Finding Colourful Trails

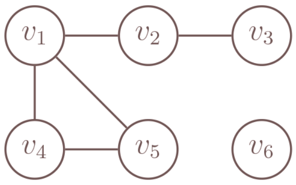
Jamie Lim Jia Sin

Introduction

We investigate a problem related to route inspection, named "Finding Colourful Trails".

1.1 Definitions

Consider a diagram made up from points and lines that connect some pairs of those points. An example is shown, where points are labelled v_i for $1 \leq i \leq 6$.

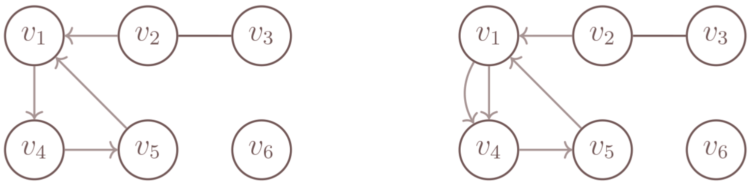


A graph

Such a diagram is called a *graph*. The points are known as the *vertices* of the graph and the lines are known as the *edges* of the graph.

A *walk* is an alternating sequence of vertices and edges in a graph, such that the sequence starts with a vertex and ends with a vertex. The *length* of a walk is the number of edges in the walk. A *trail* is a walk in which no edge is traversed more than once. A *path* is a trail in which no vertex is visited more than once.

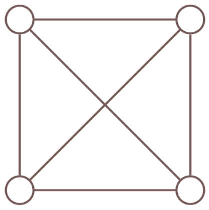
For example, consider the graph shown above. $v_2v_1v_4$ is a trail of length 2, $v_2v_1v_4v_5$ is a trail of length 3, $v_2v_1v_4v_5v_1$ is a trail of length 4 and $v_2v_1v_4v_5v_1v_4$ is a walk of length 5 (it is not a trail, because edge v_1v_4 is repeated).



$v_2v_1v_4v_5v_1$ is a trail of length 4 (left), $v_2v_1v_4v_5v_1v_4$ is a walk of length 5 (right)

Note: the graph considered in this example only has 1 edge between v_1 and v_4. There is 1 blue arrow for each time that this edge was traversed in the walk.

There are various classes of graphs, for example bipartite graphs and regular graphs. In this report, the focus is on complete graphs. Complete graphs are graphs in which every pair of distinct vertices is connected by a unique (and undirected) edge. A complete graph with n vertices is commonly denoted by K_n.



A complete graph with 4 vertices, K_4

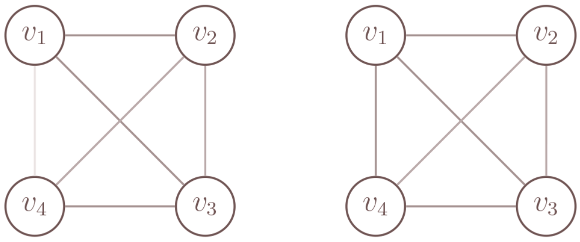
We say that two edges are adjacent if they share a common vertex. For example, in the graph above, v_1v_2 is adjacent to v_2v_3, v_1v_4 and v_1v_5.

An *edge colouring* of a graph is an assignment of "colours" to its edges such that each edge receives one colour and no two adjacent edges have the same colour.

Let C be the set of colours that are allowed to be used in a graph. We define a *valid colouring* of this graph to be an assignment of colours to its edges such that each edge has exactly one colour and each colour from C is used at least once. Note that two adjacent edges may have the same colour in a *valid colouring* of a graph.

For all integers $1 \leq i \leq |C|$, we define an *i-colourful* trail in a graph G to be a trail of length i in G such that no two edge colours in the trail are the same.

For example, the set of colours in the diagrams below is $C = \{1, 2, 3\}$. In the diagram on the left, we have v_1v_2, v_1v_3, v_3v_4 as colour 1 (blue), v_2v_4, v_2v_3 as colour 2 (red) and v_1v_4 as colour 3 (green). In the diagram on the right, colour 3 (green) is not used so the colouring is invalid.



Examples of K_4 with valid (left) and invalid (right) colouring for $|C| = 3$.
An example of a 3-colourful trail in the graph on the left is $v_1v_4v_2v_1$.

1.2 Problem Statement

Our research focuses on colourful trails in complete graphs. Let $C = \{1, 2, ...r\}$ be the set of colours for a graph of n vertices, where vertices are labelled $v_1, v_2, ..v_n$. We wish to determine the minimum and maximum possible n (i.e. the range of values n could take) such that there **always** exists an r-colourful trail **for all valid colourings of** K_n. (There must be at least 1 valid colouring of K_n.)

1.3 Inspiration and Rationale

The inspiration for this research topic comes from the route inspection problem, which is to find a shortest closed path or circuit that visits every edge of an undirected graph.

