ISSN: 0974-5823 Vol. 7 (Special Issue, Jan.-Mar. 2022)

International Journal of Mechanical Engineering

Arithmetic Number Labelling for Banana tree, Olive tree, Shrub, Jelly fish, Tadpole graphs

Dr. A. Uma Maheswari

Associate Professor & Head

PG & Research Department of Mathematics,

Quaid – E–Millath Government College For Women, Chennai – 600 002.

A.S.Purnalakshimi

Research Scholar

PG & Research Department of Mathematics

Quaid – E–Millath Government College For Women, Chennai – 600 002.

Abstract

Labelling is an interesting branch in graph theory which has wide applications in transcriptome analysis, communication network, coding and decoding, knot theory in topology, psychology, data mining, golom rulers, etc. Arithmetic number labelling introduced in 2022 by Uma Maheswari and Purnalakshimi, included Python program coding to generate the n arithmetic numbers. This paper deals with Arithmetic number labelling of Jelly fish graph, Tad pole graph, Shrub, Banana tree and Olive tree. In this paper, Python program coding to generate the vertices of a tadpole graph is presented. Suitable illustrations are given.

Keywords — Arithmetic number, Arithmetic number labelling, Jelly fish, Shrub, Banana tree, Olive tree.

AMS classification: 05C78

Introduction

Graph labelling is assigning of labels represented by integers to the vertices [8][10], edges [7], faces [11] and blocks [6][7] of a graph. Open challenge is obtaining a vertex label, edge label, face label or block label for all graphs, under certain constraints. Labelled graphs are undeniably useful building blocks of mathematical models for a wide range of applications in medical [13}, crystallography [14] data science [15] and communication network [15]. Graph theoretical approach for comparison of observational galaxy distributions is an interesting and novel application [12]. These graph-based methods are used in gene regulatory networks and also in connectomics (nervous systems).

I. PRELIMINARIES

The definitions required for this paper are given below.

Definition 1: Arithmetic number [1]

A number ‘n’ is called an Arithmetic number if the arithmetic mean of its divisors, is an integer.

For example, 15 is an arithmetic number since the arithmetic mean of its divisors 1,3,5 and 15 is an integer. Some of the Arithmetic numbers are 1,3,5,6,7,11,13, etc.

Definition 2: Arithmetic number labelling [2]

An Arithmetic number labelling of a graph G is a one - to – one function $f\colon\ V(G) \rightarrow W$, (where W is the set of whole numbers) that induces a bijection $f^{*}\colon E(G) \rightarrow (A_1, A_2, A_3, \dots, A_n)$, defined by

$f^*(u,v) = |f(u) - f(v)|, \forall\, e = uv \in E(G)$. A={$A_1, A_2, \ldots, A_n$}is a set of Arithmetic numbers.
The graph which admits Arithmetic number labelling is called Arithmetic number graph.

Note: Arithmetic number graph is abbreviated as ANG.

Definition 3: Jelly fish graph [4]

The jelly fish graph, J (k, *l*) is got from a 4 - cycle u, v, w and t by joining the vertices w and t with an edge and appending k pendent edges to u and *l* pendent edges to v.

Definition 4: Shrub graph [4]

St ($n_1, n_2, \ldots, n_k$) is a Shrub graph got by connecting a vertex v_0 to the central vertex of each of k number of stars.

Definition 5: Banana tree [4]

Banana tree graph Bt ($n_1, n_2, \ldots, n_k$) is a graph got by connecting a vertex v_0 to one leaf of each of k numbers of stars.

Definition 6 [4]: Let G be a graph with a fixed vertex v and let (P_k : G) be a graph got from k copies of the path $P_k : u_1, u_2, \ldots, u_k$ by joining u_i with the vertex v of the i^{th} copy of G with an edge for
$1 \le i \le n$.

Definition 7: Olive tree [9]

Olive tree graph (T_l) is a rooted tree consisting of *l* branches and i^{th} branch is a path of length ‘i’.

Definition 8: Tad pole graph [9]

T(k, *l*) is a graph in which path P_l is attached to any one vertex of cycle C_k.

MAIN RESULTS

Theorem 1: For k, *l* ≥ 1, Jelly fish J (k, *l*) graph is ANG.

