



**FATİH
SULTAN
MEHMET**
VAKIF ÜNİVERSİTESİ

Doç.Dr. ÖMER KORÇAK

ABDELRAHMAN TAHA ABDELTAWAB ABDELLATIF

BLM511 - Algoritma Analizi ve Tasarımında İleri Teknikler

Graph Coloring Problem

Homework 4 solutions

Graph Coloring Problem

Graph Coloring (or Vertex Coloring) problem is defined as assignment of smallest number of colors to the vertices of an undirected graph, such that no adjacent vertices are of the same color. A coloring using at most k colors is called a (proper) k -coloring. The smallest number of colors needed to color a graph G is called its chromatic number and is often denoted $\chi(G)$. The goal is to color all vertices of the graph, such that number of colors used (k) is as close as possible to the optimal result, i.e. $\chi(G)$. Since this problem is NP-Hard, it is very difficult to find the optimal solution especially for large instances.

Project Specification:

You are asked to design and implement an algorithm for Graph Coloring problem described above. Your goal is not to design an algorithm for the optimal solution, but you are requested to do your best. This is an open-ended assignment.

Input format: Inputs will always be given as a text file. Input file format should be as follows: The first line is the problem line which indicates the number of vertices and the number of edges. It starts with "p", and there is only one such line.

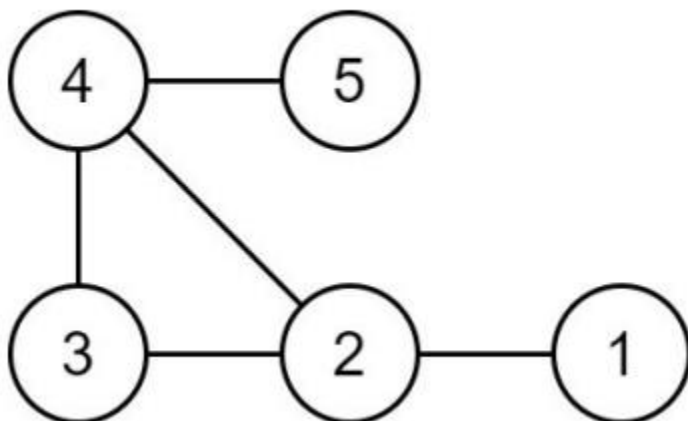
p <NumVertices> <NumEdges>

The rest of the lines are edge lines. Each line starts with "e" and indicates an edge between two vertices.

e <VertexNumber1> <VertexNumber2>

The above line means that there is an edge between VertexNumber1 and VertexNumber2.

Here is a sample graph:



Algorithms Step

Assign colors to all vertices such that no adjacent vertices have the same color and print the assigned colors.

- 1) First add an edge and form a graph by adding different vertices.
- 2) Assign the first color to the first vertex
- 3) A temporary array to store the available colors.
- 4) True value of available[cr] would mean that the
- 5) Color cr is assigned to one of its adjacent vertices
- 6) Assign colors to the remaining V-1 vertices
- 7) Process all adjacent vertices and flag their colors as unavailable
- 8) Find the first available color
- 9) Assign the found color
- 10) Reset the values back to false for the next iteration
- 11) Print the result

We run the code by first ad

- 1) First add an edge and form a graph by adding different vertices.
 - 2) Assign the first color to the first vertex
 - 3) A temporary array to store the available colors.
 - 4) True value of available[cr] would mean that the
 - 5) Color cr is assigned to one of its adjacent vertices
- values back to false for the next iteration

Outputs

Sample1.txt (87 vertices, 812 edges) is

```
12
000100100010020100234210001020153631101303403710441311311523
10000845601669054151501075111
```

