vertices in U' are also adjacent to 000000 because they have exactly one 1 in their binary sequences. Therefore, there are six vertices in set U' that are adjacent to the sixteen vertices in V'. Let four more vertices be added to the six vertices in U' while trying to find the fewest number of vertices that these four can be adjacent to in V'. There are two cases to consider.

Case 1. *Consider the case where any of the four vertices in U' have five 1s in its binary sequence. Let the vertex with five 1s in its binary sequence be called w. Then, all of w's six neighbors must be added to V', which adds 6 vertices to the 16 that are already in V'. Consequently, there are 22 neighbors in V' with three remaining vertices in U' that haven't even been considered yet.*

Case 2. *Consider the case where all four vertices have exactly three 1s in their binary sequences. Then, each of these four vertices has exactly three neighbors with two 1s in their binary sequences,* *which are already included in the sixteen vertices in V'. These four vertices also have three neighbors with exactly four 1s in their binary sequences. By Lemma 3, any two of the four vertices can have exactly one neighbor in common that has four 1s in its binary sequence. Therefore, at least $\binom{4}{2} = 6$ vertices must be added to V'. It follows that each of these four vertices are paired with each of the remaining three vertices, which establishes the required three neighbors for each vertex with exactly four 1s in the binary sequence.*

U' now has 10 vertices in total, which includes the six with exactly one 1 in their binary sequences and the four with three 1s in their binary sequences. These 10 vertices are adjacent to 22 vertices in V', which includes the vertex 000000, the fifteen with exactly two 1s in their binary sequences, and the six with four 1s in their binary sequences. This represents the fewest number of adjacencies that can occur in Q_6. □

Considering these two cases, the following proof outlines the contradiction that shows why Q_6 is not prime.

Theorem 5. *Q_6 is not prime.*

Proof. Suppose there exists a prime labeling of the vertices of Q_6 by 1 through 64. Let all of the odd numbers label the vertices in U and all of the even numbers label the vertices in V. Then, there are 10 even multiples of three and 11 odd multiples of three that must be placed into their respective bipartite sets such that none of them are adjacent. By Lemma 4, the 10 even multiples of three will be adjacent to at least 22 odd-labeled vertices. Since there are 32 vertices in each bipartite set, this leaves 10 vertices with which to label 11 odd multiples of three, which is clearly impossible. Therefore, Q_6 is not prime. □

Theorem 6. *Q_7 is prime.*

A prime labeling of the hypercube graph Q_7 can be found in the appendix.

Theorem 7. *Q_8 is prime.*

A prime labeling of the hypercube graph Q_8 can be found in the appendix.

V) **Future Work**

Over the past two semesters our CREU group has been able to determine whether or not Q_k is prime for $1 \leq k \leq 8$. To the best of our knowledge results for $k > 3$ are unpublished and worth noting. Our immediate plans include writing up our results and submitting them for publication. Currently we are planning to submit our manuscript to a peer reviewed journal which publishes papers on graph labeling problems.

Although we were able to establish whether or not a prime labeling exists for Q_k for k up to 8, we were unable to determine any general results for when Q_k is prime. Based on our work it seems this may be a difficult problem that is related to some well-known and open problems in number theory. Therefore, a general result providing conditions for when Q_k is prime for any k may be unlikely. However, despite these challenges the mentors are optimistic that it is possible to determine whether or not Q_k is prime for larger values of k and that it may also be possible to prove results that give certain conditions necessary for a prime labeling to exist.

VI) **Web Links**

The link for our project website is:
`http://library.providence.edu/dps/creu/plhg/`

VII) **Presentations and Publications**

- Joint Mathematics Meetings Student Poster Session, Atlanta, GA, January 6, 2017.
  - JMM is the largest annual mathematics meeting in the world. During the undergraduate poster session over 300 student posters were presented and judged by mathematicians.
- Annual Celebration of Student Scholarship Creativity, Providence College, Providence, RI, April 26, 2017.
  - This event showcases the scholarly, creative, and service work that Providence College students are doing. The Celebration features students from all class years and from many different departments and disciplines.
- Math and Computer Science Student Talks, Providence College, Providence, RI, May 8, 2017.
  - The CREU research group along with two other students will give a talk at Providence College discussing their work over the past two semesters.