Proof:

J (k, *l*) is a jelly fish graph.

Let $V(J(k, l)) = \{u, v, w, t, u_i, v_j; 1 \le i \le k, 1 \le j \le l\}$ and

edges $E(J(k, l)) = \{uw, wv, ut, tv. wt, uu_i, v_j; 1 \le i \le k, 1 \le j \le l\}$

Therefore $J(k, l)$ has $(k+l+4)$ number of vertices and $(k+l+5)$ number of edges.

The function $f: V(J(k, l)) \longrightarrow \{A_n\}$ is defined as follows:

$f(u) = 0$
$f(w) = 5$
$f(t) = 13$
$f(v) = 20$
$f(u_i) = A_{2i}$; $1 \le i \le 2$
$f(u_i) = A_{2+I}$; $3 \le i \le 5$
$f(u_i) = A_{4+I}$; $i = 6$
$f(u_i) = A_{i+5}$; $7 \le i \le k$
$f(v_j) = f(v) + A_{k+5+j}$ $1 \le j \le l$

Let f* be the induced edge labeling of f.

$f^*(uw) = 5$
$f^*(ut) = 19$
$f^*(wv) = 15$
$f^*(tv) = 1$
$f^*(uu_i) = A_{2i}$; $1 \le i \le 2$
$f^*(uu_j) = A_{2+I}$; $3 \le i \le 5$
$f^*(uu_j) = A_{4+I}$; $i = 6$
$f^*(uu_i) = A_{i+5}$; $7 \le i \le m$
$f^*(vv_j) = A_{k+5+j:}$ $1 \le j \le l$

$A_1, A_2, \ldots A_{k+l+5}$ are the induced distinct edge labels. Hence the jelly fish graph is ANG.

Example 1: Arithmetic number labelling for Jelly fish J (8,6).

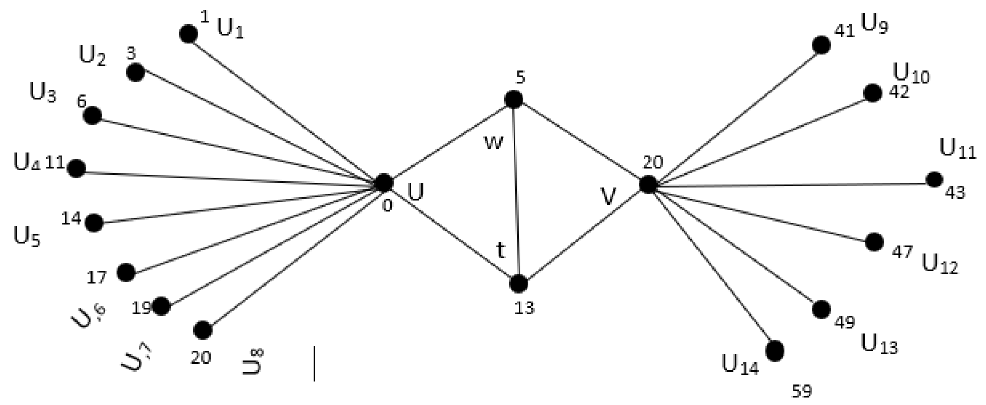

Fig 1: Jelly fish graph J (8,6)

Thus we have proved that Jelly fish graph J(8,6) is ANG.

Theorem 2: For all $n_1, n_2,\ldots,n_k \geq 1$, Shrub St $(n_1,n_2,\ldots,n_k)$ is ANG.

Let v, v_i, v_{ij} ; $1 \leq i \leq k$, $1 \leq j \leq n_i$ be the vertices of Shrub St $(n_1,n_2,\ldots,n_k)$,

Then E [St $(n_1, n_2,\ldots,n_k)$] = $\{vv_i, v_i v_{ij:}\ 1 \leq i \leq k, 1 \leq j \leq l\}$

f: V (St $(n_1,n_2,\ldots,n_k)$) $\longrightarrow$ **$\{A_n\}$** is defined as follows:

$f(v_{0,0}) = A_1$
$f(v_{i,0}) = f(v) + A_{i+1}$; $1 \leq i \leq k$
$f(v_{i,j}) = A_{k+\ n_1 + n_2 + \cdots +\ n_{(i-1)}} + j + 1 + v_i$; $1 \leq i \leq k$, $1 \leq j \leq n_i$

The edge labelling f* is given below:

$f^*(vv_i) = A_{i+1}$; $1 \leq i \leq k$
$f^*(vv_{ij}) = A_{k+\ n_1 + n_2 + \cdots +\ n_{(i-1)}} + j + 1$; $1 \leq i \leq k$, $1 \leq j \leq n_i$

$A_1, A_2, \ldots A_{k+n_1+\ n_2+\cdots n_m+1}$ are the induced distinct edge labels .

Example 2: The Shrub graph St (2,3,4,5) is ANG.

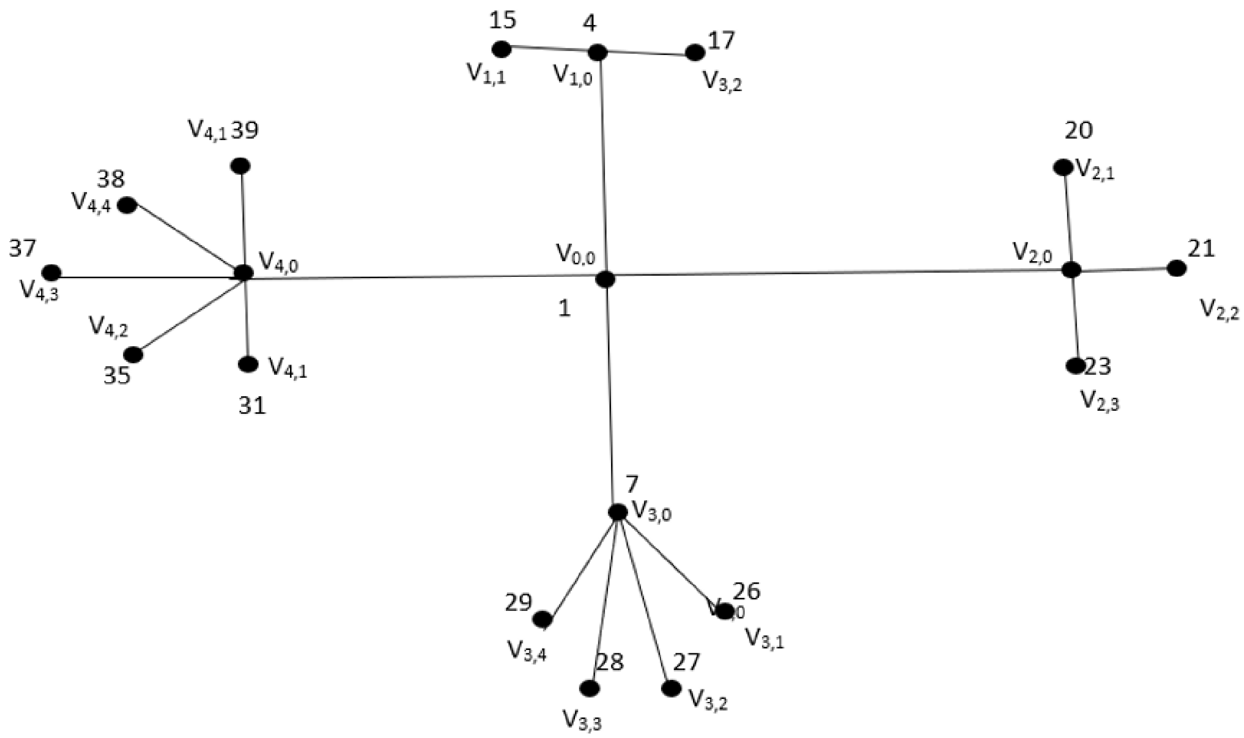

Fig 2: Shrub graph St (2,3,4,5)

Thus we have proved that Shrub graph St (2,3,4,5) is ANG.

Theorem 3: Banana tree graph Bt (k,k,k,…,k) (*l* times) is ANG.