Sample2.txt (1000 vertices, 245000 edges) is

126

```
0 1 2 1 0 2 3 4 1 5 0 2 3 4 5 0 1 4 2 6 3 6 5 0 3 7 2 7 8 8 9 5 4 9 6 8 1 4 9 6 10 6 1 3 10 7 10 9 5 11 10 12 13 5 14
11 0 8 0 13 14 11 4 0 10 9 8 15 12 16 7 12 15 12 14 14 16 17 16 18 16 17 11 19 3 11 13 10 20 11 20 7 11 19 2
0 17 21 18 15 13 14 21 13 3 18 22 8 10 15 20 9 17 17 23 21 19 22 11 23 24 18 25 23 23 6 19 24 12 22 25 25
24 24 19 8 12 26 7 27 5 2 16 28 12 18 26 21 4 1 29 15 6 26 28 30 28 13 26 29 16 21 31 27 31 30 25 22 8 32 2
3 32 32 19 30 28 14 24 17 22 33 29 20 27 25 30 27 14 34 30 8 33 27 2 31 27 31 35 29 29 24 34 31 36 32 35 2
4 36 26 18 33 7 6 34 29 2 36 28 37 38 33 14 39 19 37 22 31 34 17 38 38 13 35 21 18 25 23 37 35 32 1 33 39
18 3 36 15 14 30 33 34 39 40 32 33 36 32 40 28 31 41 42 30 40 26 43 28 38 23 41 43 44 20 38 29 41 42 41 2
2 24 31 45 16 43 5 42 37 26 39 0 35 45 42 46 29 12 43 42 44 44 46 46 45 45 47 47 47 48 2 48 27 4 32 44 49
16 45 34 41 49 47 50 10 51 44 48 9 44 1 40 43 14 48 9 25 20 51 49 40 50 51 38 11 2 48 39 51 52 19 6 47 52
21 53 0 54 49 40 13 45 53 52 49 49 3 43 46 50 13 32 54 55 37 56 47 33 50 35 54 19 52 36 53 55 1 51 57 58 1
1 6 46 42 56 55 37 57 56 53 53 48 50 58 22 54 55 23 57 57 39 50 43 46 59 51 58 59 55 60 61 62 39 37 52 59
60 58 60 50 12 39 56 57 20 56 41 41 52 61 26 58 15 62 63 64 31 62 63 61 64 65 65 63 20 58 54 36 64 66 47
63 60 52 55 23 66 60 59 53 29 67 64 62 40 67 68 51 56 66 44 65 5 69 7 65 44 30 54 50 68 17 66 70 57 61 59
58 21 59 69 70 61 34 67 69 68 65 68 71 35 61 48 68 70 59 62 48 66 49 18 53 60 66 50 71 10 64 72 65 71 67
73 49 35 74 65 69 55 74 36 61 54 72 73 70 46 75 66 63 72 62 48 42 73 69 74 76 76 72 75 67 72 22 16 74 67
73 77 78 77 57 60 56 43 79 78 75 37 24 75 75 45 70 77 79 69 18 80 76 80 81 82 76 75 79 78 76 80 7 71 81 8
2 73 59 72 71 23 76 56 82 67 80 83 77 60 80 34 76 79 84 68 84 79 47 77 74 28 83 85 62 84 68 81 79 63 67 7
82 84 68 38 86 65 83 85 84 55 64 81 82 87 78 80 25 86 81 79 86 85 82 83 74 71 88 64 70 87 78 88 63 89 77
31 74 89 71 58 84 83 83 90 89 85 88 87 86 87 75 88 78 80 70 61 90 89 91 84 89 91 90 86 92 75 48 73 70 92
85 87 88 91 93 86 90 94 94 81 92 93 93 94 54 85 39 60 95 42 45 77 66 92 85 96 12 87 66 82 92 93 57 83 94
95 56 64 92 93 97 55 96 88 98 94 72 97 25 99 97 73 38 97 96 38 98 43 95 100 95 88 98 91 70 33 47 99 49 9
7 65 101 98 102 100 92 97 103 101 53 52 59 100 102 83 93 96 43 91 104 40 99 90 102 104 103 100 36 101
103 78 103 105 90 79 106 107 98 69 102 46 96 92 78 84 104 107 100 105 108 66 32 90 46 105 101 87 99 1
04 57 87 102 62 63 27 88 105 106 89 13 106 87 101 107 100 105 35 109 108 104 99 95 73 109 99 110 40 9
7 110 91 92 82 41 108 107 108 106 111 30 108 94 3 71 109 112 26 109 41 45 86 103 69 82 14 113 106 113
111 114 110 108 113 111 112 110 21 112 111 113 114 111 109 94 115 102 15 89 15 101 89 114 115 116 11
6 35 117 34 86 51 116 110 37 111 102 96 99 83 116 114 80 107 52 30 98 117 95 118 114 117 118 91 113 2
2 119 115 67 116 120 118 29 115 91 119 81 98 112 112 115 105 3 119 115 116 117 118 121 120 121 110 1
20 93 121 103 71 122 9 123 104 122 107 124 111 11 116 122 123 65 124 37 118 119 120 77 125 94
```

Sample3.txt (500 vertices, 58862 edges) is

143

```
0 0 1 1 0 2 1 0 3 4 2 1 5 4 3 4 2 6 5 7 6 8 9 7 6 10 2 11 5 3 7 12 11 13 6 10 8 0 9 14 13 3 10 14 15 12 4 16 15 17
17 7 11 13 12 9 18 10 19 20 19 21 22 9 23 24 22 21 18 12 9 25 20 14 25 26 27 24 11 13 26 16 27 28 29 20 23
28 18 29 24 30 31 32 33 30 34 35 33 15 36 37 38 39 14 17 19 3 17 40 21 8 41 32 42 23 38 26 18 43 42 18 44
21 22 41 31 11 33 32 45 46 34 24 8 31 47 25 36 40 27 40 28 34 36 42 44 20 43 45 30 46 47 48 49 35 50 51 3
7 44 52 53 54 22 45 16 27 28 55 39 47 49 56 34 50 48 57 58 59 60 37 31 61 35 51 19 16 54 62 3 55 49 15 38
63 41 29 64 57 29 52 43 36 39 27 47 56 65 66 64 67 68 63 69 70 53 71 67 70 65 41 72 37 22 73 35 25 48 55
74 56 39 68 75 76 71 74 37 37 26 77 62 78 28 79 80 81 82 46 75 50 40 32 52 57 83 61 84 30 85 46 53 86 69
51 59 87 58 82 42 88 89 23 29 90 91 72 77 79 60 92 76 39 68 80 44 91 93 94 71 54 95 59 85 39 36 96 73 97
38 61 87 98 99 83 93 98 63 62 50 45 100 100 101 82 94 56 66 64 102 66 54 84 49 67 103 48 57 58 65 102 6
7 74 80 96 97 104 99 105 104 88 101 78 106 107 108 109 110 107 86 111 110 69 112 111 113 114 115 112
73 116 97 64 116 113 81 114 117 77 115 118 85 66 68 116 119 83 78 70 111 117 79 110 92 90 120 121 122
70 121 79 62 119 84 76 53 123 86 75 60 91 124 96 123 113 72 118 125 126 60 126 127 81 128 114 129 95
80 87 130 52 84 73 99 122 55 106 93 69 109 115 103 127 81 87 94 118 119 89 95 62 131 86 126 89 88 90 1
32 125 100 133 134 134 15 127 135 128 136 83 123 102 72 128 105 137 130 78 132 133 98 138 90 30 117
136 137 104 139 129 138 89 65 51 131 108 107 140 141 142 142 141 112 135 120 8 61 120 64 108 91
```

Link of code on GitHub is :

<https://github.com/AbderrhmanAbdellatif/-Graph-Coloring-Problem>