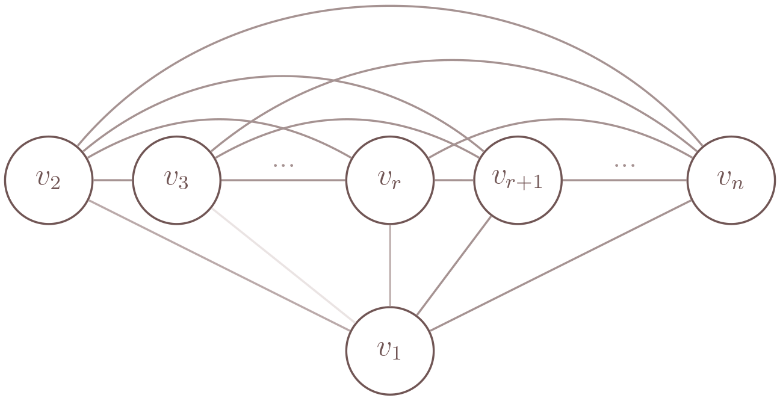
Our research aims to study graphs that have their edges coloured, and determine the size of the graph needed to guarantee that there exists an r-colourful trail. This may possibly be useful in areas where graph modelling with edge colouring can be seen, for example in round-robin tournaments, scheduling or fibre-optic communication.

Complete Graphs

2.1 Preliminary Observations

2.1.1 Observation on Possible Upper Bound on n

By considering the following construction, we can bound $n \leq r-1$, for $r > 5$. However, the tightness of the bound, especially for larger r, has not been proven. For $n \geq r$, we can construct as follows:



A visualisation of a construction to bound $n \leq r-1$

There is a connected graph of blue edges among all vertices excluding v_1. In addition, for $r+1 \leq i \leq n$, $v_1 v_i$ is blue.

Let the vertices be $v_1, v_2, ...v_n$. Let the edge from v_1 to v_i be colour i, for $2 \leq i \leq r$. For all of the remaining edges, let them be colour 1.

Suppose there exists an r-colourful trail in this graph. In order to have an r-colourful trail, we must traverse each of the edges coloured $2, 3, ...r$ (that is, edges $v_1 v_i$ for $2 \leq i \leq r$) exactly once. Hence we visit vertex v_1 at least $\lceil \frac{r-1}{2} \rceil$ times.

However, we claim that we can only visit vertex v_1 at most 2 times.

Proof. On any trail from v_1 to itself (that passes through other at least 1 other vertex), we must repeat a colour or traverse at least one edge of colour 1. This is because the trail either contains 2 vertices (v_1 to v_i to v_1, meaning the colour of $v_1 v_i$ is repeated), or we must traverse from some v_i to v_j $(i, j > 1)$ without passing through v_1, hence traversing at least one edge of colour 1.

Thus, to avoid repeating colours, if we visit v_1 k times, we must use an edge of colour 1 at least

$k-1$ times. If $k > 2$, then we use an edge of colour 1 more than 1 time, so an r-colourful trail is no longer possible. $\square$

Since $\lceil \frac{r-1}{2} \rceil > 2$ for $r > 5$, we cannot have $n \geq r$ for $r > 5$.

2.1.2 Examples for Small r

r	Minimum n	Maximum n
1	2	∞
2	3	∞
3	3	∞
4	4	5
5	4	6
6	5	5

$r = 1$

A 1-colourful trail is just an edge. Thus for an edge to exist in the complete graph K_n, the minimum n is 2 and the maximum n is ∞.

$r = 2$

A 2-colourful trail requires 2 edges that are connected at the same vertex with different colours. We claim that a 2-colourful trail exists as long as it is possible to have at least 2 colours in the graph (i.e. $n \geq 3$). This would imply the minimum n is 3 and the maximum n is ∞.

Proof. Suppose otherwise, and without loss of generality consider edge e connecting vertices v_1 and v_2. We assume e is colour 1. All edges connected to v_1 must be colour 1, otherwise if v_1v_i is colour 2, then there exists a 2-colourful trail, $v_iv_1v_2$. Therefore each vertex has an edge connected to it that is colour 1, v_1v_i. Since the entire graph must contain at least 1 colour 2 edge, let v_iv_j have colour 2. Then $v_1v_iv_j$ is a 2-colourful trail, a contradiction. $\square$

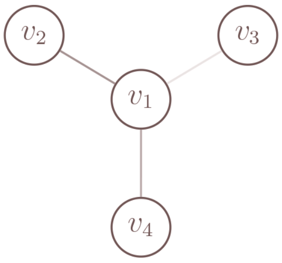
$r = 3$

Clearly the minimum n is 3, because the only way to colour K_3 using 3 colours is to colour each edge a different colour. We claim that the maximum n is infinity.

Proof. Suppose otherwise, that is for $r = 3$, there exists a construction for some sufficiently large n such that a 3-colourful trail does not exist.