Let V (Bt (k,k,…,k)) = $\{v, v_i, u_i, u_{ij};\ 1 \leq i \leq l,\ 2 \leq j \leq k\}$

Let f: V (Bt (k,k,k,…,k)) ⟶ $\{A_n\}$ is defined as follows:

f $(v_{0,0}) = 0$

f $(v_{1,i}) = A_i$

f $(u_{i,0}) = A_{l+(i-1)k_{(i-1)}+1}$

f $(u_{ij}) = A_{l+(i-1)k_{(i-1)}+1+j} + u_i$

Induced edge labeling f* is given below:

$f^*(vv_i) = A_i$

$f^*(u_i v_i) = A_{l+i}$; $1 \leq i \leq m$

$f^*(u_i u_{ij}) = A_{l+(i-1)k_{(i-1)}+1+j}$; $1 \leq i \leq l,\ 2 \leq j \leq k$

$A_1, A_2, \ldots, A_{m+mn}$ are the induced distinct edge labels.

Example 3: Arithmetic number labelling of Banana tree graph Bt (4,4,4,4,4,4).

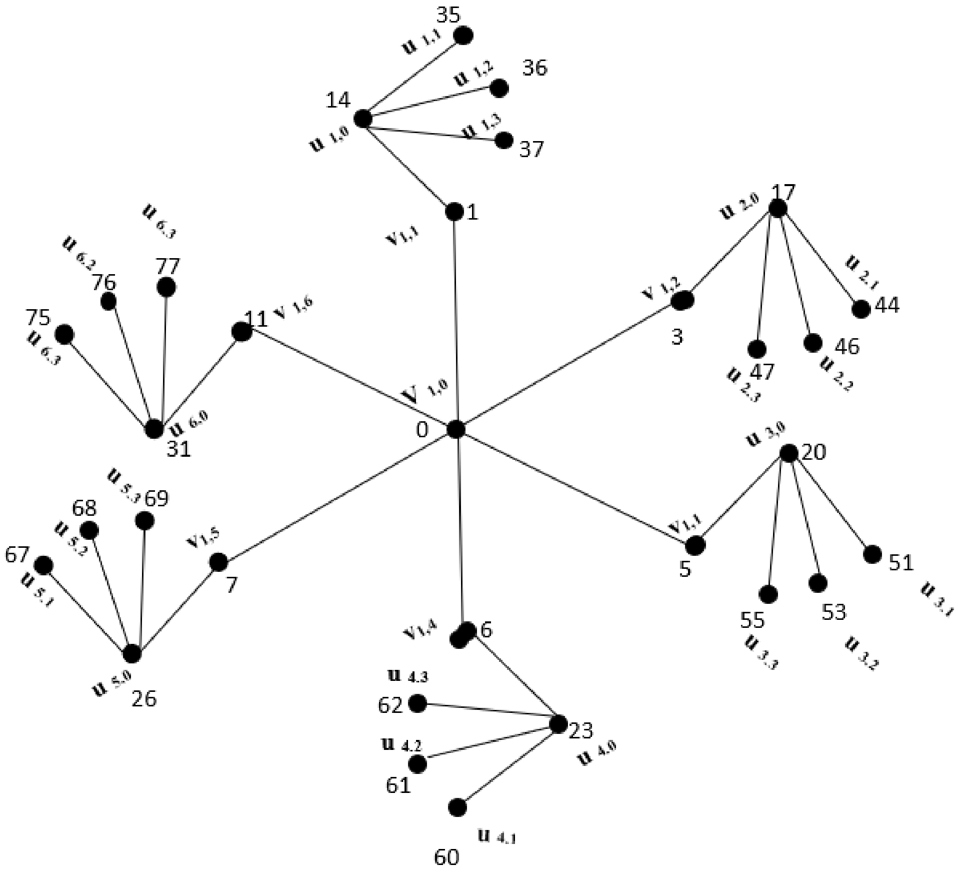

Fig 3: Banana tree Bt (4,4,4,4,4,4)

Thus we have proved that Banana tree Bt (4,4,4,4,4,4) is ANG.

