The FM Labeling Problem

1 Problem Statement:

Given a connected graph G, where each vertex v_i of G is assigned a nonnegative integer value $F(v_i)$, find the minimum value that $max(F(v_i))$ must take such that G satisfies the following condition:

$$d(x, y) + |F(x) - F(y)| > diam(G)$$

where $d(x, y)$ denotes the distance between vertices x and y and $diam(G)$ is the greatest distance in graph G.

2 Definitions:

1. Define the *maximum vertex* of G to be the vertex of greatest value, and likewise define the *minimum vertex* of G to be the vertex of least value.
2. Let the function $f : G \longrightarrow \mathbb{Z}^+$ map G to the value of $max(F(v_i))$.
3. Define the *wiggle room* between two vertices x and y to be the quantity

$$d(x, y) + |F(x) - F(y)| - (diam(G) + 1).$$

We denote the wiggle room between x and y as $wr(x, y)$.
4. Vertices x and y have a *tight fit* if $wr(x, y) = 0$.
5. Define the *tightness graph* of G to be the graph of G where all edges are removed, and and for each pair of vertices such that $wr(x, y) = 0$, an edge is inserted.
6. Define a *valid assignment* to be an assignment of values to the vertices of G such that they satisfy the problem condition.
7. Define a *solution* of G, denoted S, to be a *valid assignment* such that the solution has $max(F(v_i)) = f(G)$.

3 Observations:

All $F(i)$'s must be distinct. *Proof by Contradiction:* Assume that there exists two vertices such that $F(x) = F(y)$. Then,

$$d(x, y) + |F(x) - F(y)| > diam(G) \Rightarrow d(x, y) > diam(G).$$

This is a contradiction, as $d(x, y) \leq diam(G)$.

A General Bound for $f(G)$: Consider any graph G with n vertices. Start from an arbitrary vertex, and label the n vertices $0 \cdot diam(G)$, $1 \cdot diam(G)$, ... ,$(n - 1) \cdot diam(G)$. This satisfies the given property, and thus

$$f(G) \leq diam(G) \cdot (n - 1). \quad (1)$$

Furthermore, from the previous section, we know that all vertices must be distinct. Thus,

$$(n - 1) \leq f(G) \leq diam(G) \cdot (n - 1). \quad (2)$$

The Inverse Solution Given a solution S_1 of G, we may construct another equivalent solution S_2 of G by letting each $F(v_i)$ on vertex v_i of S_1 map to $f(G) - F(v_i)$ on each vertex v_i on S_2. This new assignment is also valid, as $d(x,y) + |(F(G) - F(x)) - (F(G) - F(y))| = d(x,y) + |F(x) - F(y)| > diam(G)$, and the maximum vertex of S_1 is the same as the maximum vertex of S_2. We call the equivalent solution S_2 the **inverse** of solution S_1.

4 Preliminary Examples:

The Complete Graph: Consider all graphs G with $diam(G) = 1$ on n vertices. The only such graph is the K_n graph. We may label the vertices, in any order, $0, 1, 2...n-1$. This satisfies the relation, and is also the lower bound for the graph. Thus, $f(G) = n - 1$ for a K_n

The Complete n-partite Graph: Every complete n-partite graph $K_{a_1,a_2,...,a_m}$ has $diam(K_{a_1,a_2,...,a_m}) = 2$. We also see that any two vertices x and y which are in the same partition a_k have a wiggle room of 0 when $|F(x) - F(y)| = 1$. Then, we label the vertices in one partition from 0 to $a_1 - 1$, and then label the next vertex in a new partition such that it has wiggle room of 0 with the the maximum vertex of the previous partition. Repeating this process over all partitions yields $f(G) = n + m - 2$.

5 More Observations:

A Flawed Algorithm: When looking at the path graph, an interesting algorithm to try to construct a solution is as follows:

Label the $\lfloor \frac{n+1}{2} \rfloor^{th}$ vertex 0. Then, create an increasing tightness sequence from vertex $\lfloor \frac{n+1}{2} \rfloor$ to vertex $\lfloor \frac{n+1}{2} \rfloor - 1$ to vertex $\lfloor \frac{n+1}{2} \rfloor - 2$... to vertex 1, and then to vertex $\lfloor \frac{n+1}{2} \rfloor + 1$. Then, create a decreasing tightness sequence from vertex $\lfloor \frac{n+1}{2} \rfloor + 1$ to vertex $\lfloor \frac{n+1}{2} \rfloor + 2$ to vertex $\lfloor \frac{n+1}{2} \rfloor + 3$ ... to vertex n. The maximum vertex here is vertex $\lfloor \frac{n+1}{2} \rfloor + 1$, with value $(n-2)\lfloor \frac{n+1}{2} \rfloor + 1$

Suspicions about Paths So far, we have found a unique solution for all paths with an even number of vertices, and a plethora of solutions for paths with odd numbers of vertices. The solution for P_n for n even follows the above algorithm, and yields

$$f(P_n) = (n-2)\lfloor \frac{n+1}{2} \rfloor + 1 \approx \binom{n}{2}.$$

The winning solution for P_n and n odd is less clear. We suspect it is like the even case in that the solution is $\binom{n}{2} - an + b$ where a, n are real numbers.