Lemma 2.1.1. *In this construction, among edges connected to a vertex, there are at most 2 colours.*

Proof of Lemma 2.1.1. Otherwise, WLOG assume v_1 is connected to v_i by an edge of colour $i-1$, where $2 \leq i \leq 4$. Then consider edge v_2v_3. If v_2v_3 is colour 1, then $v_2v_3v_1v_4$ is a 3-colourful trail. If v_2v_3 is colour 2, then $v_3v_2v_1v_4$ is a 3-colourful trail. If v_2v_3 is colour 3, then $v_1v_2v_3v_1$ is a 3-colourful trail. This is a contradiction because v_2v_3 must have a colour i where $1 \leq i \leq 3$. $\square$



By the lemma, such a setup cannot exist

Now notice that in a valid colouring, there must exist at least 1 vertex such that there are 2 edges with different colours connected to it, otherwise the whole graph is the same colour. So, WLOG let v_1 be one such vertex, v_1v_2 is colour 1, v_1v_3 is colour 2. By the lemma, v_1v_i cannot be colour 3 for $4 \leq i \leq n$. Also, v_2v_i cannot be colour 3 for $4 \leq i \leq n$, otherwise $v_iv_2v_1v_3$ is a 3-colourful trail. Similarly v_3v_i cannot be colour 3. Since there must exist at least 1 colour 3 edge in the graph, let v_iv_j be colour 3, where $4 \leq i < j \leq n$.

But now edge v_1v_i cannot be any of the 3 colours. If v_1v_i is colour 1, $v_3v_1v_iv_j$ is a 3-colourful trail, and similarly there is a contradiction if v_1v_i is colour 2. As previously mentioned, v_1v_i cannot be colour 3 by Lemma 2.1.1.

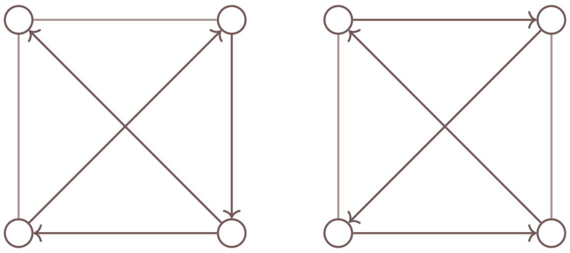
This is a contradiction, hence no construction exists. Maximum n for $r = 3$ is thus ∞. □

$r = 4$

First we show that the minimum n is 4. Since there are 6 edges in K_4, let a and b be edges such that the remaining 4 edges are of 4 distinct colours. (a and b may or may not be the same colour; for simplicity in diagrams, they are blue while the other edges are black.) We consider 2 cases.

Case 1 (left diagram): a and b share a vertex. Start from the bottom right vertex and traverse to the bottom left vertex first.

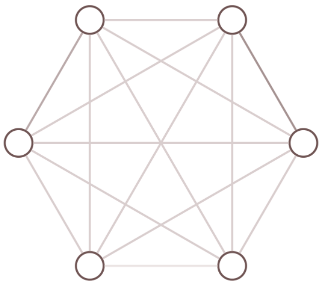
Case 2 (right diagram): a and b do not share a vertex. Start from any vertex in the diagram.



Constructions for $r = 4, n = 4$

Now we show that the maximum n is 5.

Notice that for $n \geq 6$, we can simply let v_1v_2 be colour 1, v_3v_4 be colour 2, v_5v_6 be colour 3 and the rest of the edges be colour 4. Then we clearly cannot traverse all the edges of colours 1 to 3 without traversing more than one colour 4 edge.

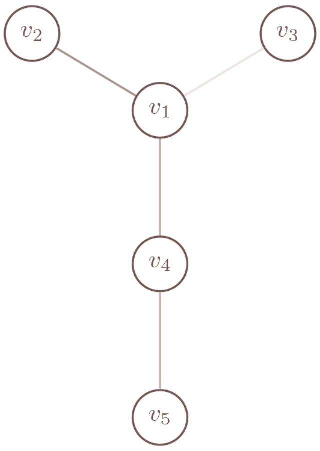


A setup which results in no r-colourful trail for $r = 6$, where $n = 6$. A similar idea can be applied for $n > 6$.

We are to show there exists an r-colourful trail for all valid colourings for $r = 4, n = 5$.

Proof. Suppose otherwise, that is there exists a construction for $n = 5$ for which there is no 4-colourful trail.

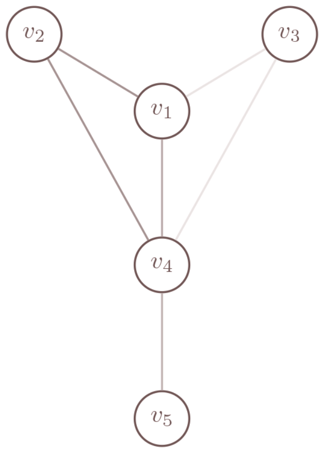
Lemma 2.1.2. *In such a construction with no 4-colourful trail, the following setup cannot exist. That is, it is not possible for $v_1v_2, v_1v_3, v_1v_4, v_4v_5$ to have pairwise distinct colours.*



We claim that such a setup cannot exist.

