Araştırma Makalesi
Adıyaman Üniversitesi
Mühendislik Bilimleri Dergisi
15 (2021) 414-426

6'YA 6 TAHTA ÜZERİNDE AT KAPLAMA PROBLEMİNİ ÇÖZMEK İÇİN DENETİMSİZ MAKİNE ÖĞRENME ALGORİTMASI

Serkan Güldal[1]

[1]Adiyaman University, Faculty of Arts and Sciences, Department of Physics, Adiyaman, 02040, Turkey

Geliş Tarihi/Received Date: 09.08.2021 Kabul Tarihi/Accepted Date: 25.10.2021 DOI: 10.54365/adyumbd.980660

ÖZET

Modülerlik, çizgelerden bilgi çıkarmak için, çok kullanılan bir makine öğrenimi algoritmasıdır. Modülerlik, özünde, ele alınan ağı daha küçük kümelere böler. Oluşturulan kümeler, aynı kümedeki düğümler arasındaki paylaşılan özellikleri vurgular. Bu çalışmada 6×6 at çizgesini modülerlik yöntemiyle analiz ederek 6 At Kaplama Probleminin (6-AKP) çözümlerini elde ettik. Araştırmamız 0,1 ile 2,0 arasındaki çözünürlükler için değişmektedir. Çözünürlük 1,2 için bulunan maksimum modülerlik puanı 0,318'dir. 0,3 ve 0,4 olmak üzere çözünürlükler, tüm çözümleri 8 at ile tanımladı. Ayrıca, 0,2 çözünürlükler için 8 at ve 0,2, 0,3 ve 0,4 çözünürlükler için 9, 10, 11, 12, 13 at ile bazı çözümler elde edilmektedir. Ayrıca, çözünürlük 0,3, 6-AKP çözümlerini bulmak için en verimli çözünürlüktür. Ayrıca, analizlerimiz gösterdi ki 0,2 çözünürlüğü, 6-AKP'nin 195 çözümünü daha fazla çözüm bulmak için en iyi çözünürlüktür. Son olarak, modülerlik yöntemi, 2253 çözüm arasından 0,5 çözünürlük için 61'den, 0,2 çözünürlük için 195'e kadar olan çözümleri çıkarıyor.

Anahtar Kelimeler: *At çizgesi, modülerlik, At Kaplama Problemi, Makine öğrenmesi*

UNSUPERVISED MACHINE LEARNING ALGORITHM TO SOLVE KNIGHT COVERING PROBLEM FOR 6 BY 6 BOARD

ABSTRACT

Modularity is a well-known method as a machine-learning algorithm to extract information from graphs. The modularity, in essence, divides the considered network into smaller clusters. The extracted clusters highlight the shared properties between the nodes in the same cluster. In the present study, we analyze the 6×6 knight graph by modularity method to obtain 6 Knight Covering Problem (6-KCP) solutions. Our investigation is ranged for the resolutions from 0.1 to 2.0. The maximum modularity score is 0.318 found for resolution 1.2. The resolutions, namely 0.3 and 0.4, identified all solutions, by 8 knights. Moreover, some solutions are obtained by 8 knights for the resolutions 0.2 and by 9, 10, 11, 12, 13 knights for the resolutions 0.2, 0.3, and 0.4. Moreover, resolution 0.3 is the most efficient resolution to find 6-KCP solutions. Also, within our analysis, resolution 0.2 is the best resolution to find more solutions, 195 solutions of 6-KCP. Lastly, the modularity method extracts the solutions from 61, for resolution 0.5, to 195, for resolution 0.2, out of 2253 solutions.

Keywords: *Knight graph, modularity, Knight Covering Problem, Machine learning*

1. Introduction

The Knight Covering Problem (KCP a.k.a. N-KCP) is an interesting problem for computer scientists since there is no analytical solution. The problem is introduced based on the knights' movements on the chess-board-likes. The uniqueness of knight moves became subject to various researches such as the knight's tour problem [1-5] extended problems [6-8]. image encryption [9-13] and the N-KCP [14-21]. In the N-KCP, the knights are placed on a N×N board in such a way so no knights attack each other and the placed knights cover the board by either occupy or threaten. The N-KCP is an NP-Complete problem with Quadratic Time Complexity: $O(n^2)$. Nevertheless, there are many numerical methods are developed to identify some or all KCP solutions such as the independent set [22, 23] and the Girvan-Newman clustering algorithm [24] of the responding knight graphs of N-KCP. Additionally, modularity algorithm is introduced for 3-KCP [25], 4-KCP [26], and 5-KCP [27]. At present, the knight

[1] e-mail: sguldal@adiyaman.edu.tr ORCID ID: https://orcid.org/0000-0002-4247-0786 (Sorumlu Yazar),

graph representation of 6-KCP is introduced and classified by the modularity. The classified graph is analyzed by means of 6-KCP solutions. The details of the analysis are discussed in the following sections.

6-KCP is the problem to place a certain number of knights on 6 by 6 chessboard-like, so every cell is either occupied or threatened. In Figure 1, 2 of the 6-KCP solutions which are found by binary graph method [22] are depicted. The solutions consist of 18 knights on the 6 by 6 boards. The solutions are shown in Figure 1 are rotationally, vertically, and horizontally symmetric solutions Thus, they represent only 1 unique solution.

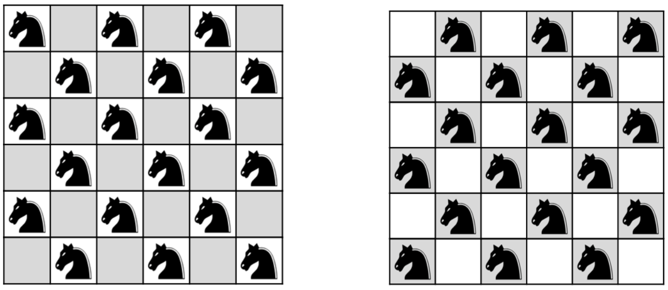
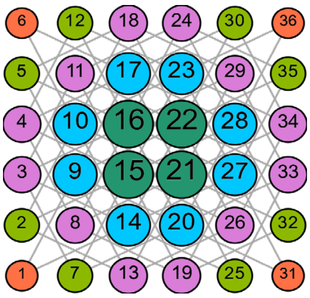


Figure 1. 2 of the 6-KCP solutions by utilizing the binary graph method.

We intend to use the unsupervised clustering algorithm, namely modularity, to solve 6-KCP. Thus, the first step of the algorithm is the conversion of the board (6x6) to the knight graph. Figure 2 shows the particular knight graph in which every node is colored and proportionally sized with respect to the degrees of the nodes. Every cell on the board is represented by a node on the board and is labeled by an index number. To highlight the indexing, the graph is presented in the board layout. The cells on the corners (colored orange) can cover 2 cells. The cells on the edges (colored green) can cover 3 cells, and the cells (colored purple) are next can cover 4 cells. The cells, colored light blue, can cover 6 cells. Lastly, 4 cells (colored dark green) can cover 8 cells. The cells which are covered are reachable by one legal knight move. Hence, the cells in which the knight covers and an extra cell is occupied by the particular knight. To sum up, the graph form of 6-KCP is composed of 36 nodes and 80 edges. The nodes have 2, 3, 4, 6, and 8 degrees. They are distributed from the entire graph by the portion respectively 11.11%, 22.22%, 33.33%, 22.22%, and 11.11%. Every knight and their place on the board is explicitly shown in Figure 3.



Color code	Degree	Number of Nodes	Percentage in the graph (%)
●	2	4	11.11
●	3	8	22.22
●	4	12	33.33
●	6	8	22.22
●	8	4	11.11

Figure 2. 6-KCP graph is depicted with the nodes that are colored and sized proportionally to the degrees of nodes. Please see Figure 3 for a more detailed view of relations.

The graph analyses provide important information to extract information from data presented in the form of networks such as computational networks [28-30], social networks [31-34], biological networks [35-37], word networks [38-40], infection networks [41, 42]. One of the many analysis methods is clustering which divides the whole network into smaller relational clusters of the nodes. Each cluster presents a relatively stronger relation of included nodes in the same cluster. The clustering algorithms are extensively utilized since it is relatively computationally efficient. There are numerous graph clustering algorithms which are based on different properties of the graph, so they end up with different clusters. For example, the Girvan-Newman algorithm is based on edge betweenness, Highly Connected Clusters utilizes graph connectivity, k-means clustering divides the network by mean value, and Modularity generates modules (a.k.a. clusters) by means of the strength of division of a network. In this study, we investigated 6-KCP by modularity. The modularity algorithm has many advantages such as it is a flexible algorithm by the changing resolution. To increase resolutions divide the network to a greater number of clusters and to decrease the resolution is a lesser number of clusters. The appropriate resolution can be found based on the intention and the modularity score. Since we intend to provide 6-KCP solutions by modularity application, we changed the resolution from 0.1 to 2.0 which are all meaningful clusters.

2. Unsupervised Machine Learning

Unsupervised machine learning algorithms have a wide range of applications such as image processing [43-45], digital signal processing [46, 47], biomedical research [48], segmentation [49-52]. Likewise, in this study, we have applied the unsupervised learning algorithm, namely modularity. Knight graph for 6×6 board, as shown in Figure 2, is clustered to extract the relational information, so the relational information provides heuristic information in the sense of 6-KCP solution.

