Volume : 2 | Issue : 3 | March 2013 ISSN - 2250-1991

Research Paper Mathematics

The Gracefulness of Finite Number of Copies of C4 and Path Margin With Dotnet Framework 3.5

*A. Solairaju **N. Abdul Ali ***M. Manikandan

* P.G. & Research Department of Mathematics, Jamal Mohamed College, Trichy – 20.

** P.G. & Research Department of Mathematics, Jamal Mohamed College, Trichy – 20.

*** M.Phil Scholar, Jamal Mohamed College, Trichy – 20.

ABSTRACT

This paper obtain the merged to finite number of merging of finite copies of C_4 and path is graceful and also can be programming dotnet framework 3.5

Keywords : Graceful labeling, Graceful Graph, Generalized the Gracefulness of finite number of copies of C4 and path merging with dotnet Framework 3.5, n=number of path merging with C_4

Introduction:
While the graceful labeling of graphs is perceived to be a primarily Theoretical subject in the field of graph theory and discrete mathematics , gracefully labeled graphs often serve as models in a wide range of applications. Such applications include coding theory and communication network addressing. The study of graceful graphs and graceful labeling methods was introduced by Rosa [2] or one given Graham and Sloane [1]. Rosa defined a function f, a b-valuation of a graph with q edges if f is an injective map from the vertices of G to the set {0, 1, 2 ,…,q} such that when each edge xy is assigned the label ½f(x)-f(y)½, the resulting edge labels are distinct.

The graceful labeling problem is to determine which graphs are graceful. Proving

a graph G is or is not graceful involves either producing a graceful labeling of or showing that no such labeling exists. Over the past 30 years, approximately 200 papers on graceful labeling methods have been published. An unpublished result of Erdos¨ that was later proven by Graham and Sloane states that almost all graphs are not graceful [1], though it does appear that most graphs having some regularity of structure to them are graceful. Most of the papers published to date on the subject of graceful labeling are theoretical, however, and principally focus on certain classes of graphs and labeling methods. Such papers often present arguments by either providing formulas for gracefully labeling graphs within a particular class, or proofs that graphs of a particular class are not graceful. The gracefulness of several classes of graphs has already been established. For example : Gracefulness of nc4 Merging with paths [3], are graceful, Gracefulness of Tp-tree with five levels obtained by java programming [4] is obtained, A new class of graceful trees [5], are graceful, Gracefulness of P_K [6], are graceful, An algorithm for finding graceful labeling for P_K [7], is obtained, The graceful of the merging graph N $_{**}$ C_4 with dot net frame work [8], is obtained, The graceful of a finite number of copies of C_4 with dotnet frame work 3.5 [9], is obtained and [10,11,12] used for dotnet frame work 3.5.

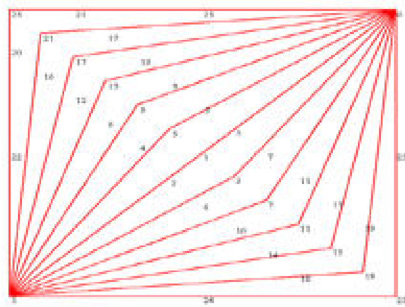
Section I: Preliminaries
Definition 1.1:
Let G = (V, E) be a simple graph with p vertices and q edges.

A map f: V (G) ® {0, 1, 2,…, q} is called a graceful labeling if

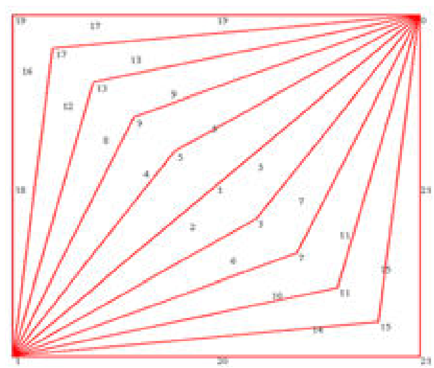
(i) f is one – to – one
(ii) The edges receive all the labels (numbers) from 1 to q where the label of an edge is the absolute value of the difference between the vertex labels at its ends.

A graph having a graceful labeling is called a graceful graph.

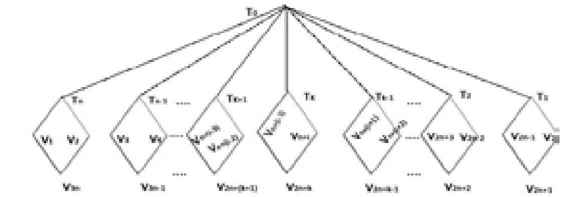
Example 1 : n is Even (n=6) :



Example 2 : n is Odd (n=5)



Section – II: Theorem:
The Gracefulness of C_4 merging with path Generalization:



n = Number of copies of path merging with C_4.