Proof of Lemma 2.1.2. Refer to the above diagram.

First we consider v_2v_4. If v_2v_4 is green, then $v_5v_4v_1v_2v_4$ is a 4-colourful trail. If v_2v_4 is red, then $v_5v_4v_2v_1v_3$ is a 4-colourful trail. If v_2v_4 is purple, then $v_3v_1v_2v_4v_1$ is a 4-colourful trail. So v_2v_4 must be blue, and similarly v_3v_4 must be green.



We realise v_2v_4 must be blue and v_3v_4 must be green to avoid 4-colourful trails.

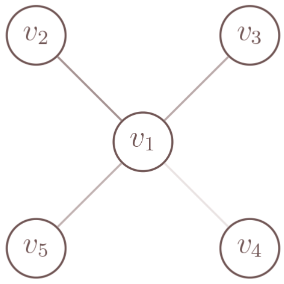
Next we consider v_2v_5. If v_2v_5 is green, then $v_1v_4v_5v_2v_4$ is a 4-colourful trail. If v_2v_5 is red, then $v_3v_4v_5v_2v_4$ is a 4-colourful trail. If v_2v_5 is blue, then $v_2v_5v_4v_1v_3$ is a 4-colourful trail. If v_2v_5 is purple, then $v_5v_2v_4v_1v_3$ is a 4-colourful trail. This is a contradiction. □

WLOG, v_1v_2 is colour 1 and v_1v_3 is colour 2. At least one of v_1, v_2, v_3 must have a colour 3 or 4 edge connected to it, otherwise since colour 3 and colour 4 must exist in the graph, v_4v_5 will be both colour 3 and colour 4 which is impossible.

Case 1: v_1 is connected to v_4 or v_5 by an edge that is neither colour 1 nor 2. WLOG, v_1v_4 is colour 3.

If any of v_2v_3, v_2v_4, v_3v_4 is colour 4, then it is clear by the construction in Figure 3, by considering v_1, v_2, v_3, v_4. Otherwise, one of $v_5v_1, v_5v_2, v_5v_3, v_5v_4$ is colour 4.

Case 1(a): v_5v_1 is colour 4. Then we have the following setup:



Case 1(a)

v_2v_3 cannot be colour 3, otherwise $v_1v_2v_3v_1v_5$ is a 4-colourful trail. Similarly v_2v_3 cannot be colour 4. WLOG let v_2v_3 be colour 1. Similarly let v_4v_5 be colour 3. Then $v_2v_3v_1v_5v_4$ is a 4-colourful trail. Thus there is no way for there to be no 4-colourful trail, contradiction.

Case 1(b): WLOG v_5v_2 is colour 4. Apply Lemma 2.1.2.

Case 2: Otherwise; that is, at least one of v_2, v_3 must have an edge that is neither colour 1 nor 2. WLOG, v_2 has a colour 3 edge.

Case 2(a): WLOG, v_2v_3 is colour 3. If any of v_1, v_2, v_3 has a colour 4 edge connected to it, WLOG v_1v_4 is colour 4. $v_1v_2v_3v_1v_4$ is a 4-colourful trail.

Otherwise, v_4v_5 is colour 4. Consider v_1v_4.

v_1v_4 cannot be colour 1, otherwise $v_5v_4v_1v_3v_2$ must be a 4-colourful trail. Similarly, v_1v_4 cannot be colour 2.

v_1v_4 cannot be colour 3, otherwise there is a setup as shown in Lemma 2.1.2.

v_1v_4 cannot be colour 4, otherwise $v_4v_1v_2v_3v_1$ is a 4-colourful trail.

This is a contradiction because v_1v_4 cannot have any colour.

Case 2(b): WLOG, v_2v_4 is colour 3.

If any of v_1v_4, v_2v_3, v_3v_4 is colour 4, then it is clear by the construction in Figure 3, by considering v_1, v_2, v_3, v_4.

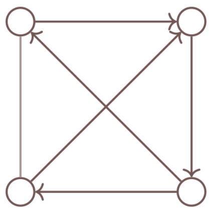
If v_3v_5 is colour 4, then $v_4v_2v_1v_3v_5$ is a 4-colourful trail, so v_3 cannot have a colour 4 edge. Similarly, v_4 cannot have a colour 4 edge.

So either v_1 or v_2 has a colour 4 edge to v_5. But this is not possible by Lemma 2.1.2.

Therefore, there exists no construction for $r = 4, n = 5$ such that there is no 4-colourful trail. □

$r = 5$

First we show that the minimum n is 4. Since there are 6 edges in K_4, one of the colours is appears exactly 2 times while the rest of the colours appear exactly 1 time. Suppose we remove one of the edges in the graph that has repeated colour (in the diagram, it is the blue edge). Then we obtain a Semi-Eulerian graph, and an Eulerian trail exists in this modified graph. Hence, there is a 5-colourful graph in K_4. Refer to the diagram. (The 5 black edges have distinct colours, and the repeated edge is not in the trail.)