There are developed algorithms to solve N-KCP based on knight graphs [22]. In Figure 1, two of the 6-KCP solutions are presented which are found by the binary layout of the graph. Likewise, in this study, we are benefited by graph forms in the search of 6-KCP solutions. Specifically, we use modularity to analyze the 6-KCP graph to find solutions. The considered analysis method divides the 6-KCP network into densely connected smaller clusters. The clustered nodes present a stronger relationship between the knights. Thus, this heuristic highlights the knights which are less likely to be in the same solution. The details of the modularity method and the solution algorithm is introduced in the following sections.

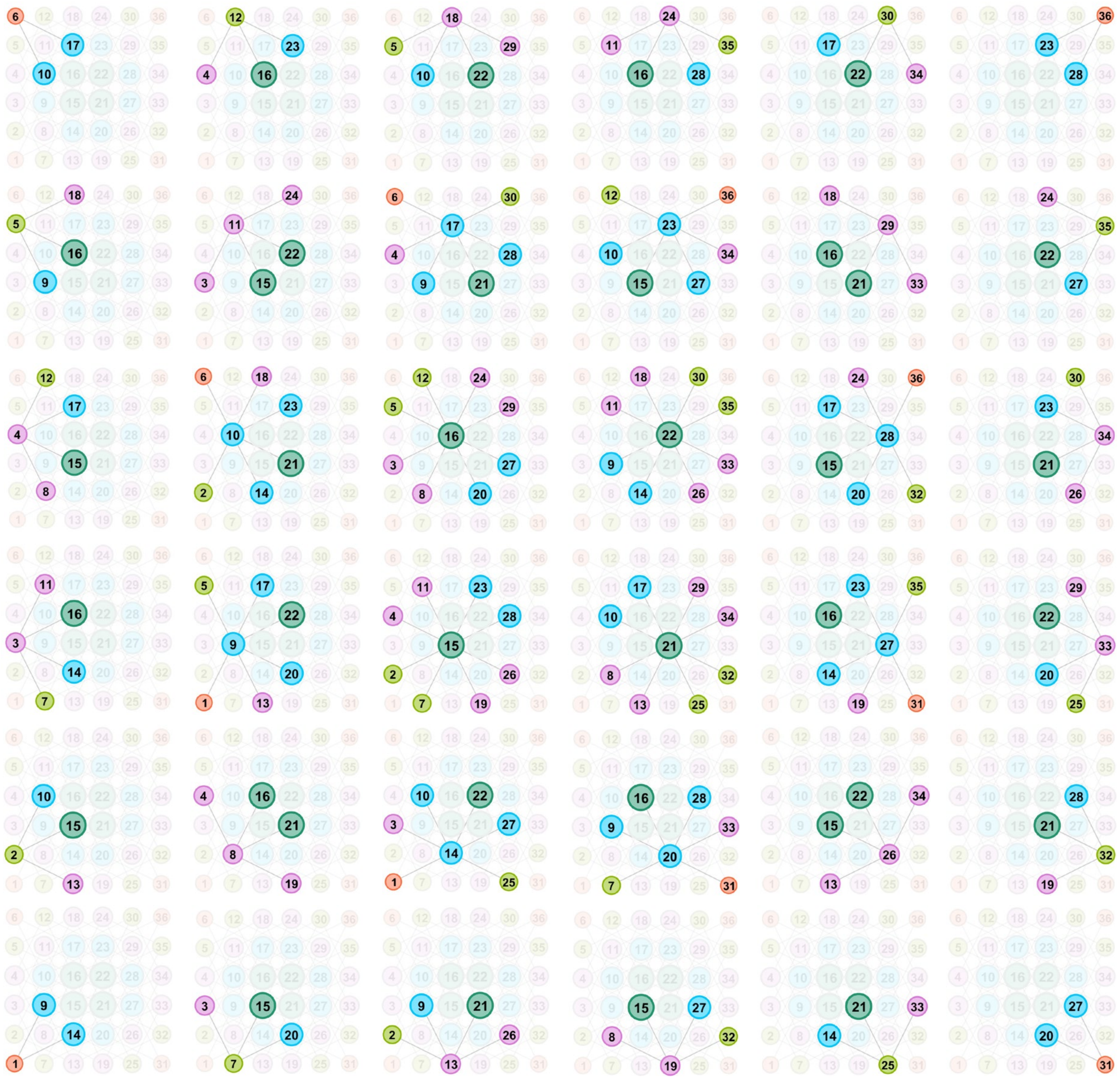


Figure 3. 6-KCP has 36 cells to place a knight. Each cell is represented by a node, and they are connected to the nodes which they can cover

3. Modularity

We used the modularity to identify the closely related knights for the 6-KCP. The modularity score is calculated by various formulas. The formula which we utilized is as follows [53, 54]

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \gamma \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \quad (1)$$

where m stands for the number of edges in the graph. A_{ij} represents the weights of the edge between nodes i and j. γ is the resolution parameter. δ-function is 1 if $c_i = c_j$; in other words, node i and j are in the same cluster and 0 otherwise. k_i is the degree of node i and k_j is the degree of node j.

Throughout our analysis, we used the Gephi [55-57], and the resolutions are limited from 0.1 to 2.0 which is defined specific to the 6-KCP graph. The analysis and implementation results will be given in the Results and Discussion section.

4. Results and Discussion

The investigated relational information of the 6-KCP graph by modularity score extracted communities from 1 to 16 with respect to the changing modularity resolution.

The modularity identifies the strong relationships between nodes. However, the 6-KCP solutions are to place the knight which should have weak/no relations in the same solution. Thus, the extracted clusters reveal the list of positions that are the least likely to be in the same solutions.

In Figure 5, modularity results on 6-KCP (for resolution = 0.1 – 2.0) graphs are presented. The resolutions 1.9 and 2.0 is extracted 1 cluster in 6-KCP graph as shown in Figure 5.s and t. Thus, in the generated possible solutions from the graphs, there is no two nodes could be included simultaneously in a solution, so no solution is identified. In Figure 5.o, p, q, and r, 2 clusters are generated by modularity with the resolutions 1.5, 1.6, 1.7, and 1.8 respectively. Likewise, 2 nodes (a.k.a. 2 knights) solutions do not exist for 6-KCP. Likewise, modularity application for resolution 1.4 cannot generate 6-KCP solutions since there is only 3 clusters. For the resolution = 1.1, 1.2 and 1.3, in Figure 5.k, l, and m, the 6-KCP graph is divided into 4 clusters which leads to no solutions. For resolutions = 0.7, 0.8, 0.9, 1.0, 6-KCP graph is divided to 6 cluster with different topology therefore no solution generated. The resolution is 0.6 extracts 6 clusters again which is not sufficient for 6-KCP solutions. The resolution 0.5 divides 12 clusters, so 61 solutions are identified. Resolution 0.4 leads to 80 solutions with length 8, 9, 10, 11, and 12. The resolution 0.3 is found the solutions for changing lengths from 8 and 13, and the total number of solutions are 185. The resolution 0.2 results with 2 solutions with length 8 and 13, also, 6, 36, 48, 101 solutions with length 9,10, 11, 12 respectively. In summary, the modularity mostly covers the solutions with length from 8 to 13. The relation with the number of clusters and found solutions are summarized in Figure 10.

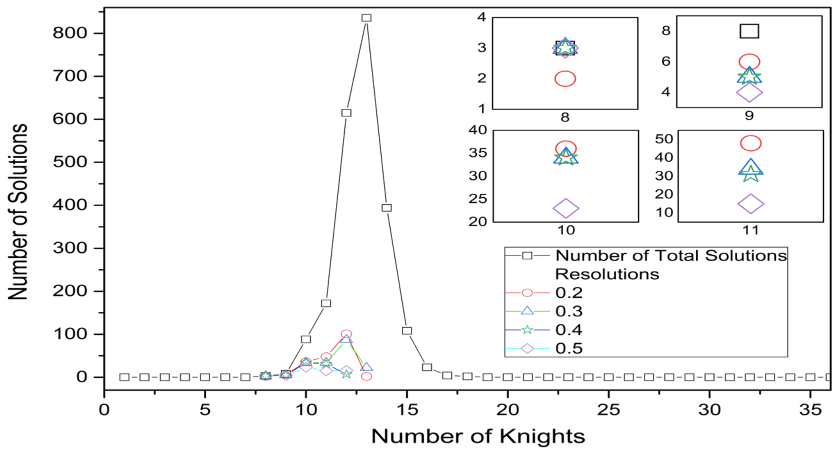


Figure 4. (Color Online) Black empty squares show the number of total solutions by the specified number of knights. The red circle, upside blue triangle, and downside triangle show the number of found solutions for the resolutions 0.2, 0.3, and 0.4 respectively.

The modularity identifies some solutions of 6-KCP. Figure 7 presents the number of solutions vs the number of required knights. The found solutions are limited to length 8 and 13. While resolution 0.2 has the highest capacity for the solution places more knights on the board such as 11 and 12, resolution 0.5 finds less number of solutions such as 9, 10, and 11.