1 Prime Labeling For Q_7

Vertex	Label
1001000	96
1001001	53
1001010	83
1001011	76
1001100	107
1001101	92
1001110	38
1001111	39
1000000	67
1000001	108
1000010	102
1000011	71
1000100	114
1000101	73
1000110	113
1000111	62
1011000	89
1011001	2
1011010	124
1011011	45
1011100	104
1011101	57
1011110	51
1011111	58
1010000	120
1010001	101
1010010	97
1010011	116
1010100	103
1010101	100
1010110	80
1010111	63
0001000	35
0001001	12
0001010	6
0001011	25
0001100	18
0001101	5
0001110	13
0001111	14

Vertex	Label
0000000	24
0000001	85
0000010	7
0000011	36
0000100	11
0000101	42
0000110	30
0000111	17
0011000	48
0011001	55
0011010	65
0011011	28
0011100	19
0011101	56
0011110	46
0011111	3
0010000	49
0010001	26
0010010	54
0010011	23
0010100	60
0010101	121
0010110	77
0010111	10
0101000	66
0101001	115
0101010	95
0101011	22
0101100	125
0101101	52
0101110	44
0101111	9
0100000	29
0100001	84
0100010	72
0100011	37
0100100	78
0100101	41
0100110	47

Vertex	Label
0100111	20
0111000	31
0111001	98
0111010	34
0111011	15
0111100	88
0111101	27
0111110	21
0111111	68
0110000	90
0110001	59
0110010	61
0110011	112
0110100	119
0110101	50
0110110	40
0110111	33
1101000	43
1101001	4
1101010	64
1101011	69
1101100	82
1101101	81
1101110	75
1101111	8
1100000	126
1100001	79
1100010	109
1100011	70
1100100	1
1100101	110
1100110	118
1100111	87
1111000	16
1111001	93
1111010	99
1111011	74
1111100	111
1111101	86

Vertex	Label
1111110	106
1111111	105
1110000	127
1110001	94
1110010	32
1110011	117
1110100	122
1110101	91
1110110	123
1110111	128

2 Prime Labeling For Q_8

Vertex	Label
0	3
1	182
2	238
3	9
4	22
5	27
6	81
7	44
8	88
9	243
10	15
11	176
12	45
13	26
14	242
15	75
16	52
17	135
18	225
19	104
20	21
21	38
22	34
23	63
24	147
25	208
26	68
27	189
28	136
29	35
30	175
31	76
32	70
33	33
34	99
35	140
36	39
37	170
38	190
39	117

Vertex	Label
40	51
41	130
42	14
43	153
44	154
45	19
46	221
47	28
48	57
49	46
50	56
51	171
52	110
53	91
54	13
55	220
56	230
57	77
58	143
59	10
60	247
61	6
62	12
63	169
64	62
65	69
66	207
67	20
68	87
69	152
70	82
71	93
72	105
73	58
74	116
75	111
76	124
77	55
78	65

Vertex	Label
79	98
80	123
81	232
82	248
83	129
84	92
85	187
86	209
87	40
88	74
89	119
90	133
91	50
92	245
93	18
94	24
95	253
96	141
97	80
98	100
99	159
100	164
101	161
102	203
103	160
104	148
105	217
106	11
107	200
108	85
109	36
110	48
111	121
112	184
113	7
114	17
115	250
116	49
117	54

Vertex	Label
118	72
119	23
120	29
121	96
122	108
123	31
124	144
125	95
126	115
127	162
128	86
129	165
130	177
131	172
132	183
133	94
134	188
135	195
136	201
137	106
138	212
139	213
140	118
141	37
142	41
143	112
144	219
145	122
146	244
147	231
148	236
149	43
150	53
151	134
152	142
153	47
154	59
155	146
156	61

Vertex	Label
157	192
158	216
159	145
160	237
161	178
162	194
163	249
164	166
165	67
166	71
167	196
168	202
169	73
170	79
171	224
172	89
173	30
174	60
175	83
176	206
177	97
178	101
179	158
180	107
181	90
182	120
183	103
184	109
185	150
186	180
187	113
188	240
189	127
190	1
191	42
192	255
193	214
194	218
195	131
196	226
197	155

Vertex	Label
198	185
199	254
200	2
201	205
202	215
203	4
204	235
205	84
206	126
207	5
208	8
209	25
210	137
211	16
212	139
213	168
214	252
215	149
216	151
217	66
218	132
219	157
220	198
221	125
222	163
223	78
224	32
225	167
226	173
227	64
228	179
229	156
230	234
231	181
232	191
233	102
234	204
235	193
236	114
237	197
238	199

Vertex	Label
239	228
240	211
241	138
242	174
243	223
244	186
245	227
246	229
247	210
248	222
249	233
250	239
251	246
252	241
253	128
254	256
255	251

References

[1] Joseph A. Gallian. A dynamic survey of graph labeling. *The Electronic Journal of Combinatorics*, (DS6), December 2015.

[2] A. Tout, A.N. Dabboucy, and K. Howalla. Prime labelings of graphs. *National Academy Science Letters*, 11:365–368, 1982.