Theorem 4: $(P_k: K_{1,l})$ is ANG for all $k > 1$ and $l \geq 1$

Let $V((P_k: K_{1,l})) = \{ v_i, u_i u_{i\,j} ;\ 1 \leq i \leq k,\ 1 \leq j \leq l\}$

and $E((P_k: K_{1,l})) = \{v_i v_{i+1}, v_j u_j, \underline{u_j u_{jm}} ;\ 1 \leq i \leq k-1,\ 1 \leq j \leq k,\ 1 \leq m \leq l\}$

f: $V((P_k: K_{1,l})) \longrightarrow \{A_n\}$ is defined as follows:

$f(v_{1,0}) = 0$

$f(v_{i,0}) = A_{i-1} + V_{i-1}$

$f(u_{i,0}) = A_{k-1+i} + f(v_i)$

$f(u_{i\,j}) = A_{2k-4+3i+j} + f(u_i)$; $1 \leq i \leq k,\ 1 \leq j \leq l$

The induced edge labeling f* is given below:

$f^*(v_i v_{i+1}) = A_i$; $1 \leq i \leq k\text{-}1$
$f^*(v_j u_j) = A_{n-1+j}$; $1 \leq j \leq k$
$f^*(u_i u_{j\,j}) = A_{2n-4+3i+j}$; $1 \leq i \leq k$, $1 \leq j \leq l$
$A_1, A_2, \ldots, A_{kl+2k-1}$ are the induced distinct edge labels.

Example 4: Arithmetic number labelling of (P_6 ; $K_{1,3}$).

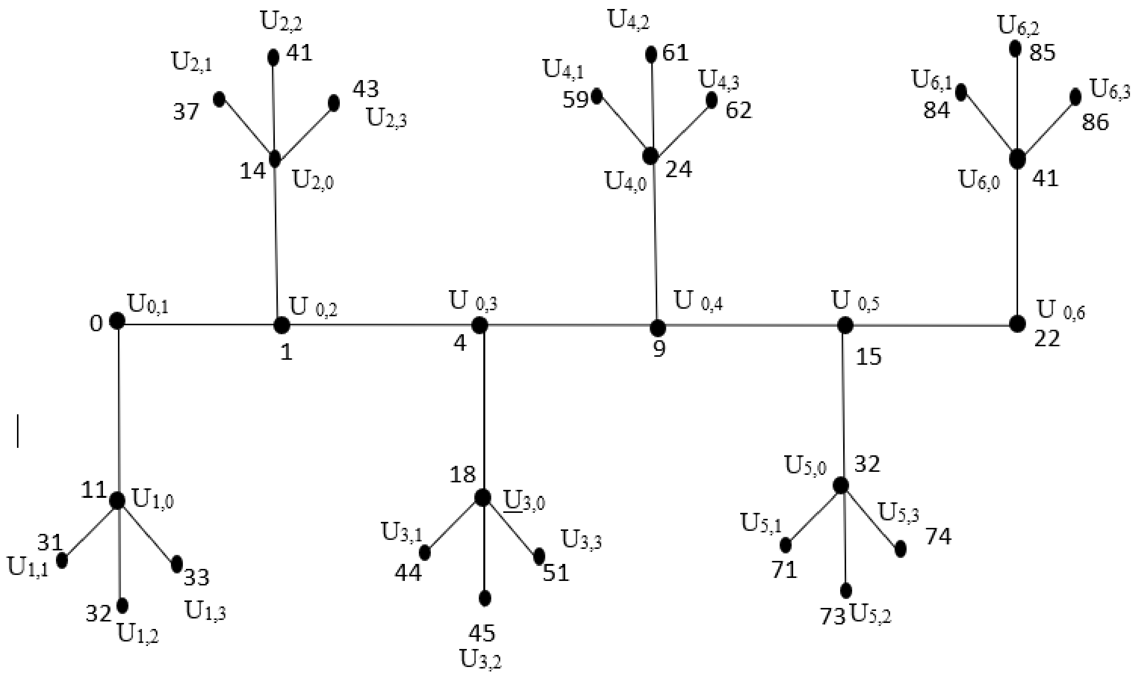

Fig 4: (P_6: $K_{1,3}$)

Thus we have proved that (P_6: $K_{1,3}$) is ANG.

Theorem 5: Olive tree graph (T_l) is ANG.