Construction for $r = 5, n = 4$ (start from bottom right vertex and traverse to bottom left vertex first)

Now we show that the maximum n is 6. For $n \geq 7$, we can let v_1v_2 be colour 1, v_2v_3 be colour 2, v_4v_5 be colour 3, v_6v_7 be colour 4 and the rest be colour 5. Then we clearly cannot traverse all the edges of colours 1 to 4 without traversing more than one colour 5 edge.

We are to show that for all valid colourings for $r = 5, n = 6$, there exists an r-colourful trail. We use the following C++ code to find all valid colourings (WLOG, v_1v_2 is colour 1 and v_1v_3 is colour 2), then check whether there exists an r-colourful trail in each of them using Dynamic Programming.

Dynamic Programming is a method for solving a complicated problem by recursively breaking it down into simpler subproblems, solving each of those subproblems just once, and then storing their solutions in a data structure. This ensures that the answer for each subproblem is computed at most once, reducing the time taken to solve the problem.

The first function, `checkTrail`, is passed two variables `x` and `bitmask`. `x` is the current vertex number (0-indexed). `bitmask` is an integer that acts as a boolean array (i.e. an array of '0's and '1's), which stores for each colour whether it has been traversed yet (1) or not (0). Suppose `bitmask` has p colours that have not been traversed yet. `checkTrail` checks whether there exists a p-colourful trail that starts at x and traverses the remaining p colours that have not been traversed. It does so by considering all possible vertices that can be reached from x. Suppose some vertex y can be reached from x, and the colour of the edge xy has not been traversed before. Then we recursively call `checkTrail`, but with different input variables; the current vertex number becomes y and the bitmask is modified to indicate that the colour of edge xy has been traversed.

The second function, `checkValid`, checks whether a colouring given by the `construct` function (see below) is valid, by counting the number of occurrences of each colour in the colouring that has been constructed. For every vertex, `checkValid` then calls `checkTrail` starting at this vertex (since the trail could start at any vertex), where all colours have not been traversed before. If any of these calls to `checkTrail` indicate that a r-colourful trail exists, `checkValid` terminates. Otherwise, there exists a valid colouring in which an r-colourful trail does not exist, and a matrix representing the colours of the edges will be printed.

The third function, `construct`, assigns colours to the edges in the graph and stores the colours in a matrix. For every possible assignment of colours generated (regardless of whether it is valid or not), `construct` calls `checkValid` when all edges have been assigned a colour.

`construct` is called in the `main` function to assign colours to all the edges except v_1v_2 and v_1v_3, which have already been assigned colours WLOG.

```
#include <bits/stdc++.h>
using namespace std;
int r, n;
int col[10][10]; //array to store edge colours

int memo[10][1000]; //stores results of previous calculations to prevent recalculation and speed up the program
bool checkTrail(int x, int bitmask){
    if(memo[x][bitmask] != -1) return memo[x][bitmask];
    if(bitmask == (1 << r) - 1) return 1;
    bool ans = 0;
    for(int i = 0; i < n; i++){
        if(i == x) continue;
        if((1 << (col[x][i] - 1)) & bitmask)continue;
        ans = (ans || checkTrail(i, bitmask + (1 << (col[x][i] - 1))));
    }
    if(ans) return memo[x][bitmask] = 1;
    return memo[x][bitmask] = 0;
}

int countColours[10];
void checkValid(){
    memset(countColours, 0, sizeof(countColours));
    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++)countColours[col[i][j]]++;
```

```
    }
    for(int i = 1; i <= r; i++){
        if(countColours[i] == 0) return;
    }
    for(int i = 0; i < n; i++){
        if(checkTrail(i, 0))return;
    }
    //if the function has not terminated by now, it means that there is no possible vertex to start from that produces an r-colourful trail.
    printf("The following valid colouring does not produce an r-colourful trail:\n");
    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++){
            printf("%d ", col[i][j]);
        }
        printf("\n");
    }
}

void construct(int x, int y){
    if(x == n){
        checkValid();
        return;
    }
    if(y >= n){
        construct(x + 1, x + 2);
        return;
    }
    for(int i = 1; i <= r; i++){
        col[x][y] = i;
        col[y][x] = i;
        construct(x, y + 1);
    }
}

int main() {
    r = 5; n = 6; //initialisation of variables
    memset(memo, -1, sizeof(memo));
    //note: vertices are 0-indexed in this program
    col[0][1] = 1; col[1][0] = 1; //without loss of generality, this edge is colour 1
    col[0][2] = 2; col[2][0] = 2; //we set the edge between 0 and 2 to be a different colour from the edge between 0 and 1. (this is because somewhere in the graph, there exists 2 edges of different colours connected to the same vertex. so WLOG it is here.)
    construct(0, 3);
}
```

Since the program terminated without output or errors after approximately 130 seconds, we conclude that an r-colourful trail exists in all valid colourings for $r = 5, n = 6$.