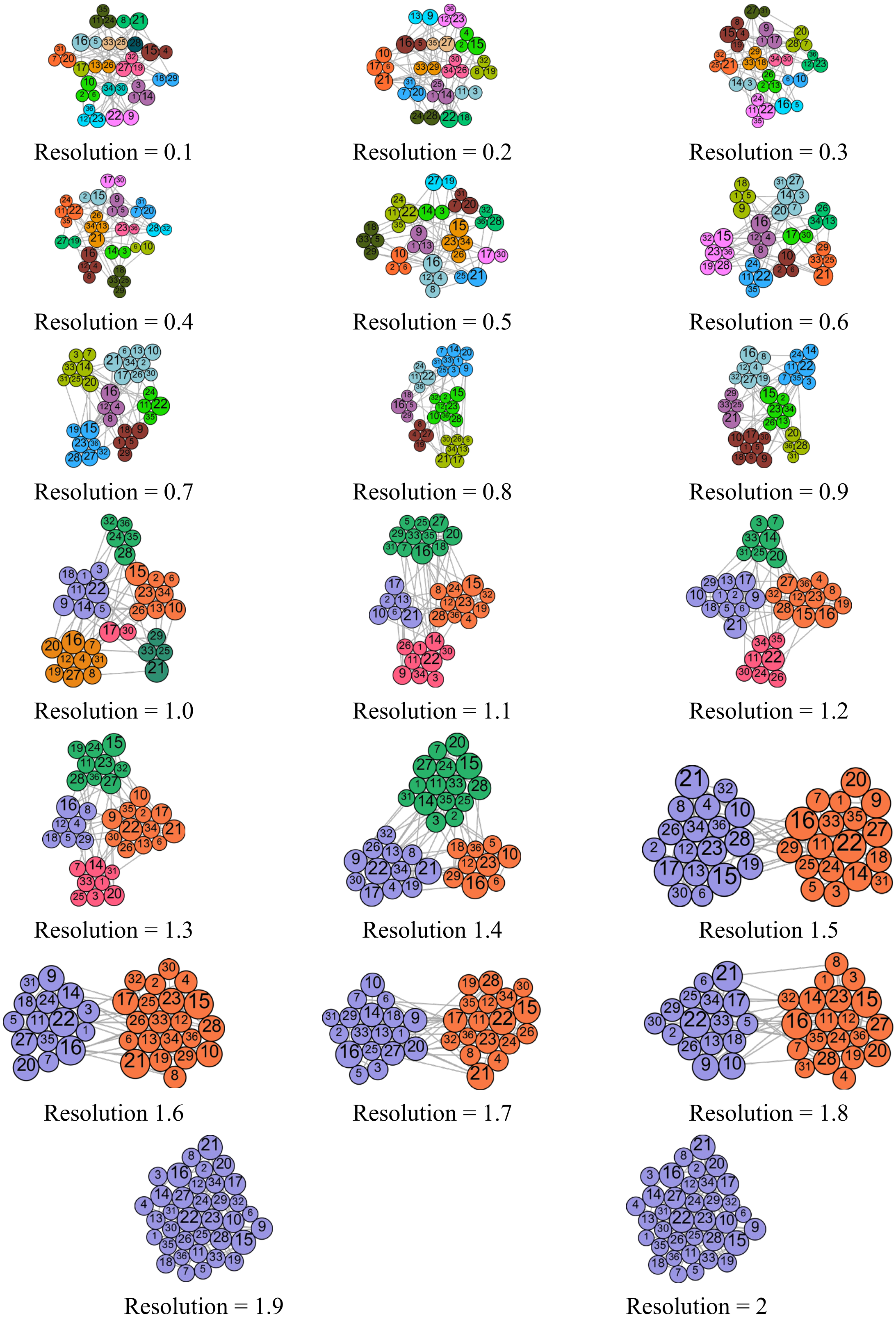


Figure 5. The modularity method is applied to 6-KCP graph for various resolutions from 0.1 to 2.0

The applied modularities for the resolutions between 0.1 to 2.0 clustered the 6-KCP graph. In Figure 6, the increasing resolutions divide the network into smaller numbers of clusters. However, the modularity score does not follow any particular trend. The maximum modularity score is 0.318 for the resolution 1.2, so the best clusters in the sense of modularity algorithm. The investigations show the increasing number of clusters more likely to find 6-KCP solutions (See Figure 4 and Figure 7). The modularity score as a quality measurement shows no explicit correlation with the number of found solutions for the same resolutions as shown in Figure 6. The highest modularity score is 0.318 (for resolution 1.2). Thus, clustering for 1.2 presents the best sub-communities by means of modularity.

In Figure 7, the number of generated permutations is compared with the number of identified solutions for the resolution between 0.1 and 1.2. Although communities are identified, no solutions are obtained for the resolutions 1.2-1,8. The resolution 0.1 leads to 107495424 permutations which is beyond our computational capability. The resolutions 0.2, 0.3, 0.4, 0.5, and 0.6 generates 44789760, 24883200, 21870000, 12960000, and 1470000permutations respectively. These permutations are ended with several solutions such as 195, 185, 80, 61 respectively. The relational information is depicted in computational efficiency (See Figure 8).

The number of generated permutations has a strong correlation with the identified solutions. Thus, the computational efficiency of the particular resolution shows the efficiency of this correlation. The computational efficiency of a resolution is defined in equation 2 and presented with changing resolution in Figure 8. The computational efficiency is formulated as:

$$Efficiency\ of\ the\ cluster = \frac{Number\ of\ found\ solutions * 100}{Number\ of\ permutations} \tag{2}$$

Resolution 0.2 introduces the best clustering by means of 6-KCP, 195 solutions. However, it is not computationally efficient (4.35367×10^{-4}) because it is one of the resolutions which generate the highest number of permutations. The most computationally efficient resolution is 0.3 by 7.43474×10^{-4}. It finds 185 solutions by 24883200 permutations. Thus, resolution 1.0 extracts relatively more meaningful clusters for solution identification of 6-KCP.

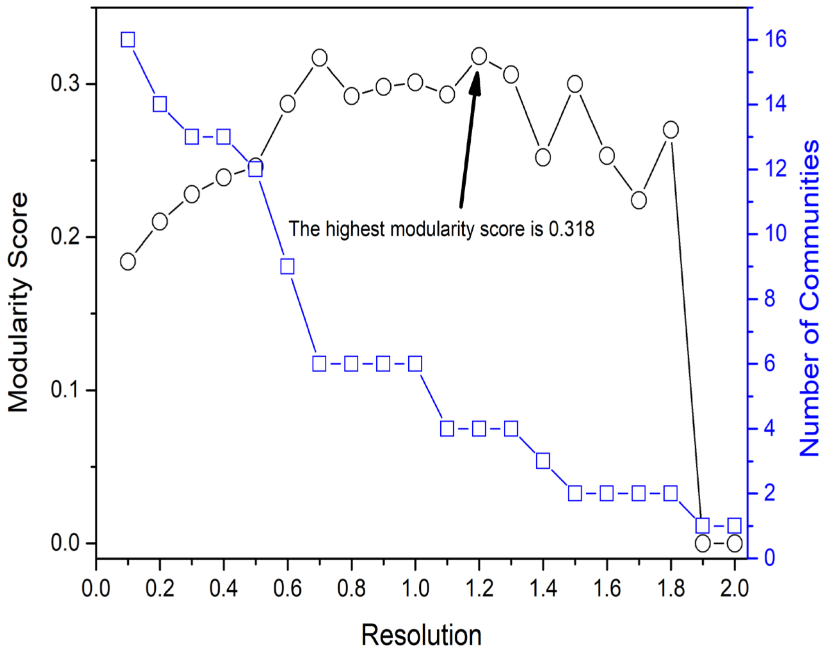


Figure 6. (Color online) While the resolution increase causes to the lower number of communities for 6-KCP graph, the modularity score does not follow a particular pattern

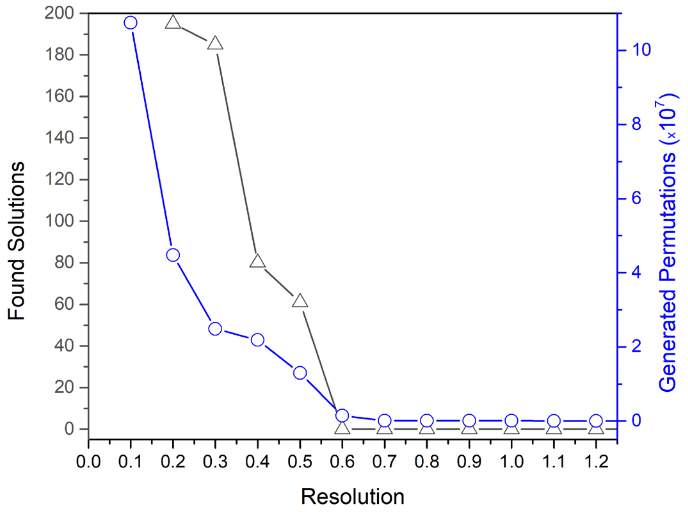


Figure 7. (Color online) The number of found solutions are strongly related to the number of permutations.