$T = \{T_0, T_1, T_2, \ldots, T_{k-1}, T_k, T_{k+1} \ldots, T_{n-1}, T_n\}$

$V = \{V_1, V_2, V_3, V_4, \ldots, V_{n+(i-3)}, V_{n+(i-2)}, V_{n+(i-1)}, V_{n+i}, V_{n+(i+1)}, V_{n+(i+2)}, \ldots, V_{2n-3}, V_{2n-2}, V_{2n-1}, V_{2n}, V_{2n+1}, V_{2n+2}, \ldots, V_{2n+k-1}, V_{2n+k}, V_{2n+k+1}, \ldots, V_{3n-1}, V_{3n}\}$

q = Number of edges = n×5

$f(T_0) = 0$,

$f(T_1) = q$,

$f(T_i) =$,

Where k=

$f(V_1)=q-2$,

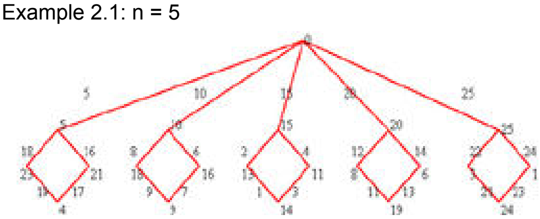
$f(V_2)=q-4$,

$f(V_i) =$

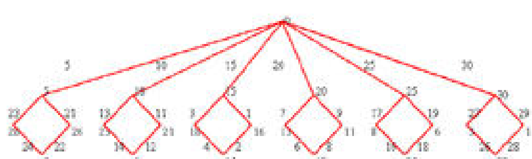
Where k=

$f(V_{2n+1})=q-1$.

Example 2.1: n = 5



Example 2. 2 : n = 6



Section – 3:
An algorithm for THE GRACEFULNESS OF FINITE NUMBER OF COPIES OF C_4 AND PATH MARGIN WITH DOTNET Framework 3.5

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace Finite_Logices
{
public partial class Form1 : Form
{
public Form1()
{
InitializeComponent();
}
private void btnDraw_Click(object sender, EventArgs e)
{
Graphics g;
g = this.CreateGraphics();
g.Clear(Color.White);
SolidBrush myBrush = new SolidBrush(Color.Black);
Font font = new Font("Times New Roman", 12.0f);
Pen myPen = new Pen(Color.Red);
myPen.Width = 2;
int n = Convert.ToInt32(txtNo.Text);
int arraysize = (3*n) + 1;
int[] V = new int[arraysize];
int[] T = new int[n + 1];
int q = n * 5;
T[0] = 0;
T[1] = q;
V[1] = q - 2;
V[2] = q - 4;
for (int j = 3; j <= (3 * n); j++)
{
if (j <= 2 * n)
{
V[j] = V[j - 2] - 5;
}
else if (j > (2 * n + 1))
{
V[j] = V[j - 1] - 5;
}
else
{
V[j] = q - 1;
}
}
int StartX = 600;
int StartY = 50;
int StartX1 = StartX+(n*50);
int diff = StartX1 / (n+1);
int StartY1 = StartY + 100;
g.DrawLine(myPen, StartX, StartY, StartX1, StartY1);
g.DrawLine(myPen, StartX1, StartY1, StartX1 + 40, StartY1+40); g.DrawLine(myPen, StartX1, StartY1, StartX1 - 40, StartY1 + 40);

DrawLine(myPen, StartX1 + 40, StartY1 + 40, StartX1, StartY1+80);

g.DrawLine(myPen, StartX1 - 40, StartY1 + 40, StartX1, StartY1+80);

g.DrawString(T[0].ToString(), font, myBrush, StartX, StartY-10);

g.DrawString(T[1].ToString(), font, myBrush, StartX1, StartY1-10);

g.DrawString(T[1].ToString(), font, myBrush, StartX1-50, StartY1 - 50);

int decrement1 = 2 * n;
int decrement2 = (2 * n - 1);

int Increment1 = (2 * n + 1);
g.DrawString(V[decrement1].ToString(), font, myBrush, StartX1 + 40, StartY1 + 40);

g.DrawString((T[1]-V[decrement1]).ToString(), font, myBrush, StartX1 +30, StartY1 + 15);

g.DrawString(V[decrement2].ToString(), font, myBrush, StartX1 - 40, StartY1 + 40);

g.DrawString((T[1] - V[decrement2]).ToString(), font, myBrush, StartX1 - 40, StartY1 + 15);
g.DrawString(V[Increment1].ToString(), font, myBrush,
```

```
StartX1, StartY1 + 80);

g.DrawString((V[Increment1] - V[decrement1]).ToString(), font, myBrush, StartX1+15, StartY1 + 60);

g.DrawString((V[Increment1] - V[decrement2]).ToString(), font, myBrush, StartX1-25, StartY1 + 60);

for (int i = 2; i <= n; i++)
{
StartX1 = StartX1 - diff;
T[i] = T[i - 1] - 5;

g.DrawLine(myPen, StartX, StartY, StartX1, StartY1);

g.DrawString(T[i].ToString(), font, myBrush, StartX1, StartY1 - 15);

if(i<=n/2)
{
g.DrawString(T[i].ToString(), font, myBrush, StartX1-60, StartY1 - 50);