$r = 6$

The minimum n cannot be < 4, otherwise K_n has < 6 edges.

If $n = 4$, there is only one possible colouring for $r = 6$, where each edge is a different colour. However, there is no Eulerian trail in K_4 because 4 is even, so there is also no 6-colourful trail in K_4. Hence minimum $n \geq 5$.

Now consider the observation in section 2.1.1. From this observation, we see that $n \leq 5$ for $r = 6$.

We claim that for all valid colourings for $r = 6, n = 5$, there exists an r-colourful trail. We can use code that is almost exactly the same as the code shown above for $r = 5, n = 6$, except a minor change in line 62 of the code above (initialisation of variables):

```
r = 6; n = 5; //initialisation of variables
```

The modified code terminates after approximately 0.15 seconds without output or errors. Hence, an r-colourful trail exists in all valid colourings for $r = 6, n = 5$. So minimum and maximum n is 5 for $r = 6$.

2.2 Partial Solutions (Loose Bounds)

For $r \leq 6$, the minimum and maximum n have already been proven above. For this section, we assume $r > 6$. Note that the tightness of the bounds in this section have not been proven.

2.2.1 Minimum n

Let M be the integer such that $\binom{M-1}{2} < r \leq \binom{M}{2}$.

If $r = \binom{M}{2}$ and M is even

Minimum $n \geq M + 1$ in this case.
If $n < M$, then $r = \binom{M}{2} > \binom{n}{2}$, so there is no valid colouring with r colours.
Since M is even, we see that there is no Eulerian trail in K_M. $r = \binom{M}{2}$ is the number of edges in K_M, so it is impossible to find an r-colourful trail (which is an Eulerian trail) for $n = M$.
Therefore, the minimum $n \geq M + 1$.

Otherwise

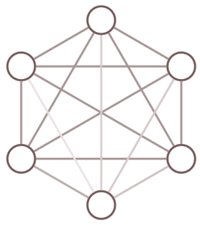
If $n < M$, then $r > \binom{M-1}{2} \geq \binom{n}{2}$, so there is no valid colouring with r colours. Therefore, the minimum $n \geq M$.
The tightness of this bound can be easily seen when $r = \binom{M}{2}$ and M is odd. This would mean that K_M is a complete graph with odd number of vertices, and the only possible colouring is when each edge is coloured with a different colour. Hence, in this graph, there exists an Eulerian trail which is also an r-colourful trail.

2.2.2 Maximum n

Suboptimal Bound

A suboptimal bound with similar ideas to the better bound was found. Its proof makes use of previous observations to lead to the better bound, as the thought process may not be obvious.

Suppose $r = 6, n \geq 6$ and the following setup is an induced subgraph of a K_n with valid colouring. The edges not drawn in the diagram from these 6 vertices are all colour 1 (blue in the diagram). Then it is impossible to find a 6-colourful trail, by using a similar proof as in section 2.1.



A setup which results in no r-colourful trail for $r = 6$, where $n \geq 6$

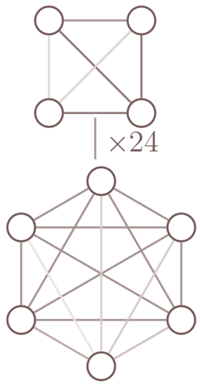
Let L be the integer such that $\binom{L-1}{2} < r-6 \leq \binom{L}{2}$. *(Note the different definition of L compared to in section 2.2.2.)*

We claim the maximum $n \leq L + 5$. We show that for all $n \geq L + 6$, there exists a construction such that there is no r-colourful trail.

Proof. Suppose $n \geq L + 6$. Consider $v_1, v_2, ...v_6$. Let $v_1 v_i$ be colour i for $2 \leq i \leq 6$.

Next we consider $v_7, ...v_n$. Notice that since $n \geq L+6$, there are at least L vertices from v_7 to v_n. Since $r - 6 \leq \binom{L}{2}$, there are enough edges among these vertices to use colours 7 to r, such that there is exactly 1 edge of each of these colours.

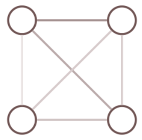
All the remaining edges are colour 1. Since no trail can traverse all of colours 2 to 6 without traversing a colour 1 edge more than once, there is no r-colourful trail. □



An example of the above construction for $r = 10$

Better Bound

For the better bound, we use a similar idea as the one mentioned in the previous section. (That is, the graph is considered in two "parts" with only 1 colour used for all the edges that cross between the two "parts".) Consider this setup:



$r = 6, n = 4$ results in no r-colourful trail

Now suppose $r \geq 7$. If $r > 7$, let L be the integer such that $\binom{L-1}{2} < r - 7 \leq \binom{L}{2}$. If $r = 7$, set $L = 1$. *(Note the different definition of L compared to in Appendix B.)*

We claim the maximum $n \leq L + 3$. We show that for all $n \geq L + 4$, there exists a construction such that there is no r-colourful trail.

