Solairaju, N. Abdul Ali, A. Sumaiya Banu / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 1, January -February 2013, pp.1258-1266

The Graceful Of A Finite Number Of Copies Of C_4 With Dotnet Framework 3.5

A. SOLAIRAJU[1] , N. ABDUL ALI[2] and A. SUMAIYA BANU[3]

[1-2]: P.G. & Research Department of Mathematics, Jamal Mohamed College, Trichy – 20.

[3]: M.Phil Scholar, Jamal Mohamed College, Trichy – 20.

Abstract:

The gracefulness of a finite number of copies for circuit with length 4 is obtained with Dotnet Framework 3.5.

Keyword: Graceful labeling, Graceful Graph, Generalized the Graceful of a Finite number of Copies of C_4

Introduction:

Most graph labeling methods trace their origin to one introduced by Rosa [2] or one given Graham and Sloane [1]. Rosa defined a function f, a β-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 $|f(x)-f(y)|$, the resulting edge labels are distinct.

A. Solairaju and K. Chitra [3] first introduced the concept of edge-odd graceful labeling of graphs, and edge-odd graceful graphs.

A. Solairaju and others [5,6,7,8,9] proved the results that(1) the Gracefulness of a spanning tree of the graph of Cartesian product of P_m and C_n, was obtained (2) the Gracefulness of a spanning tree of the graph of Cartesian product of S_m and S_n, was obtained (3) edge-odd Gracefulness of a spanning tree of Cartesian product of P_2 and C_n was obtained (4) Even -edge Gracefulness of the Graphs was obtained (5) ladder P_2 x P_n is even-edge graceful, and (6) the even-edge gracefulness of $P_{n\ O}$ nC_5 is obtained.(8) Gracefulness of Tp-tree with five levels obtained by java programming,(9) Gracefulness of ${}^{n}C_{4\ \text{Merging with paths}}$,(10) A new class of graceful trees and (11) Gracefulness of $P_K \circ 2_{C_k}$, is obtained. (12, 13, 14) used for dot net framework 3.5

Section I: Preliminaries

Definition 1.1:

Let G = (V, E) be a simple graph with p vertices and q edges.

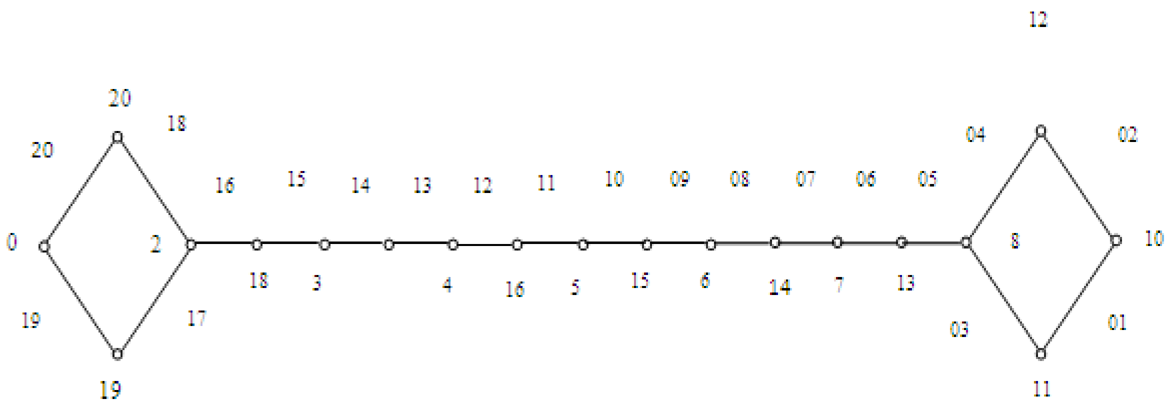
A map f: V (G) → {0, 1, 2,…, q} is called a graceful labeling if