Let f: $V(T_l) \longrightarrow \{A_n\}$ be defined as follows:
$f(v_{1,0}) = 0$
$f(v_{1,1}) = A_1$
$f(v_{1,2}) = A_2$
$f(v_{1,j}) = A_j$ for all j
$f(v_{i,j}) = A_{l+(l-1)+\ldots+(i-1)\text{ times }+j} + v_{(i-1),j}$

The induced edge labeling f* is given below:

$f^*(v_{1,1}) = A_1$
$f^*(v_{1,2}) = A_2$

$f^*(v_{1,j}) = A_j$ for all j

$f^*(v_{i,j}) = A_{l+(l-1)+\ldots+(i-1)\text{ times }+j}$; $1\leq i \leq l$, $1\leq j \leq l-1$

$A_1, A_2 \ldots A_{l+(l-1)+\ldots+(i-1)\text{ times }+j}$ are the induced distinct edge labels.

Example 5: Arithmetic number labelling of Olive tree graph (T_6).

Let f: $V(T_k) \longrightarrow \{A_n\}$ is defined as follows:

$f(v_{1,0}) = 0$

$f(v_{1,1}) = A_1$

$f(v_{1,2}) = A_2$

$f(v_{1,j}) = A_j$ for all j

$f(v_{i,j}) = A_{k+(k-1)+\ldots+(i-1)\text{ times }+j} + v_{(i-1),j}$

The induced edge labeling of f* is given below:

$f^*(v_{1,1}) = A_1$

$f^*(v_{1,2}) = A_2$

$f^*(v_{1,j}) = A_j$ for all j

$f^*(v_{i,j}) = A_{k+(k-1)+\ldots+(i-1)\text{ times }+j}$; $1\leq i \leq k$, $1\leq j \leq k-1$

$A_1, A_2 \ldots A_{k+(k-1)+\ldots+(i-1)\text{ times }+j}$ are the induced distinct edge labels.

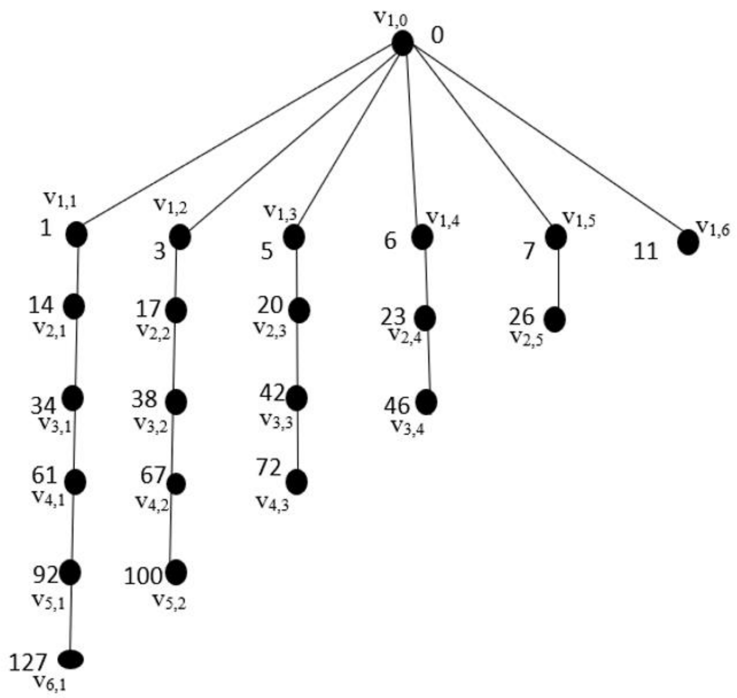

Fig 5: Olive tree (T_6)

Thus we have proved that Olive tree T_6 is ANG.

Theorem 6: Tad pole graph T(3,k) is ANG.

Let f: V(T (3,k)) ⟶ {A_n} is defined as follows:

$f(v_1) = 0$
$f(v_2) = 6$
$f(v_3) = 5$
$f(v_4) = A_2+v_3$
$f(V_i) = A_i+ v_{(i-1)}$ for all $5 \leq i \leq k$
The induced edge labeling of f* is given below:

$f^*(v_1v_2) = 6$
$f^*(v_2v_3) = 1$
$f^*(v_1v_3) = 5$
$f^*(v_3 v_4) = A_2$
$f^*(v_{(i-1)}v_i) = A_i \quad ; 5 \leq i \leq k$
$A_1,A_2 \ldots A_n$ are the induced distinct edge labels.