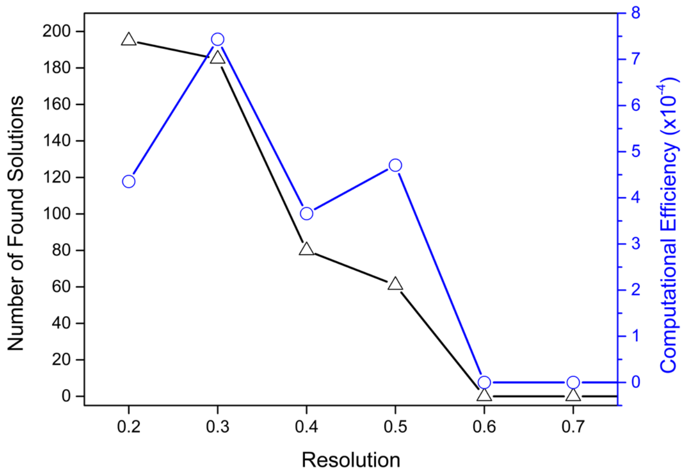


Figure 8. (Color online) Resolution 0.3 has the highest computational efficiency concerning the other resolutions. However, the highest number of solutions, namely 195, is obtained by resolution 0.2 which is considerably computationally less efficient. Our analysis does not include resolution 0.1.

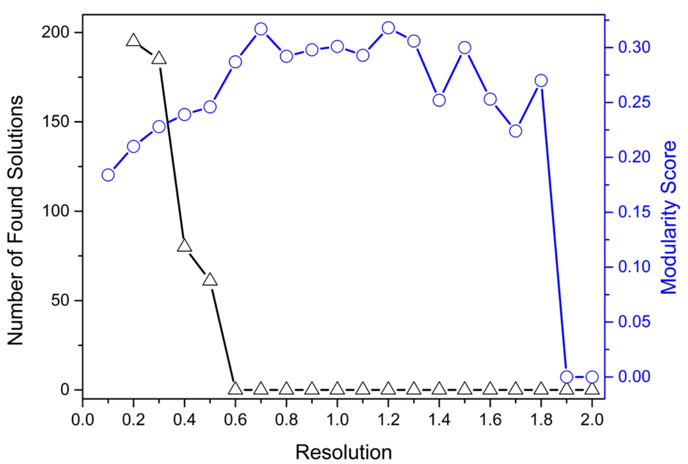


Figure 9. (Color online) Modularity score has no explicit relation with the number of found solutions for the 6-KCP graph in the application of modularity algorithm

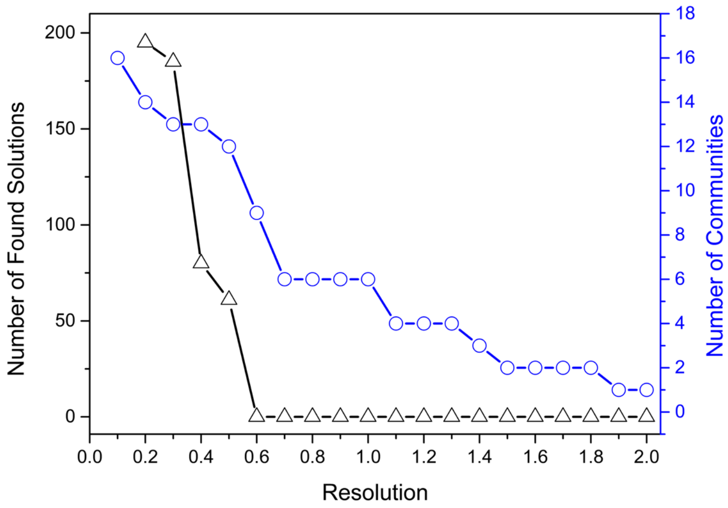


Figure 10. The number of communities increases up to reach a threshold and then it starts to identify the 6-KCP solution.

5. Conclusion

In this study, we have applied an unsupervised learning algorithm namely the Modularity on 6-KCP. The analyses show resolution 0.3 is the computationally efficient resolution to find some solutions of 6-KCP. Moreover, the analysis shows resolution 0.2 is the best resolution to find some solutions of 6-KCP. The maximum modularity score is 0.318 found for resolution 1.2. Moreover, resolution 0.2 is the best resolution to find more solutions of 6-KCP. However, no solution is identified for that particular resolution. Lastly, modularity extracts the solutions from 61 to 195 out of 2253 solutions.

Acknowledgment

The authors declare no conflict of interest.

References

[1] S. Bai, G. Zhu, and J. Huang, "An Intelligent Algorithm for the (1,2,2)-Generalized Knight's Tour Problem," in *2013 Ninth International Conference on Computational Intelligence and Security*, 14-15 Dec. 2013 2013, pp. 583-588, doi: 10.1109/CIS.2013.129.

[2] H. Jian and B. Sen, "An Efficient Algorithm for the Generalized (1,k)-Knight's Tours Problem," in *2009 First International Workshop on Education Technology and Computer Science*, 7-8 March 2009 2009, vol. 1, pp. 697-701, doi: 10.1109/ETCS.2009.161.

[3] P. Hingston and G. Kendall, *Ant Colonies Discover Knight's Tours*. 2004, pp. 1213-1218.

[4] S. Bai, X. Liao, X. Qu, and Y. Liu, "Generalized Knight's Tour Problem and Its Solutions Algorithm," in *2006 International Conference on Computational Intelligence and Security*, 3-6 Nov. 2006 2006, vol. 1, pp. 570-573, doi: 10.1109/ICCIAS.2006.294200.

[5] I. Parberry, "An Efficient Algorithm for the Knight's Tour Problem," *Discrete Applied Mathematics,* vol. 73, pp. 251-260, 03/01 1997, doi: 10.1016/S0166-218X(96)00010-8.

[6] J. Demaio and B. Mathew, "Which Chessboards have a Closed Knight's Tour within the Rectangular Prism?," *Electr. J. Comb.,* vol. 18, 01/05 2011, doi: 10.37236/495.

[7] A. Kumar, "Non-crossing Knight's Tour in 3-Dimension," 03/29 2008.

[8] P. Aliquippa and Pennsylvania, "Thematic Knight's Tour Quotes " 06/29 2020.

[9] A. Philip, "A Generalized Pseudo-Knight?s Tour Algorithm for Encryption of an Image," *IEEE Potentials,* vol. 32, no. 6, pp. 10-16, 2013, doi: 10.1109/MPOT.2012.2219651.

[10] J. Kumar and S. Nirmala, "Securing the contents of document images using knight moves and genetic approach," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 10-13 Aug. 2015 2015, pp. 1091-1095, doi: 10.1109/ICACCI.2015.7275755.

[11] J. Delei, B. Sen, and D. Wenming, "An Image Encryption Algorithm Based on Knight's Tour and Slip Encryption-Filter," in *2008 International Conference on Computer Science and*