(i) f is one – to – one

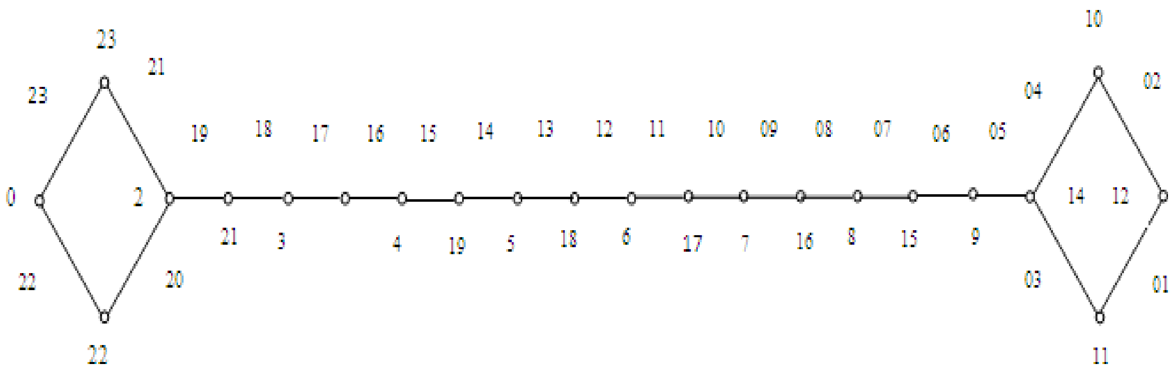
(ii) The edges receive all the labels (numbers) from 1 to q where the label of an edge is the absolute value of the difference between the vertex labels at its ends.

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

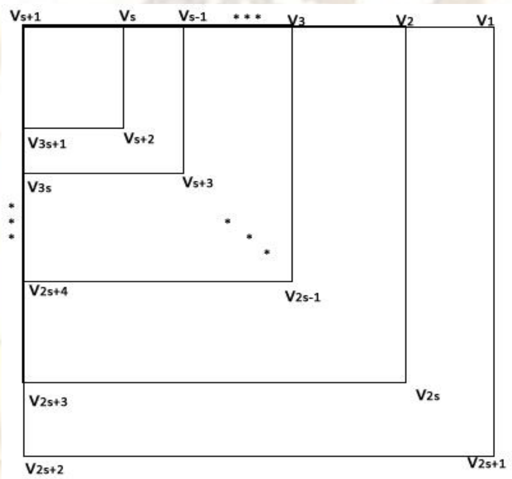
Example 1.1: k = 11 (odd) ; P: V |→ 19; Q: e |→ 20



Example 1.2: k =14 (even); P: V |→ 22; Q: e |→ 23



Section – 2:Theorem:
The Graceful of a Finite Number of Copies of C_4 Generalization



CASE 1: S is even:-

S = Number of square.
V = $\{V_1, V_2, \ldots, V_s, V_{s+1}, \ldots, V_{2s}, V_{2s+1}, \ldots, V_{3s}, V_{3s+1}\}$
q = Number of edges

$f(V_1)$ = Number of Edges = $4\times S = L$.
$f(V_2) = 4$.

$$f(V_n) = \begin{cases} f(Vn-2)-3, & for \quad n = 3,5,\ldots\ldots(S-1), \\ f(Vn-2)+5, & for \quad n = 4,6,\ldots..(S-2) \\ f(Vn-2)+4, & for \quad n = s \end{cases}$$

$f(V_{s+1}) = f(V_{s-1}) - 4$.

$$f(V_n) = \begin{cases} f(Vn-1)+1, & for \quad n = S+2, \\ f(Vn-2)-4, & for \quad n = S+3 \end{cases}$$

$f(V_{s+4}) = f(V_{s+2}) + 4$

$$f(V_n) = \begin{cases} f(Vn-2)-5, & for \quad n = (S+5),(S+7),\ldots\ldots(2S+1) \\ f(Vn-2)+3, & for \quad n = (S+6),(S+8),\ldots\ldots(2S) \\ f(Vn-2)-3, & for \quad n = (2s+4),\ldots.3S \\ f(Vn-2)+5, & for \quad n = (2s+5),\ldots.(3S+1) \end{cases}$$

$f(V_{2s+3}) = 1$; $f(V_{2s+2}) = L-1$

CASE 2: S is odd:-

S = Number of square.

$V = \{V_1, V_2, \ldots, V_s, V_{s+1}, \ldots, V_{2s}, V_{2s+1}, \ldots, V_{3s}, V_{3s+1}\}$

q = Number of edges

$f(V_1)$ = Number of Edges = $4 \times S = L$.

$f(V_2) = 4$.