Example 6: Arithmetic number labelling of Tadpole graph T (3,7).

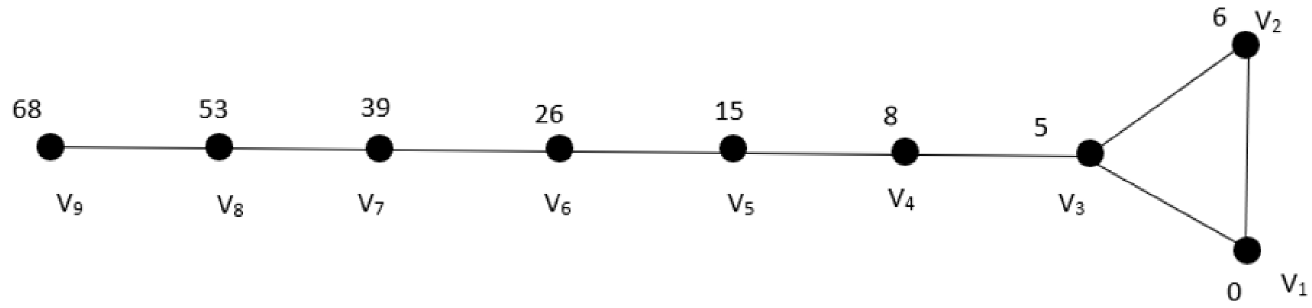

Fig 6: Tad pole graph T(3,7)

Thus we have proved that Tadpole graph is ANG.

Python program coding to generate n Arithmetic numbers in general was already given in [2]. For the sake of completeness, we give the link below:

https://colab.research.google.com/drive/1eZcI5io92XOplEJpYiWDH0w84Qrgpxz5?usp=sharing

Python program coding link to generate the vertex labels of a Tadpole graph is given below:

https://colab.research.google.com/drive/1c8gUKU6VTtKyyGC2eEgxVGO9XbyYhXtF?usp=sharing

```
def findarithmetic(no):
    count=0
    sum=0
    for i in range(1,no+1):
      if(no%i==0):
        count=count+1
        sum=i+sum;
    avg=int(sum/count)
    avg1=sum/count
    if (avg1-avg==0):
          return(no)
    else:
        return(0)
```

```
limit=int(input("Enter your limit : "))
c=0
number=[]
size=1000
for i in range(1,size):
    ans=findarithmetic(i)
    if c<limit  and  ans >0 :
     number.append(ans)
     c=c+1
print(number)
```

```
Enter your limit : 15
[1, 3, 5, 6, 7, 11, 13, 14, 15, 17, 19, 20, 21, 22, 23]
```

```
sbs=['₀','₁','₂','₃','₄','₅','₆','₇','₈','₉']
ssbs=[3,5,6,24,5,8,2,9,7,7,4]

m=25
def subs(n):
  n=str(n)
  for i in n:
    print(sbs[int(i)],end='')

subs(m)
```

```
₂₅
```

```
[2] sbs=['₀','₁','₂','₃','₄','₅','₆','₇','₈','₉']

    def subs(n):
      n=str(n)
      for i in n:
        print(sbs[int(i)],end='')

    def findarithmetic(no):
        count=0
        sum=0
        for i in range(1,no+1):
          if(no%i==0):
            count+=1
            sum+=i
        avg=int(sum/count)
        avg1=sum/count
        if (avg1-avg==0):
              return(no)
        else:
            return(0)
```

```
limit=2000
c=0
number=[]
v=[]
v.append(0)
v.append(6)
v.append(5)
v.append(8)
size=1000
for i in range(1,size):
    ans=findarithmetic(i)
    if c<limit  and  ans >0 :
     number.append(ans)
     c=c+1
```

```
j=int(input("enter the value for i : "))
print("f(v",end='')
subs(j)
print(")=(A",end='')
subs(j)
print("+V",end='')
subs(j-1)
print(")")
for j1 in range(5,20):
  k=v[j1-2] +number[j1-1]
  #print(v[j-2])
  #print(number[j-1])
  v.append(k)
print("A" ,end='')
subs(j)
print("=",end='')
print(number[j-1])
print("v",end='')
subs(j-1)
print("=",end='')
print(v[j-2])
print("f(v",end='')
subs(j)
print(")=",end='')
print(v[j-1])
```