Software Engineering, 12-14 Dec. 2008 2008, vol. 1, pp. 251-255, doi: 10.1109/CSSE.2008.1142.
[12] M. Singh, A. Kakkar, and M. Singh, "Image Encryption Scheme Based on Knight's Tour Problem," *Procedia Computer Science,* vol. 70, pp. 245-250, 12/31 2015, doi: 10.1016/j.procs.2015.10.081.
[13] Q. B. Hou, X. F. Yang, Y. S. Wang, and X. S. Huang, "An image scrambling algorithm based on wavelet transform and Knight's tour," vol. 41, pp. 369-375, 02/01 2004.
[14] D. C. Fisher, "On the n x n Knight Cover Problem," *Ars Comb.,* vol. 69, / 2003.
[15] F. Rubin, "Improved Knight Coverings," *Ars Combinatoria,* vol. 69, / 2003.
[16] F. Rubin, "Knight Covers for the 50x50 Chessboard," presented at the Mathfest 2004, Providence RI, 2004.
[17] F. Rubin, "A Family of Efficient Knight Covering Patterns," *Journal of Recreational Mathematics,* Article vol. 33, no. 3, pp. 165-175, 2005.
[18] F. Rubin, "An Improved Method for Finding Knight Covers," *Ars Combinatoria,* vol. 82, / 2007.
[19] B. Lemaire, "Knights Covers on NxN Chessboards," *J. Recr. Math.,* vol. 31, pp. 87-99, 2003.
[20] A. H. Jackson and R. P. Pargas, "Solutions to the NxN Knights Covering Problem," *Journal of Recreational Mathematics* vol. 23, pp. 255-267, 1991.
[21] F. Wei, "Research on Knight Covering Based on Breadth First Search Algorithm," (in English), *Applied Mechanic and Materials,* vol. 686, pp. 377-380, 2014.
[22] S. Güldal, M. Lipscomb, and M. M. Tanik, "Solving Knights Covering Problem: Backtracking, Permutation, Bipartite Graph, and Independent Set," presented at the Nineteenth Annual Early Career Technical Conference, Birmingham, Alabama USA, 2019.
[23] S. Güldal, M. M. Tanik, and M. M. Lipscomb, "Solving Knights Covering Problem by a Hybrid Algorithm," presented at the IEEE SouthEastConn, Huntsville, Alabama, April 11 - 14 2019.
[24] S. Güldal, "Connectives of Knights Covering Problem By Girvan-Newman Clustering," presented at the SDPS 2019 Workshop, Madrid, Spain, November 25-26 2019, 2019.
[25] S. Güldal, "Unsupervised Machine Learning Algorithms to Find 3-KCP Solution: Modularity, Clique Percolation, Spectral, Centrality, and Hierarchical Clustering," *8,* 2021, doi: https://dergipark.org.tr/tr/pub/adyumbd/927555.
[26] S. Güldal, "4×4 Knight’s Graph Analysis by Modularity A Knight Graph Application," *16,* 2021, doi: https://dergipark.org.tr/tr/pub/tjst/710151.
[27] S. Güldal, "Identification of Knights’ Relations for 5×5 Knight Graph by Modularity," *7,* 2021, doi: https://dergipark.org.tr/tr/pub/adyumbd/792772.
[28] "Analysis of the Application of Artificial Intelligence in Computer Networks Technology," *IOP Conference Series: Materials Science and Engineering,* vol. 750, p. 012097, 03/24 2020, doi: 10.1088/1757-899X/750/1/012097.
[29] C. Engström and S. Silvestrov, "PageRank for networks, graphs, and Markov chains," *Theory of Probability and Mathematical Statistics,* vol. 96, p. 1, 10/05 2018, doi: 10.1090/tpms/1034.
[30] Y. L. Karpov, I. Volkova, A. A. Vylitok, L. Karpov, and Y. G. Smetanin, "Designing classes’ interfaces for neural network graph model," *Proceedings of the Institute for System Programming of RAS,* vol. 31, pp. 97-112, 10/01 2019, doi: 10.15514/ISPRAS-2019-31(4)-6.
[31] M. Gençer, "Sosyal Ağ Analizi Yöntemlerine Bir Bakış," *Yildiz Social Science Review,* vol. 3, pp. 19-34, 12/15 2017.
[32] P. Nerurkar, M. Chandane, and S. Bhirud, "Understanding attribute and social circle correlation in social networks," *Turkish Journal of Electrical Engineering and Computer Sciences,* vol. 27, pp. 1228-1242, 03/22 2019, doi: 10.3906/elk-1806-91.
[33] N. Almolhem, Y. Rahal, and M. Dakkak, "Social network analysis in Telecom data," *Journal of Big Data,* vol. 6, 12/01 2019, doi: 10.1186/s40537-019-0264-6.
[34] M. E. J. Newman, "Modularity and community structure in networks," vol. 103, no. 23, pp. 8577-8582, 2006, doi: 10.1073/pnas.0601602103 %J Proceedings of the National Academy of Sciences.
[35] K. Kugler, L. Mueller, A. Graber, and M. Dehmer, "Integrative Network Biology: Graph Prototyping for Co-Expression Cancer Networks," *PloS one,* vol. 6, p. e22843, 07/29 2011, doi: 10.1371/journal.pone.0022843.

[36] B. Eckman and P. Brown, "Graph data management for molecular and cell biology," *IBM Journal of Research and Development,* vol. 50, pp. 545-560, 12/01 2006, doi: 10.1147/rd.506.0545.

[37] V. Gadiyaram, S. Vishveshwara, and S. Vishveshwara, *From Quantum Chemistry to Networks in Biology: A Graph Spectral Approach to Protein Structure Analyses*. 2019.

[38] L. Johnsen, "Graph Analysis of Word Networks," *http://ceur-ws.org/Vol-2021/,* vol. 2021, 01/01 2017.

[39] E. Hasanah and D. Agustiningsih, *Analysis of "Halal" Word in Social Media Using Text Mining and Word Networking*. 2020.

[40] R. Ozcelik, G. Uludoğan, S. Parlar, Ö. Bakay, O. Ergelen, and O. Yildiz, *User Interface for Turkish Word Network KeNet*. 2019, pp. 1-4.

[41] S. Valverde, B. Vidiella Rocamora, R. Montañez Martínez, A. Fraile, S. Sacristán, and F. García-Arenal, "Coexistence of nestedness and modularity in host–pathogen infection networks," *Nature Ecology & Evolution,* pp. 1-10, 03/09 2020, doi: 10.1038/s41559-020-1130-9.

[42] T. Shiino, "Phylodynamic analysis of a viral infection network," *Frontiers in microbiology,* vol. 3, p. 278, 07/31 2012, doi: 10.3389/fmicb.2012.00278.

[43] Q. Fan, J. Yang, D. Wipf, B. Chen, and X. Tong, "Image smoothing via unsupervised learning," vol. 37, no. 6 %J ACM Trans. Graph., p. Article 259, 2018, doi: 10.1145/3272127.3275081.

[44] Y. Zheng, X. Li, and X. Lu, "Unsupervised Learning of Human Action Categories in Still Images with Deep Representations," vol. 15, no. 4 %J ACM Trans. Multimedia Comput. Commun. Appl., p. Article 112, 2019, doi: 10.1145/3362161.

[45] A. Hart, "Using Neural Networks for Classification Tasks – Some Experiments on Datasets and Practical Advice," *Journal of the Operational Research Society,* vol. 43, no. 3, pp. 215-226, 1992/03/01 1992, doi: 10.1057/jors.1992.31.

[46] R. Marxer and H. Purwins, "Unsupervised incremental online learning and prediction of musical audio signals," vol. 24, no. 5 %J IEEE/ACM Trans. Audio, Speech and Lang. Proc., pp. 863–874, 2016, doi: 10.1109/taslp.2016.2530409.

[47] F. Saki, N. Kehtarnavaz, F. Saki, and N. Kehtarnavaz, "Real-Time Unsupervised Classification of Environmental Noise Signals," vol. 25, no. 8 %J IEEE/ACM Trans. Audio, Speech and Lang. Proc., pp. 1657–1667, 2017, doi: 10.1109/taslp.2017.2711059.

[48] J. E. Vogt, "Unsupervised structure detection in biomedical data," vol. 12, no. 4 %J IEEE/ACM Trans. Comput. Biol. Bioinformatics, pp. 753–760, 2015, doi: 10.1109/tcbb.2015.2394408.

[49] J. G. Dy and C. E. Brodley, "Feature Selection for Unsupervised Learning," vol. 5, pp. 845–889, 2004.

[50] H. Hammarström and L. Borin, "Unsupervised learning of morphology," vol. 37, no. 2 %J Comput. Linguist., pp. 309–350, 2011, doi: 10.1162/COLI_a_00050.

[51] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," vol. 4, no. 1 %J ACM Trans. Speech Lang. Process., p. Article 3, 2007, doi: 10.1145/1187415.1187418.

[52] S. Küçükpetek, F. Polat, and H. Oğuztüzün, "Multilevel graph partitioning: an evolutionary approach," *Journal of the Operational Research Society,* vol. 56, no. 5, pp. 549-562, 2005/05/01 2005, doi: 10.1057/palgrave.jors.2601837.

[53] M. E. J. Newman, "Analysis of weighted networks," *Physical Review E,* vol. 70, no. 5, p. 056131, 11/24/ 2004, doi: 10.1103/PhysRevE.70.056131.

[54] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," vol. 2008, p. 10008, October 01, 2008 2008, doi: 10.1088/1742-5468/2008/10/p10008.

[55] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks," in *International AAAI Conference on Weblogs and Social Media*, 2009.

[56] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. J. J. o. s. m. t. Lefebvre, and experiment, "Fast unfolding of communities in large networks," vol. 2008, no. 10, p. P10008, 2008.

[57] R. Lambiotte, J.-C. Delvenne, and M. J. a. p. a. Barahona, "Laplacian dynamics and multiscale modular structure in networks," 2008.

APPENDİX

The extracted solutions are listed by the clustered graph for the resolutions, namely 0.2, 0.3, 0.4, and 0.5.

For resolution 0.2.