$f(V_{2s+3}) = 1$; $f(V_{2s+2}) = L-1$

$$f(V_n) = \begin{cases} f(Vn-2)-3, & for \quad n = 3,5, \ldots \ldots S, \\ f(Vn-2)+5, & for \quad n = 4,6, \ldots \ldots (S-1) \end{cases}$$

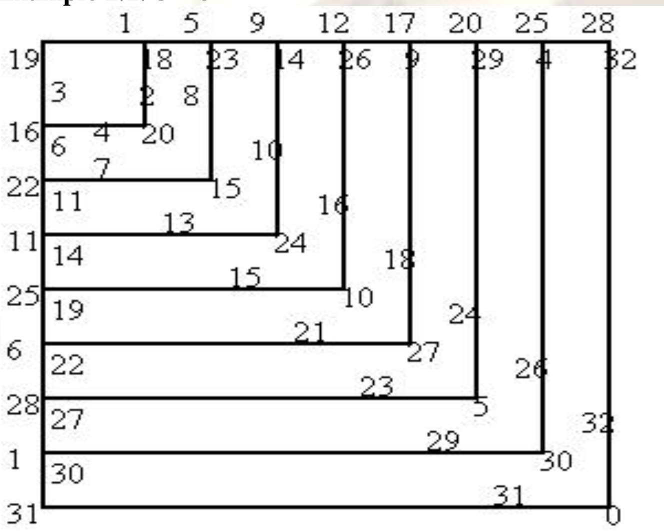
$f(V_{s+1}) = f(V_s) - 2$.

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

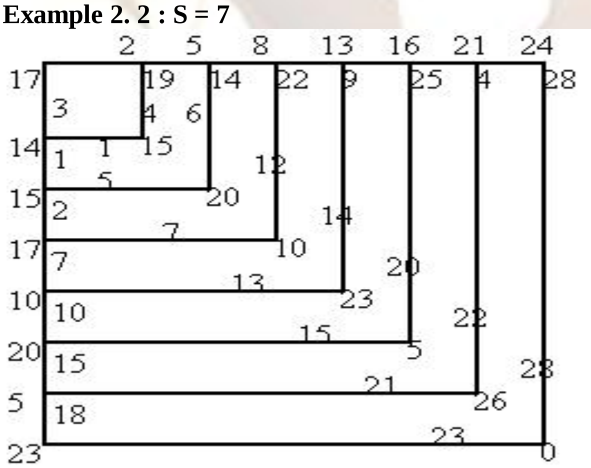
$f(V_{s+3}) = f(V_{s+1}) + 3$.

$$f(V_n) = \begin{cases} f(Vn-2)-5, & for \quad n = (S+4), (S+6), \ldots \ldots (2S+1) \\ f(Vn-2)+3, & for \quad n = (S+5), (S+7), \ldots \ldots (2S) \\ f(Vn-2)-3, & for \quad n = (2s+4), \ldots . (3S+1) \\ f(Vn-2)+5, & for \quad n = (2s+5), \ldots . (3S) \end{cases}$$

Example 2.1: S = 8



Example 2. 2 : S = 7



Section – 3:

An algorithm for THE GRACEFUL OF A FINITE NUMBER OF COPIES OF C_4 in Dotnet Framework 3.5

Case 1 : S is Even

```
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 Square
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }

      private void btnDraw_Click(object sender, EventArgs e)
      {
         //Initialize Graphics tools.....
         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;

         //Get V values for Even Numbers
         int s = Convert.ToInt32(txtSquare.Text);
         int arraysize = (s * 3) + 2;
         int[] V = new int[arraysize];

         if (s % 2 == 0)
         {
            V[1] = s * 4;
            V[2] = 4;
            for (int j = 3; j < arraysize; j++)
            {
               if (j < s)
               {
                  if (j % 2 == 0)
                  {
                     V[j] = V[j - 2] + 5;
                  }
                  else
                  {
                     V[j] = V[j - 2] - 3;
                  }
               }
               else if (j == s)
               {
                  V[j] = V[j - 2] + 4;
               }
               else if (j > s)
               {
                  if (j == s + 1)
                  {
```

```
            V[s + 1] = V[s - 1] - 4;
        }
        else if (j == s + 2)
        {
            V[s + 2] = V[s + 1] + 1;
        }
        else if (j == s + 3)
        {
            V[s + 3] = V[s + 1] - 4;
        }
        else if (j == s + 4)
        {
            V[s + 4] = V[s + 2] + 4;
        }
        else if (j > (s + 4))
        {
            if (j <= ((2 * s) + 1))
            {
                if (j % 2 == 0)
                {
                    V[j] = V[j - 2] + 3;
                }
                else
                {
                    V[j] = V[j - 2] - 5;
                }
            }
            else if (j == ((2 * s) + 2))
            {
                V[j] = (4 * s) - 1;
            }
            else if (j == ((2 * s) + 3))
            {
                V[j] = 1;
            }
            else
            {
                if (j % 2 == 0)
                {
                    V[j] = V[j - 2] - 3;
                }
                else
                {
                    V[j] = V[j - 2] + 5;
                }
            }
        }
    }
  }
}
```