}
else if (i == (n / 2) + 1)
{
g.DrawString(T[i].ToString(), font, myBrush, StartX1, StartY1 - 50);
}
else
{

g.DrawString(T[i].ToString(), font, myBrush, StartX1 + 30, StartY1 - 50);

}
g.DrawLine(myPen, StartX1, StartY1, StartX1 + 40, StartY1 + 40);
g.DrawLine(myPen, StartX1, StartY1, StartX1 - 40, StartY1 + 40);

decrement1=decrement1 - 2;
decrement2 = decrement2 - 2;

g.DrawString(V[decrement1].ToString(), font, myBrush, StartX1 + 40, StartY1 + 40);

g.DrawString(V[decrement2].ToString(), font, myBrush, StartX1 - 50, StartY1 + 40);

g.DrawString((Math.Abs(T[i] - V[decrement1])).ToString(), font, myBrush, StartX1 + 30, StartY1 + 15);

g.DrawString((Math.Abs(T[i] - V[decrement2])).ToString(), font, myBrush, StartX1 - 50, StartY1 + 15);

g.DrawLine(myPen, StartX1 + 40, StartY1 + 40, StartX1, StartY1 + 80);

g.DrawLine(myPen, StartX1 - 40, StartY1 + 40, StartX1, StartY1 + 80);

Increment1 = Increment1 +1;
g.DrawString(V[Increment1].ToString(), font, myBrush, StartX1, StartY1 + 80);

g.DrawString((Math.Abs(V[Increment1] - V[decrement1])).ToString(), font, myBrush, StartX1 + 15, StartY1 + 60);

g.DrawString((Math.Abs(V[Increment1] - V[decrement2])).ToString(), font, myBrush, StartX1 - 30, StartY1 + 60);

}
}
}
}
```

REFERENCES

R. L. Graham and N. J. A. Sloane, On additive bases and harmonious graph, SIAM J. Alg. Discrete Math., 1 (1980) 382 – 404. | A. Rosa, On certain valuation of the vertices of a graph, Theory of graphs (International | Synposium,Rome,July 1966),Gordon and Breach, N.Y.and Dunod Paris (1967), 349-355. | A. Solairaju, N.Abdul ali, Gracefulness of nc4 Merging with paths, International Organization of Scientific Research (IOSR),Volume 4,Issue 4, 20 December 2012.Paper ID:G22078 [ISSN 2278 – 5728] | Page : 10 - 12 | A. Solairaju, N.Abdul ali , s. Abdul saleem, Gracefulness of Tp-tree with five levels obtained by java programming, The International Journal of Scientific and Research Publication (IJSRP), | Volume 2,Issue 12,December 2012 Edition, [ISSN 2250 – 3153] Paper id: P12552 | A. Solairaju, N.Abdul ali, A new class of graceful trees. International journal of science & engineering research (IJSER),Volume 4,issue 1, January 2013.paper ID : I01653 [ISSN 2229 – 5518] | A. Solairaju, N.Abdul ali, Gracefulness of PK 2_(C_k), International Journal of Engineering Research and Technology(IJERT),Volume 1,Issue 10, December 2012,Paper ID :P12552 [ISSN 2278 – 0181] | A. Solairaju, N.Abdul ali , s. Abdul saleem, An algorithm for finding graceful labeling for PK 2_(C_k),The international journal of Engineering and Science [IJES] volume 1, issue 2, December 2012, Page 287 – 296 [ISSN 2319 – 1813] Paper id: 12151 | A. Solairaju, N.Abdul ali, and R.M.Karthikkeyn, The graceful of the merging graph N ** C4 with dot net frame work, International Journal of Computational Engineering Research (IJCER) volume 3, issue 1, 15th January 2013 [ISSN 2250 – 3005] Paper id: 31014 | A. Solairaju, N.Abdul ali, and A. Sumaiya banu, The graceful of a finite number of copies of C4 with dotnet frame work 3.5, International Journal of Engineering Research and Application(IJERA), volume 3, issue 1, January – February 2013, Page: 1258 – 1266 [ISSN 2248 – 9622] Paper id: 31397 | Tbuan thai & hoang Q.Lan,".Net Framework Essentials",O`reilly,2nd Edition 2007. | D.Nikhil Kothari,",.Net Framework",Addison-Wesley Professional,2008. | Christian Nagel et al,"Programming in C#",Wrox Publication,2001 | Contact Detail: | N.Abdul ali, | Associate professor of mathematics, | PG& Research department of mathematics, | Jamal Mohamed college, | Trichy – 620020, | Tamilnadu. | Cell: 9345116670. |