3,8,9,10,12,27,28,33,34
3,8,9,10,12,28,31,33,34,35
3,9,10,12,19,28,31,33,34,35
7,8,9,10,11,12,27,28,33,34
2,3,6,8,9,12,18,27,28,33,34
2,3,6,8,9,12,18,28,31,33,34,35
2,3,6,9,12,18,19,28,31,33,34,35
2,6,7,8,9,11,12,18,27,28,33,34
4,7,9,10,24,27,32,33,34,36
3,4,9,10,27,28,33,34
3,4,9,10,24,27,32,33,34,36
3,4,9,10,31,32,33,34,35,36
3,4,9,10,24,31,32,33,34,36
3,4,9,10,19,28,31,33,34,35
3,4,9,10,19,31,33,34,35,36
3,4,9,10,19,24,31,33,34,36
4,7,9,10,11,27,28,33,34
4,7,9,10,11,27,32,33,34,36
9,10,15,16,30,31,32,33,35,36
9,10,15,16,31,32,33,34,35,36
1,3,6,8,12,13,18,27,28,30,33
1,3,6,8,12,13,18,27,28,33,34
1,3,6,8,12,13,18,28,30,31,33,35
1,3,6,8,12,13,18,28,31,33,34,35
1,3,6,8,13,18,23,28,30,31,33,35
1,3,6,12,13,18,19,28,30,31,33,35
1,3,6,12,13,18,19,28,31,33,34,35
1,3,13,17,18,19,23,31,33,35
1,3,13,17,18,19,23,24,31,33
1,3,5,8,10,12,13,27,28,30,33
1,3,5,8,10,12,13,27,28,33,34
1,3,5,6,8,12,13,27,28,30,33
1,3,5,6,8,12,13,27,28,33,34
1,3,5,6,12,13,19,28,30,31,33,35
1,3,5,6,12,13,19,28,31,33,34,35
1,3,5,10,12,13,19,28,30,31,33,35
1,3,5,10,12,13,19,28,31,33,34,35
1,3,5,13,17,19,23,31,33,35
1,3,5,13,17,19,23,24,31,33
1,3,5,6,8,12,13,28,30,31,33,35
1,3,5,6,8,12,13,28,31,33,34,35
1,3,5,6,8,13,23,28,30,31,33,35
1,3,5,8,10,12,13,28,30,31,33,35
1,3,5,8,10,12,13,28,31,33,34,35
1,6,7,8,11,12,13,18,27,28,30,33
1,6,7,8,11,12,13,18,27,28,33,34
1,5,6,7,8,11,12,13,27,28,30,33
1,5,6,7,8,11,12,13,27,28,33,34
1,5,7,8,10,11,12,13,27,28,30,33
1,5,7,8,10,11,12,13,27,28,33,34
1,2,3,6,8,12,18,27,28,33,34
1,2,3,6,8,12,18,28,31,33,34,35
1,2,3,6,12,18,19,28,31,33,34,35
1,2,3,17,18,19,23,26,31,33,35
1,2,3,17,18,19,23,24,26,31,33
1,2,3,5,6,8,12,27,28,33,34
1,2,3,5,6,8,12,28,31,33,34,35
1,2,3,5,6,12,19,28,31,33,34,35
1,2,3,5,17,19,23,26,31,33,35
1,2,3,5,17,19,23,24,26,31,33
1,2,6,7,8,11,12,18,27,28,33,34
1,2,5,6,7,8,11,12,27,28,33,34
1,4,6,7,13,18,24,27,30,32,33,36
1,4,6,7,13,18,24,27,32,33,34,36
1,4,5,6,7,13,24,27,30,32,33,36
1,4,5,6,7,13,24,27,32,33,34,36
1,4,5,7,10,13,24,27,30,32,33,36
1,4,5,7,10,13,24,27,32,33,34,36
1,3,4,6,13,18,27,28,30,33
1,3,4,6,13,18,27,28,33,34
1,3,4,6,13,18,24,27,30,32,33,36
1,3,4,6,13,18,24,27,32,33,34,36
1,3,4,6,13,18,30,31,32,33,35,36
1,3,4,6,13,18,31,32,33,34,35,36
1,3,4,6,13,18,24,30,31,32,33,36
1,3,4,6,13,18,24,31,32,33,34,36
1,3,4,5,6,13,27,28,30,33
1,3,4,5,6,13,27,28,33,34
1,3,4,5,10,13,27,28,30,33
1,3,4,5,10,13,27,28,33,34
1,3,4,6,13,18,19,28,31,33,34,35
1,3,4,6,13,18,19,23,24,30,31,33
1,3,4,6,13,18,19,23,28,30,31,33,35
1,3,4,6,13,18,19,30,31,33,35,36
1,3,4,6,13,18,19,31,33,34,35,36
1,3,4,6,13,18,19,24,30,31,33,36
1,3,4,6,13,18,19,24,31,33,34,36
1,3,4,6,13,18,23,30,31,32,33,35
1,3,4,6,13,18,23,24,30,31,32,33
1,3,4,5,10,13,24,27,30,32,33,36
1,3,4,5,10,13,24,27,32,33,34,36
1,3,4,5,6,13,24,27,30,32,33,36
1,3,4,5,6,13,24,27,32,33,34,36
1,3,4,5,6,13,19,28,31,33,34,35
1,3,4,5,6,13,19,23,24,30,31,33
1,3,4,5,6,13,19,23,28,30,31,33,35
1,3,4,5,6,13,19,30,31,33,35,36
1,3,4,5,6,13,19,31,33,34,35,36
1,3,4,5,6,13,19,24,30,31,33,36
1,3,4,5,6,13,19,24,31,33,34,36
1,3,4,5,10,13,19,28,30,31,33,35
1,3,4,5,10,13,19,28,31,33,34,35
1,3,4,5,10,13,19,30,31,33,35,36
1,3,4,5,10,13,19,31,33,34,35,36
1,3,4,5,10,13,19,24,30,31,33,36
1,3,4,5,10,13,19,24,31,33,34,36
1,3,4,5,6,13,23,30,31,32,33,35
1,3,4,5,6,13,23,24,30,31,32,33
1,3,4,5,6,13,30,31,32,33,35,36
1,3,4,5,6,13,31,32,33,34,35,36
1,3,4,5,6,13,24,30,31,32,33,36
1,3,4,5,6,13,24,31,32,33,34,36
1,3,4,5,10,13,30,31,32,33,35,36
1,3,4,5,10,13,31,32,33,34,35,36
1,3,4,5,10,13,24,30,31,32,33,36
1,3,4,5,10,13,24,31,32,33,34,36
1,4,6,7,11,13,18,27,28,30,33
1,4,6,7,11,13,18,27,28,33,34
1,4,6,7,11,13,18,27,30,32,33,36
1,4,6,7,11,13,18,27,32,33,34,36
1,4,5,6,7,11,13,27,28,30,33
1,4,5,6,7,11,13,27,28,33,34
1,4,5,7,10,11,13,27,28,30,33
1,4,5,7,10,11,13,27,28,33,34
1,4,5,6,7,11,13,27,30,32,33,36
1,4,5,6,7,11,13,27,32,33,34,36
1,4,5,7,10,11,13,27,30,32,33,36
1,4,5,7,10,11,13,27,32,33,34,36
1,6,13,15,16,18,30,31,32,33,35,36
1,6,13,15,16,18,31,32,33,34,35,36
1,10,13,15,16,30,31,32,33,35,36
1,13,15,16,17,18,31,32,33,34,35,36
13,14,17,18,19,20,23,24
5,13,14,17,19,20,23,24,29
11,13,14,17,18,19,20,23,35
5,11,13,14,17,19,20,23,29,35
2,14,17,18,19,20,23,24,26
2,5,14,17,19,20,23,24,26,29
2,11,14,17,18,19,20,23,26,35
2,5,11,14,17,19,20,23,26,29,35
4,6,13,14,18,19,20,23,24,30
4,6,13,14,18,19,20,24,30,36
4,6,13,14,18,19,20,24,34,36
4,6,13,14,18,20,23,24,30,32
4,6,13,14,18,20,24,30,32,36
4,6,13,14,18,20,24,32,34,36
4,5,6,13,14,20,23,24,29,30,32
4,5,6,13,14,20,24,29,30,32,36
4,5,6,13,14,20,24,29,32,34,36
4,5,6,13,14,19,20,23,24,29,30
4,5,6,13,14,19,20,24,29,30,36
4,5,6,13,14,19,20,24,29,34,36
4,6,11,13,14,18,19,20,23,30,35
4,6,11,13,14,18,19,20,30,35,36
4,6,11,13,14,18,19,20,34,35,36
4,6,11,13,14,18,20,23,30,32,35
4,6,11,13,14,18,20,30,32,35,36
4,6,11,13,14,18,20,32,34,35,36
4,5,6,11,13,14,19,20,23,29,30,35
4,5,6,11,13,14,19,20,29,30,35,36
4,5,6,11,13,14,19,20,29,34,35,36
4,5,6,11,13,14,20,23,29,30,32,35
4,5,6,11,13,14,20,29,30,32,35,36
4,5,6,11,13,14,20,29,32,34,35,36
6,9,14,15,16,18,30,31,32,33,35,36
6,9,14,15,16,18,31,32,33,34,35,36
6,13,14,15,16,18,30,31,32,33,35,36
6,13,14,15,16,18,31,32,33,34,35,36
13,14,15,16,17,18,31,32,33,34,35,36
3,8,9,10,12,25,27,28,29,34
3,8,9,10,12,25,28,29,31,34,35
3,9,10,12,19,25,28,29,31,34,35
7,8,9,10,11,12,25,27,28,29,34
2,3,6,8,9,12,25,27,28,29,34
2,3,6,8,9,12,18,25,27,28,34
2,3,6,8,9,12,25,28,29,31,34,35
2,3,6,8,9,12,18,25,28,31,34,35
2,3,6,9,12,19,25,28,29,31,34,35
2,3,6,9,12,18,19,25,28,31,34,35
2,6,7,8,9,11,12,25,27,28,29,34
2,6,7,8,9,11,12,18,25,27,28,34
4,7,9,10,24,25,27,29,32,34,36
3,4,9,10,24,25,27,29,32,34,36
3,4,9,10,25,27,28,29,34
3,4,9,10,19,25,28,29,31,34,35
3,4,9,10,19,25,29,31,34,35,36
3,4,9,10,19,24,25,29,31,34,36
3,4,9,10,25,29,31,32,34,35,36
3,4,9,10,24,25,29,31,32,34,36
4,7,9,10,11,25,27,28,29,34
4,7,9,10,11,25,27,29,32,34,36
6,9,15,16,18,25,30,31,32,35,36
6,9,15,16,18,25,31,32,34,35,36
9,10,15,16,25,30,31,32,35,36
9,10,15,16,25,31,32,34,35,36