Case 2 : S is Odd

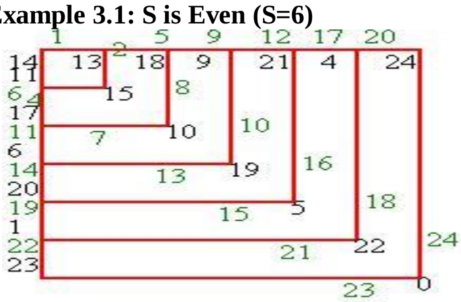
```
//Get V values for Odd Numbers
else
{
    V[1] = s * 4;
    V[2] = 4;
    V[(2 * s) + 3] = 1;
    V[(2 * s) + 2] = V[1] - 1;
```

```
    for (int j = 3; j < arraysize; j++)
    {
      if (j <= s)
      {
        if (j % 2 == 0)
        {
          V[j] = V[j - 2] + 5;
        }
        else
        {
          V[j] = V[j - 2] - 3;
        }
      }
      else if (j > s)
      {
        if (j == s + 1)
        {
          V[s + 1] = V[s] - 2;
        }
        else if (j == s + 2)
        {
          V[s + 2] = V[s + 1] - 2;
        }
        else if (j == s + 3)
        {
          V[s + 3] = V[s + 1] + 3;
        }
        else if (j >= (s + 4))
        {
          if (j <= ((2 * s) + 1))
          {
            if (j % 2 == 0)
            {
              V[j] = V[j - 2] + 3;
            }
            else
            {
              V[j] = V[j - 2] - 5;
            }
          }
          else
          {
            if (j % 2 == 0)
            {
              V[j] = V[j - 2] - 3;
            }
            else
            {
              V[j] = V[j - 2] + 5;
            }
          }
        }
      }
    }
  }

  ///Draw Square
  int StartX = 20;
  int StartY = 20;
```

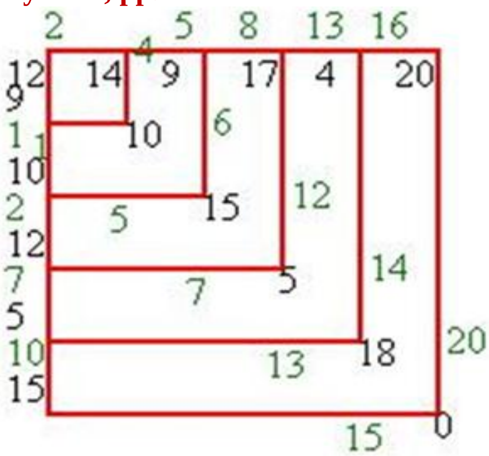
```
int Width = 600;
int Height = 600;
for (int square = 0; square < s; square++)
{
   g.DrawRectangle(myPen, StartX, StartY, Width, Height);
   Width = Width - 30;
   Height = Height - 30;
}
int increment = Width + 25;
int increment1 = Height + 25;
StartX = 20;
StartY = 20;
Width = 600;
Height = 600;
for (int p = 1; p < arraysize; p++)
{
   if (p <= (s + 1))
   {
      g.DrawString(V[p].ToString(), font, myBrush, Width, StartY);
      if (p == s)
      {
         Width = 0;
      }
      else
      {
         Width = Width - 30;
      }
   }
   else if (p > (s + 1) && p <= ((2 * s) + 1))
   {
      if (p == (s + 2))
      {
         StartX = StartX + increment;
         StartY = StartY + increment1;
      }
      else
      {
         StartX = StartX + 30;
         StartY = StartY + 30;
      }
      g.DrawString(V[p].ToString(), font, myBrush, StartX, StartY);
   }
   else
   {
      g.DrawString(V[p].ToString(), font, myBrush, 0, Height);
      Height = Height - 30;
   }


   // Find Difference
   int StartX1 = 600;
   int StartY1 = 0;
   SolidBrush myBrush1 = new SolidBrush(Color.Green);
   for (int diff =1; diff <(s+1); diff++)
   {
      if (diff == s)
      {
         StartX1 = StartX1 - 50;
      }
      else if (diff == 1)
```

```
{     StartX1 = StartX1 - 10;              }
else
{
   StartX1 = StartX1 - 25;
}
g.DrawString(Convert.ToString(Math.Abs(V[diff] - V[diff + 1])), font, myBrush1, StartX1,
StartY1);
}
StartX1 = 0;
StartY1 = 600;
for (int diff1 = ((2 * s) + 2); diff1 <= ((3 * s) + 1); diff1++)
{
   if (diff1 == ((2 * s) + 2))
   {
      StartY1 = StartY1 - 15;
   }
   else
   {
      StartY1 = StartY1 - 30;
   }

      if (diff1 == (3 * s + 1))
      {
         StartY1 = StartY1 - 50;
         g.DrawString(Convert.ToString(Math.Abs(V[diff1] - V[s + 1])), font, myBrush1, StartX1,
StartY1);
      }
      else
      {                    g.DrawString(Convert.ToString(Math.Abs(V[diff1] - V[diff1 + 1])), font,
myBrush1, StartX1, StartY1);
      }
}
StartX1 = 620;
StartY1 = 550;
int Xaxis = 1;
int Yaxis = 2*s + 1;
for (int diff = 1; diff < s + 1; diff++)
{
   g.DrawString(Convert.ToString(Math.Abs(V[Yaxis] - V[Xaxis])), font, myBrush1, StartX1,
StartY1);
   Xaxis = Xaxis + 1;
   Yaxis = Yaxis - 1;
   StartX1 = StartX1 - 30;
   StartY1 = StartY1 - 30;
}

StartX1 = 550;
StartY1 = 620;
 Xaxis = 2*s+1;
 Yaxis =  2*s+2;
for (int diff = 2 * s + 1; diff <(3*s+1); diff++)
{                    g.DrawString(Convert.ToString(Math.Abs(V[Yaxis] - V[Xaxis])), font, myBrush1,
StartX1, StartY1);
   Xaxis = Xaxis -1;
   Yaxis = Yaxis + 1;
   StartX1 = StartX1 - 30;
   StartY1 = StartY1 - 30;

}}}}}
```