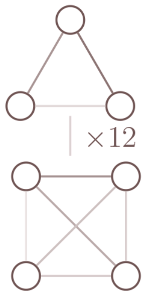
Proof. Consider v_1, v_2, v_3, v_4. Colour the edges $v_i v_j$ $(1 \leq i < j \leq 4)$ each with a different colour from colour 1 to colour 6, as shown in the diagram above.

Next we consider $v_5, ...v_n$. Notice that since $n \geq L + 4$, there are at least L vertices from v_5 to v_n. Since $r - 7 \leq \binom{L}{2}$, there are enough edges among these vertices to use colours 7 to $r - 1$, such that there is exactly 1 edge of each of these colours.

Colour r has not been used in the above construction. We use colour r for all the remaining edges.

Suppose there exists an r-colourful trail in this construction. Let G be the induced subgraph given by vertices v_1, v_2, v_3, v_4. Since there is no Eulerian trail in G, and there are distinct colours 1 to 6 in G that do not appear anywhere else in K_n, all the vertices in G cannot be visited consecutively. That is, for some $1 \leq i_1, i_2 \leq 4$ and $5 \leq j_1, j_2 \leq n$, there exists some edge $v_{i_1} v_{j_1}$ being traversed before some other edge $v_{j_2} v_{i_2}$.

However, these two edges are both colour r, which is a contradiction. Hence there does not exist an r-colourful trail for $n \geq L + 4$. □



An example of the above construction for $r = 10$

Other Graphs

Though the focus of the project is complete graphs, as a by-product, we have extended the study of r-colourful trails to some other classes of graphs. The results are detailed below.

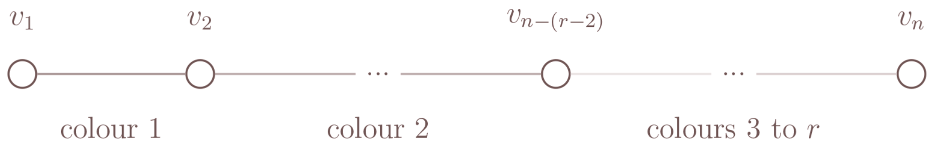
Note that in this section, we assume $r \geq 3$. This is because the bounds for $r = 1$ ($n \geq 2$) and $r = 2$ ($n \geq 3$) have the same proof as in section 2.1.2.

3.1 Paths

We claim that $n = r + 1$ if the graph is a path.

Clearly, $n > r$ because a path of n vertices has exactly $n - 1$ edges.

If $n > r + 1$, then the number of edges is strictly greater than the number of colours, so there exists a repeated colour. It is easy to see that the following construction results in no possible r-colourful trail:



If $n = r + 1$, then there are exactly r edges so each edge is a different colour, and we just traverse from one endpoint of the path to the other.

3.2 Cycles

A cycle is a nonempty trail in which the only repeated vertex is the first vertex, which is also the last vertex.

We claim that $r \leq n \leq r + 1$ if the graph is a cycle.

Clearly, $n \geq r$ because a cycle of n vertices has exactly n edges.

If $n = r$, each edge is a different colour so we just traverse around the cycle.

Notice that given a cycle, we can ignore 1 edge to make a path.

If $n = r + 1$, there is a repeated colour. So we ignore one of the edges that has that repeated colour to obtain a path with the same number of vertices and colours, where $n = r + 1$. Hence there always exists an r-colourful trail for a cycle with $n = r + 1$ vertices.

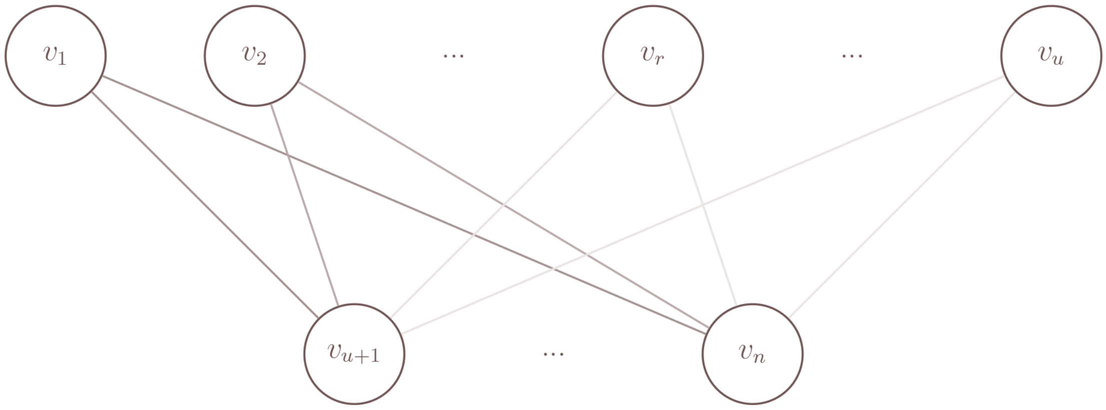
If $n > r + 1$, there is more than 1 edge whose colour appears more than once. We ignore any of these edges to obtain a path with the same number of vertices and colours, i.e. $n > r + 1$. Hence there exists a construction for cycles with $n > r + 1$ vertices, where there is no r-colourful trail.