For the resolution 0.3

2,4,5,6,7,14,24,29,31,32,34,36
2,4,5,6,7,14,24,31,32,33,34,36
2,4,6,14,18,20,24,32,34,36
2,4,5,6,14,20,24,29,32,34,36
4,5,6,7,13,14,23,24,29,30,31,32
4,5,6,7,13,14,23,24,30,31,32,33
4,5,6,7,13,14,24,29,30,31,32,36
4,5,6,7,13,14,24,29,31,32,34,36
4,5,6,7,13,14,24,30,31,32,33,36
4,5,6,7,13,14,24,31,32,33,34,36
4,6,13,14,18,20,23,24,30,32
4,5,6,13,14,20,23,24,29,30,32
4,6,13,14,18,20,24,30,32,36
4,6,13,14,18,20,24,32,34,36
4,5,6,13,14,20,24,29,30,32,36
4,5,6,13,14,20,24,29,32,34,36
5,6,7,8,13,14,23,24,29,30,31,32
5,6,7,8,13,14,23,24,30,31,32,33
6,8,13,14,18,20,23,24,30,32
5,6,8,13,14,20,23,24,29,30,32
1,6,15,16,21,22,31,36
1,2,4,5,6,7,11,27,32,33,34,36
1,2,4,5,6,7,24,27,32,33,34,36
1,2,4,5,6,7,24,31,32,33,34,36
1,2,3,5,6,12,21,22,27,28
1,2,3,4,5,6,24,27,32,33,34,36
1,2,3,4,5,6,24,31,32,33,34,36
1,2,3,4,5,6,31,32,33,34,35,36
1,2,3,4,5,6,27,28,33,34
1,2,3,4,5,6,21,22,27,28
1,2,3,4,6,18,25,27,28,34
1,2,3,4,5,6,22,25,27,28,29,34
1,2,3,5,6,8,12,27,28,33,34
1,2,3,5,6,8,12,28,31,33,34,35
1,2,3,6,8,12,18,25,27,28,34
1,2,3,5,6,8,12,22,25,27,28,29,34
1,2,3,5,6,8,12,22,25,28,29,31,34
1,2,3,6,8,12,18,25,28,31,34,35
1,2,3,5,6,8,12,25,28,29,31,34,35
1,2,3,5,6,8,22,23,25,28,29,31
1,2,3,5,6,12,19,28,31,33,34,35
1,2,3,5,6,12,19,21,22,28,31
1,2,3,5,6,12,19,22,25,28,29,31,34
1,2,3,6,12,18,19,25,28,31,34,35
1,2,3,5,6,12,19,25,28,29,31,34,35
1,4,5,7,10,11,13,27,30,32,33,36
1,4,5,7,10,11,13,27,32,33,34,36
1,4,5,7,10,13,24,27,30,32,33,36
1,4,5,7,10,13,24,27,32,33,34,36
1,4,5,7,10,13,24,30,31,32,33,36
1,4,5,7,10,13,24,31,32,33,34,36
1,4,5,6,7,13,23,24,30,31,32,33
1,4,5,6,7,11,13,27,30,32,33,36
1,4,5,6,7,11,13,27,32,33,34,36
1,4,5,6,7,13,24,27,30,32,33,36
1,4,5,6,7,13,24,27,32,33,34,36
1,4,5,6,7,13,24,30,31,32,33,36
1,4,5,6,7,13,24,31,32,33,34,36
1,5,6,7,8,13,23,24,30,31,32,33
1,10,13,15,16,30,31,32,33,35,36
1,3,4,5,6,13,23,24,30,31,32,33
1,3,4,5,6,13,23,30,31,32,33,35
1,3,4,5,6,13,24,27,30,32,33,36
1,3,4,5,6,13,24,27,32,33,34,36
1,3,4,5,6,13,24,30,31,32,33,36
1,3,4,5,6,13,24,31,32,33,34,36
1,3,4,5,6,13,30,31,32,33,35,36
1,3,4,5,6,13,31,32,33,34,35,36
1,3,4,5,10,13,27,28,30,33
1,3,4,5,10,13,27,28,33,34
1,3,4,5,10,13,25,27,28,29,30
1,3,4,5,10,13,22,25,27,28,29,34
1,3,4,5,6,13,27,28,30,33
1,3,4,5,6,13,27,28,33,34
1,3,4,6,13,18,25,27,28,30
1,3,4,6,13,18,25,27,28,34
1,3,4,5,6,13,25,27,28,29,30
1,3,4,5,6,13,22,25,27,28,29,34
1,3,4,5,10,13,24,27,30,32,33,36
1,3,4,5,10,13,24,27,32,33,34,36
1,3,4,5,10,13,24,30,31,32,33,36
1,3,4,5,10,13,24,31,32,33,34,36
1,3,4,5,10,13,30,31,32,33,35,36
1,3,4,5,10,13,31,32,33,34,35,36
1,3,5,6,8,13,23,24,30,31,32,33
1,3,5,6,8,13,23,30,31,32,33,35
1,3,5,8,10,12,13,27,28,30,33
1,3,5,8,10,12,13,27,28,33,34
1,3,5,8,10,12,13,28,30,31,33,35
1,3,5,8,10,12,13,28,31,33,34,35
1,3,5,8,10,12,13,25,27,28,29,30
1,3,5,8,10,12,13,22,25,27,28,29,34
1,3,5,8,10,12,13,22,25,28,29,31,34
1,3,5,8,10,12,13,25,28,29,30,31,35
1,3,5,8,10,12,13,25,28,29,31,34,35
1,3,5,6,8,12,13,27,28,30,33
1,3,5,6,8,12,13,27,28,33,34
1,3,5,6,8,12,13,28,30,31,33,35
1,3,5,6,8,12,13,28,31,33,34,35
1,3,5,6,8,13,23,28,30,31,33,35
1,3,6,8,12,13,18,25,27,28,30
1,3,6,8,12,13,18,25,27,28,34
1,3,5,6,8,12,13,25,27,28,29,30
1,3,5,6,8,12,13,22,25,27,28,29,34
1,3,5,6,8,12,13,22,25,28,29,31,34
1,3,6,8,12,13,18,25,28,30,31,35
1,3,6,8,12,13,18,25,28,31,34,35
1,3,5,6,8,12,13,25,28,29,30,31,35
1,3,5,6,8,12,13,25,28,29,31,34,35
1,3,5,6,8,13,22,23,25,28,29,31
1,3,6,8,13,18,23,25,28,30,31,35
1,3,5,6,8,13,23,25,28,29,30,31,35
1,3,5,10,12,13,19,28,30,31,33,35
1,3,5,10,12,13,19,28,31,33,34,35
1,3,5,10,12,13,19,22,25,28,29,31,34
1,3,5,10,12,13,19,25,28,29,30,31,35
1,3,5,10,12,13,19,25,28,29,31,34,35
1,3,5,6,12,13,19,28,30,31,33,35
1,3,5,6,12,13,19,28,31,33,34,35
1,3,5,6,12,13,19,22,25,28,29,31,34
1,3,6,12,13,18,19,25,28,30,31,35
1,3,6,12,13,18,19,25,28,31,34,35
1,3,5,6,12,13,19,25,28,29,30,31,35
1,3,5,6,12,13,19,25,28,29,31,34,35
1,4,5,7,10,11,26,27,30,32,33,36
1,4,5,7,10,24,26,27,30,32,33,36
1,4,5,7,10,24,26,30,31,32,33,36
1,3,4,5,10,26,27,28,30,33
1,3,4,5,10,25,26,27,28,29,30
1,3,4,5,10,24,26,27,30,32,33,36
1,3,4,5,10,24,26,30,31,32,33,36
1,3,4,5,10,26,30,31,32,33,35,36
1,3,5,8,10,12,26,27,28,30,33
1,3,5,8,10,12,26,28,30,31,33,35
1,3,5,8,10,12,25,26,27,28,29,30
1,3,5,8,10,12,25,26,28,29,30,31,35
1,3,5,10,12,19,26,28,30,31,33,35
1,3,5,10,12,19,25,26,28,29,30,31,35
4,7,9,10,11,27,32,33,34,36
4,7,9,10,24,27,32,33,34,36
4,7,9,10,24,31,32,33,34,36
9,10,15,16,30,31,32,33,35,36
9,10,15,16,31,32,33,34,35,36
3,4,9,10,24,27,32,33,34,36
3,4,9,10,24,31,32,33,34,36
3,4,9,10,31,32,33,34,35,36
3,4,9,10,27,28,33,34
3,4,9,10,25,27,28,29,34
3,8,9,10,12,27,28,33,34
3,8,9,10,12,28,31,33,34,35
3,8,9,10,12,25,27,28,29,34
3,8,9,10,12,25,28,29,31,34,35
3,9,10,12,19,28,31,33,34,35
3,9,10,12,19,25,28,29,31,34,35
2,3,4,6,9,18,25,27,28,34
2,3,4,6,9,25,27,28,29,34
2,3,6,8,9,12,18,25,27,28,34
2,3,6,8,9,12,25,27,28,29,34
2,3,6,8,9,12,18,25,28,31,34,35
2,3,6,8,9,12,25,28,29,31,34,35
2,3,6,9,12,18,19,25,28,31,34,35
2,3,6,9,12,19,25,28,29,31,34,35
2,4,6,7,9,14,24,29,31,32,34,36
4,7,9,10,11,26,27,30,32,33,36
4,7,9,10,24,26,27,30,32,33,36
4,7,9,10,24,26,30,31,32,33,36
3,4,9,10,24,26,27,30,32,33,36
3,4,9,10,24,26,30,31,32,33,36
3,4,9,10,26,30,31,32,33,35,36
3,4,9,10,26,27,28,30,33
3,4,9,10,25,26,27,28,29,30
3,8,9,10,12,26,27,28,30,33
3,8,9,10,12,26,28,30,31,33,35
3,8,9,10,12,25,26,27,28,29,30
3,8,9,10,12,25,26,28,29,30,31,35
3,9,10,12,19,25,26,28,29,30,31,35
3,9,10,12,19,26,28,30,31,33,35
5,7,8,13,14,17,23,24,29,31,32
5,7,8,13,14,17,23,24,31,32,33
8,13,14,17,18,20,23,24,32
5,8,13,14,17,20,23,24,29,32
5,7,13,14,17,19,23,24,29,31
5,7,13,14,17,19,23,24,31,33
13,14,17,18,19,20,23,24
5,13,14,17,19,20,23,24,29