Example 3.1: S is Even (S=6)



Example 3.2: S is Odd (S=5)



References:

1. R. L. Graham and N. J. A. Sloane, On additive bases and harmonious graph, SIAM J. Alg. Discrete Math., 1 (1980) 382 – 404.
2. 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.
3. A.Solairaju and K.Chitra Edge-odd graceful labeling of some graphs, Electronics Notes in Discrete Mathematics Volume 33,April 2009,Pages 1.
4. A. Solairaju and P.Muruganantham, even-edge gracefulness of ladder, The Global Journal of Applied Mathematics & Mathematical Sciences(GJ-AMMS). Vol.1.No.2, (July-December-2008):pp.149-153.
5. A. Solairaju and P.Sarangapani, even-edge gracefulness of P_n o nC_5, Preprint (Accepted for publication in Serials Publishers, New Delhi).
6. A.Solairaju, A.Sasikala, C.Vimala Gracefulness of a spanning tree of the graph of product of P_m and C_n, The **Global Journal** of Pure and Applied Mathematics of Mathematical Sciences, Vol. 1, No-2 (July-Dec 2008): pp 133-136.
7. A. Solairaju, C.Vimala,A.Sasikala Gracefulness of a spanning tree of the graph of Cartesian product of S_m and S_n, The **Global Journal** of Pure and Applied Mathematics of Mathematical Sciences, Vol. 1, No-2 (July-Dec 2008): pp117-120.
8. A. Solairaju, N.Abdul ali , s. Abdul saleem Gracefulness of Tp-tree with five levels obtained by java programming, The International Journel of Scientific and Research Publication (IJSRP),Volume 2,Issue 12,December 2012 Edition [ISSN 2250 – 3153]
9. A. Solairaju, N.Abdul ali, Gracefulness of $^{n}\mathbf{C_4}$ Merging with paths, International Organization of Scientific Research (IOSR),Volume 4,Issue 4, 20 December 2012.Paper ID:G22078
10. A. Solairaju, N.Abdul ali, A new class of gracefull trees. International journal of science & engineering research (IJSER),Volume 4,1st dec 2013.paper ID : I01653
11. A. Solairaju, N.Abdul ali, Gracefull ness of $P_K \circ 2_{C_k}$, International Journal of Engineering Research and Technology(IJERT),Volume 1,Issue :10th dec 2012,Paper ID :P12552 [ISSN 2278 – 0181]
12. Tbuan thai & hoang Q.Lan,".Net Framework Essentials",O`reilly,2nd Edition 2007.
13. D.Nikhil Kothari,",.Net Framework",Addison-Wesley Professional,2008.
14. Christian Nagel et al,"Programming in C#",Wrox Publication,2001