enter the value for i : 6
$f(v_6)=(A_6+V_5)$
$A_6=11$
$v_5=15$
$f(v_6)=26$

enter the value for i : 11
$f(v_{11})=(A_{11}+V_{10})$
$A_{11}=19$
$v_{10}=85$
$f(v_{11})=104$

enter the value for i : 14
$f(v_{14})=(A_{14}+V_{13})$
$A_{14}=22$
$v_{13}=145$
$f(v_{14})=167$

enter the value for i : 18
$f(v_{18})=(A_{18}+V_{17})$
$A_{18}=30$
$v_{17}=246$
$f(v_{18})=276$

Conclusion:

Arithmetic number labelling for Jelly fish, Tadpole graph, Shrub, Banana tree graph, $P_n,k_{1,m}$ and Olive tree graph is given in this paper. Python program coding to generate the vertex labels of a Tadpole graph is given. Many researchers are inspired by the applications of graph labelling in software programming, cryptography, psychotherapy etc. There is further scope for research on Arithmetic number labelling for more graphs with applications to diverse fields.

References:

[1] Oller-Marcén, A. M. (2015). On arithmetic numbers. *Mathematische Nachrichten*, *288*(5-6), 665-669.

[2] A.Uma Maheswari & A.S.Purnalakshimi (2022) Arithmetic number labelling of graphs in “Advances in graph labelling, coloring and power domination theory -Volume 1

[3] Muthumanickavel, G., & Murugan, K. (2020). Oblong Sum Labeling of Union of Some Graphs. World Scientific News, 145, 85-94.

[4] Prema, M., & Murugan, K. (2018). Oblong sum labeling of some special graphs. World Scientific News, 98, 12-22.

[5] Hegde, S.M. & Shankaran, P. (2018). On Triangular Sum Labelling on Graphs, GRD Journals-Global Research and Development Journal in Engineering, Vol.3, No.7, pp. 29-33.

[6] Uma Maheswari, A. & Azhagarasi, S New Labeling for Graphs-AUM Block Sum Labeling, International Journal of Current Science, Vol.12, No.1, pp.574-584.

[7] Uma Maheswari, A. & Azhagarasi, S. (2022b). AUM Block Labelling for Cycle Cactus Block Graphs, Compliance Engineering Journal, Vol.13, No.4, pp.84-96.

[8] Balamurugan, B.J., Thirusangu, K., Thomas, D.G. & Murali, B.J. (2018). k-Zumkeller labeling of graphs. Int. J. Eng. Technol, Vol.7, No.4.10, pp. 460–463.

[9] Deshmukh, U., & Shaikh, V. Y. (2016). Mean cordial labeling of tadpole and olive tree. Annals of Pure and Applied Mathematics, 11(2), 109-116.

[10] Murali, B. J., Thirusangu, K., & Madura Meenakshi, R. (2016). Zumkeller cordial labeling of graphs. In Computational Intelligence, Cyber Security and Computational Models (pp. 533-541). Springer, Singapore.

[11] Martin Baca, “Face Antimagic labelings of Convex Polytopes”, Utilitas Mathematica 55(1999),pp 221-226.

[12] Ueda, H., Takeuchi, T. T., & Itoh, M. (2003). A graph-theoretical approach for comparison of observational galaxy distributions with cosmological N-body simulations. Astronomy & Astrophysics, 399(1), 1-7.

[13] Maheswari, A. U., & Purnalakshimi, A. S. Graph Theory in the Analysis of Arithmophobia.

[14] Prathik, A., Uma, K., & Anuradha, J. (2016). An Overview of application of Graph theory. International Journal of ChemTech Research, 9(2), 242-248.

[15] Dehmer, M., Emmert-Streib, F., & Shi, Y. (2017). Quantitative graph theory: a new branch of graph theory and network science. Information Sciences, 418, 575-580.