A general upper bound for cycles, (although almost certainly not $f(G)$) We may think of a cycle as two paths of length $\lceil \frac{n+1}{2} \rceil$, as a path of this length has equal diameter to the cycle. Then, we label around the graph 0, D, $2D$, ... $(\lceil \frac{n+1}{2} \rceil - 1)D$, 1, $D + 1$, $2D + 1$, ... , $(\lceil \frac{n+1}{2} \rceil - 1)D + 1$. This labeling works, and consequently,

$$f(G) = (\lceil \frac{n+1}{2} \rceil - 1)D + 1 = (\lceil \frac{n+1}{2} \rceil - 1)^2 + 1$$

Tightness Graphs We may prove that the tightness graph of any connected G contains a path starting from min walking to max. Let this walk from min to max be called a *hopping* on G. We may use these hoppings to determine lower bounds of $f(G)$. However, not every vertex is part of in the hopping from min to max. In this hopping, each hop has a value associated with it, which is the value by which the maximum vertex increases throughout the hopping... We know that there is 1 hop with value 1, 2 hoppings with value 2, 3 hoppings with value 3 and so on. However, a greedy algorthim does not necessarily work, as grabbing the hop with value 1 is not always the best.

Diameter Manipulation Tables One problem with solving the path problem is that the diameter changes with every increase in vertices. Thus, we manipulate the diameters in the following table, and document the $f(G)$ max label in every case.

$$d(x, y) + |F(x) - F(y)| > D'$$

0	0	0	0	0	0	0
1	2	3	4	5	6	7
1	3	5	7	9	11	13
1	3	5	8	11	14	17
1	4	6	10	13		
1	4			13		
1	4				20	
1	4					25

Where the rows denote the number of vertices in the path, and the columns denote the D' value.

Tight Hopping - Basic Rules How may we hop on a sequence of vertices on the path graph, such that each vertex we hop to is tight with the previous vertex? There are two ways a vertex could be tight. If we jump from v_k to v_{k+1}, it could be that $v_k > v_{k+1}$ or that $v_k < v_{k+1}$. We define an increasing hop to be when $v_k < v_{k+1}$ and a decreasing hop when $v_k > v_{k+1}$. We find that

1) if the label is increasing, and we hop k vertices to the right/left and j more vertices in the same direction, this is always valid.

2) if the label is increasing, and we hop k vertices to the right/left and j more vertices in the opposite direction with $j < k$, we find that $\frac{j}{2} \leq n$.

3) if the label is increasing, and we hop k vertices to the right/left and j more vertices in the opposite direction with $j > k$, we find that $\frac{k}{2} \leq n$.

4) if the label is decreasing, and we hop k vertices to the right/left and j more vertices in the same direction, we find that it is only valid when $\frac{j}{2} \geq n$.

5) if the label is decreasing, and we hop k vertices to the right/left and j more vertices in the opposite direction, this is never valid.

6 Questions:

There is the question of "descendents" and family lines of graphs with certain diameters.

Let the tightness graph be denoted $t(G)$

1. What is the maximum number of edges on a tightness graph?
2. What different structures can be described by the tightness graph?
3. For odd number of vertices what is different? Why is there only 1 solution for even numbered vertices? (as we know it?) Do all even numbers follow a fixed tightness graph structure?
4. Can we prove that the hopping path (ie diameter of the tightness graph) is strictly increasing from min to max? We know that it is not true that every vertex is in this path of course. 5. Find the pattern of variance from $\binom{n}{2}$ in the even and odd numbers case.

6. Fix the diameter table/chart to accomodate the new understanding of odd paths. Also, include the diameter/vertex table on this PDF.
7. Investigate "descendency" among solutions of various (D, V) cases. Find their $t(G)$'s too.
8. When not every vertex contributes to the increasing diameter sequence of the tightness graph, why is the max the same as in those chain examples?
9. What are the rules of tightness graphs? what sequences are you allowed or not allowed to hop to?

1.0. Is hopping sequence number recursive? Also, hopping sequences are PARTITIONS of the max. The number of partitions can vary, with vertices of degree 3.
1.1. Investigate vertices of degree 3.
1.2. Come up with some rules of valid jumpings.
1.3. Are there classes of graphs that yield a certain tightness graph, or do tightness graphs have a 1-1 correspondence with original graphs?

7 Testing Methodology

7.1 By Hand

7.1.1 Fix the max and the min

If we have a conjecture as to what $f(G)$ is, we may find two vertices and label one the min and the other the max. Having two values to start each case reduces the computations quite a bit. In general, one must deal with around $\frac{1}{2} \cdot \binom{n}{2}$ cases to exhaust all possibilities. This is not too bad for small paths of length n

7.1.2 Hop the Graph!

It remains to be proved whether this experimental method works. The idea is to start on a vertex, and to use increasing hops to hop through all the vertices. The last vertex hopped to will the maximum vertex. This method works and achieves solutions for $2, 3, 4, 5, 6, 7, 8, 10, 11, 12$ vertices on the path, but no solution has been found using this method for 9 vertices.

7.2 Using a Computer

The following JavaScript code excerpts are used to collect data about the path case. The path is represented as an array.

This function verifies whether or not a given path labeling satisfies the relationship.

```
function isgood(arr) {
        var n = arr.length - 1;
        var result = true;

        for (var i=0; i<=n; i++) {
                for (var j=0; j<=n; j++) {
                        if (j <= i) continue;
                        var d = arr[j] - arr[i];
                        if (d<0) d = -d;
                        if (d + j - i <= n) result = false;
                }
```

```
        }
        return result;
}
```

This function checks for all tight relationships between vertices and prints the pairs of vertices that are tight.

```
function tight(arr) {
        var n = arr.length;
        var m = 0;
        var t = "";
        for (var i=0; i<n; i++) {
                for (var j=0; j<n; j++) {
                        var d = Math.abs(arr[j] - arr[i]);
                        if(d+j-i-n == 0) t = t + '('+arr[i]+','+arr[j]+')='
                        + (d + j - i - n) + '\n';
                                m++;
                }
        }
        return t;
}
```