For the resolution 0.4

13,14,17,18,19,20,23,24
4,6,13,14,18,20,23,24,30,32
4,6,13,14,18,20,24,30,32,36
4,6,13,14,18,19,20,23,24,30
4,6,13,14,18,19,20,24,30,36
8,13,14,17,18,20,23,24,32
6,8,13,14,18,20,23,24,30,32
6,12,13,14,18,19,20,24,30,36
2,14,17,18,19,20,23,24,26
2,4,6,14,18,20,24,32,34,36
2,4,6,14,18,20,23,24,26,30,32
2,4,6,14,18,20,24,26,30,32,36
2,4,6,14,18,19,20,24,34,36
2,8,14,17,18,20,23,24,26,32
2,6,8,14,18,20,23,24,26,30,32
2,12,14,17,18,19,20,24,26,36
2,12,14,17,18,19,20,24,34,36
2,6,12,14,18,19,20,24,34,36
6,12,14,15,18,20,21,24,30,36
6,12,14,15,18,20,21,30,35,36
1,6,15,16,21,22,31,36
1,10,13,15,16,25,30,31,32,35,36
1,10,13,15,16,30,31,32,33,35,36
1,3,6,12,15,18,20,21,24,27,30,36
1,3,6,12,15,18,20,21,30,35,36
5,13,14,17,19,20,23,24,29
4,5,6,13,14,20,23,24,29,30,32
4,5,6,13,14,20,24,29,30,32,36
4,5,6,13,14,19,20,23,24,29,30
4,5,6,13,14,19,20,24,29,30,36
5,6,8,13,14,20,23,24,29,30,32
5,8,13,14,17,20,23,24,29,32
5,6,12,13,14,19,20,24,29,30,36
2,5,14,17,19,20,23,24,26,29
2,4,5,6,14,20,24,29,32,34,36
2,4,5,6,14,20,23,24,26,29,30,32
2,4,5,6,14,20,24,26,29,30,32,36
2,4,5,6,14,19,20,24,29,34,36
2,4,5,6,14,19,20,23,24,26,29,30
2,4,5,6,14,19,20,24,26,29,30,36
2,5,8,14,17,20,23,24,26,29,32
2,5,6,8,14,20,23,24,26,29,30,32
2,5,12,14,17,19,20,24,26,29,36
2,5,12,14,17,19,20,24,29,34,36
2,5,6,12,14,19,20,24,29,34,36
2,5,6,12,14,19,20,24,26,29,30,36
5,6,12,14,15,21,24,30,31,33,36
5,6,12,14,15,21,30,31,33,35,36
5,6,12,14,15,20,21,24,30,36
5,6,12,14,15,20,21,30,35,36
4,7,9,10,11,27,28,33,34
4,7,9,10,11,27,32,33,34,36
4,7,9,10,11,26,27,28,30,33
4,7,9,10,11,26,27,30,32,33,36
4,7,9,10,24,27,32,33,34,36
4,7,9,10,24,26,27,30,32,33,36
9,10,15,16,25,31,32,34,35,36
9,10,15,16,31,32,33,34,35,36
9,10,15,16,25,30,31,32,35,36
9,10,15,16,30,31,32,33,35,36
3,4,9,10,27,28,33,34
3,4,9,10,26,27,28,30,33
3,4,9,10,24,27,32,33,34,36
3,4,9,10,24,26,27,30,32,33,36
3,4,9,10,24,31,32,33,34,36
3,4,9,10,24,26,30,31,32,33,36
3,4,9,10,19,24,31,33,34,36
3,4,9,10,19,24,26,30,31,33,36
3,4,9,10,31,32,33,34,35,36
3,4,9,10,26,30,31,32,33,35,36
3,4,9,10,19,31,33,34,35,36
3,4,9,10,19,28,31,33,34,35
3,4,9,10,19,26,30,31,33,35,36
3,4,9,10,19,26,28,30,31,33,35
3,9,10,12,19,24,31,33,34,36
3,9,10,12,19,24,26,30,31,33,36
3,9,10,12,19,31,33,34,35,36
3,9,10,12,19,28,31,33,34,35
3,9,10,12,19,26,30,31,33,35,36
3,9,10,12,19,26,28,30,31,33,35

For the resolution 0.5

13,14,17,18,19,20,23,24
3,4,9,10,27,28,33,34
1,6,15,16,21,22,31,36
8,13,14,17,18,20,23,24,32
4,7,9,10,11,27,28,33,34
3,4,9,10,26,27,28,30,33
3,4,9,10,25,27,28,29,34
12,13,14,17,18,19,20,24,34,36
2,12,14,17,18,19,20,24,34,36
2,12,14,17,18,19,20,24,26,36
4,6,13,14,18,19,20,24,34,36
6,12,13,14,18,19,20,24,34,36
6,8,13,14,18,20,23,24,30,32
4,6,13,14,18,20,23,24,30,32
4,6,13,14,18,19,20,23,24,30
4,6,13,14,18,19,20,24,30,36
6,12,14,15,18,20,21,24,30,36
6,12,13,14,18,19,20,24,30,36
6,12,14,15,18,20,21,30,35,36
5,6,12,14,15,20,21,30,35,36
5,6,12,14,15,20,21,24,30,36
3,4,9,10,19,24,31,33,34,36
3,9,10,12,19,24,31,33,34,36
3,4,9,10,19,28,31,33,34,35
3,9,10,12,19,28,31,33,34,35
3,9,10,12,19,31,33,34,35,36
3,4,9,10,19,31,33,34,35,36
4,7,9,10,11,26,27,28,30,33
3,4,9,10,25,26,27,28,29,30
4,7,9,10,11,25,27,28,29,34
3,4,9,10,19,24,26,30,31,33,36
3,4,9,10,19,26,30,31,33,35,36
3,4,9,10,19,26,28,30,31,33,35
3,9,10,12,19,24,26,30,31,33,36
3,9,10,12,19,26,30,31,33,35,36
3,9,10,12,19,26,28,30,31,33,35
3,4,9,10,19,24,25,29,31,34,36
3,4,9,10,19,25,29,31,34,35,36
3,4,9,10,19,25,28,29,31,34,35
4,7,9,10,11,25,26,27,28,29,30
3,9,10,12,19,24,25,29,31,34,36
3,9,10,12,19,25,29,31,34,35,36
3,9,10,12,19,25,28,29,31,34,35
1,3,6,12,15,18,20,21,30,35,36
1,3,5,6,12,15,20,21,30,35,36
3,4,9,10,19,24,25,26,29,30,31,36
3,4,9,10,19,25,26,29,30,31,35,36
3,4,9,10,19,25,26,28,29,30,31,35
3,9,10,12,19,24,25,26,29,30,31,36
3,9,10,12,19,25,26,29,30,31,35,36
3,9,10,12,19,25,26,28,29,30,31,35
1,2,3,12,17,18,19,20,25,34,35,36
1,2,3,12,17,18,19,20,24,25,34,36
1,2,3,12,17,18,19,25,31,34,35,36
1,2,3,12,17,18,19,24,25,31,34,36
1,2,3,12,17,18,19,20,25,26,35,36
1,2,3,12,17,18,19,20,24,25,26,36
1,2,3,12,17,18,19,25,26,31,35,36
1,2,3,12,17,18,19,24,25,26,31,36
1,3,6,12,15,18,20,21,24,27,30,36
1,3,5,6,12,15,20,21,24,27,30,36