3.3 Bipartite Graphs

A bipartite graph is a graph whose vertices can be divided into two disjoint, independent sets such that every edge connects a vertex in one set to a vertex in the other set. We denote the size of the larger set by u.

Here we wish to determine the minimum and maximum possible n such that there always exists an r-colourful trail for all possible valid colourings of all possible bipartite graphs with n vertices.

We see that if either set of vertices contains $\geq r$ vertices and the other set is nonempty, then there is a construction for an r-colourful trail. For $1 \leq i \leq r, u+1 \leq j \leq n$, $v_i v_j$ is colour i, and for $r+1 \leq i \leq u, u+1 \leq j \leq n$, $v_i v_j$ is colour r. We cannot traverse through more than 2 colours without repeating a colour.



Hence, the maximum n will be $\leq r+1$.

Conclusion

4.1 Summary of Results

Overall, we have realised that the ideas used to prove some results were useful in finding tighter bounds or proving other similar results. We hope to work on proving the tightness of the bounds provided.

4.1.1 Complete Graphs

Tight bounds have been found for $r \leq 6$:

r	Minimum n	Maximum n
1	2	∞
2	3	∞
3	3	∞
4	4	5
5	4	6
6	5	5

It is worth noting that the minimum and maximum n are not necessarily monotonically increasing.

We notice that if $r = 7$, then $n \leq 4$ and $n \geq 5$, so there is no possible n.
For $r \geq 8$, some bounds have been found. The tightness of these bounds have not been proven in general.
If $r = \binom{M}{2}$ for some even M, then the minimum $n \geq M + 1$.
Otherwise, the minimum $n \geq M$, and this bound is tight when $r = \binom{M}{2}$ and M is odd.
The maximum $n \leq L + 3$ where $\binom{L-1}{2} < r - 7 \leq \binom{L}{2}$.

4.1.2 Other Graphs

Considering $r \geq 3$, we have:

Class of Graph	Bounds
Paths	$n = r + 1$
Cycles	$r \leq n \leq r + 1$
Bipartite	Maximum $n \leq r + 1$

4.2 Applications

Applications of this problem would involve complete graphs where edges are coloured.

A possible application of this problem could be during a round-robin tournament. Suppose teams are represented by vertices, games are represented by edges and the different rounds are represented by different edge colours. A reporter may start reporting about one team, report about the team's game, then move to follow the other team to their game in the next round. If the reporter wants to report about 1 game in each round, our research could help to bound the number of teams in the tournament.

Another application, especially for more general graphs, is in a situation where a person wants to review some r airlines operating among n airports, using the least number of flights (i.e. r flights). Our research bounds the number of airports that the person will have to stop at.

4.3 Further Investigation and Extension

In the future, we hope to investigate the tightness of the bounds provided above. We would also like to consider the following questions:

1. Does there exist an r-colourful trail for all K_n for all positive integers n between the minimum and the maximum found (inclusive)? (That is, is the range continuous?)
2. What if the trail must have its colours in a specific order?
3. What if the graph is a general graph? Is it possible to bound the number of edges in the graph given n and r?
4. What if it must be a path instead of a trail, i.e. vertices cannot be repeated?

4.4 Acknowledgements

We would like to thank the following people:

- Mr Chia Vui Leong, teacher-mentor, for his guidance throughout the course of this project
- Mr Chai Ming Huang, teacher-in-charge, for his support that made this project possible
- Family members, for their support and valuable advice
- Friends, for their feedback and encouragement

Bibliography

[1] Bóna, M. *A Walk Through Combinatorics: An Introduction to Enumeration and Graph Theory.* World Scientific, 2017.

[2] Koh, K. M., Dong, F., and Tay, E. G. *Introduction to Graph Theory.* World Scientific, 2007.

[3] WikiBooks. Latex bibliography management. Avaliable at https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management (2015/08/26).

[4] Wikipedia. Complete graph. Available at https://en.wikipedia.org/wiki/Complete_graph (2020/01/01).

[5] Wikipedia. Dynamic programming. Available at https://en.wikipedia.org/wiki/Dynamic_programming (2020/01/01).

[6] Wikipedia. Edge colouring. Available at https://en.wikipedia.org/wiki/Edge_coloring (2020/01/01).

[7] Wikipedia. Eulerian path. Available at https://en.wikipedia.org/wiki/Eulerian_path (2020/01/01).

[8] Wikipedia. Route inspection problem. Available at https://en.wikipedia.org/wiki/Route_inspection_problem (2020/01/01).