**Inequalities, linear programming, proof, normal forms, Boolean algebra, graphs**

Inequalities

We solve linear inequalities just like we solve linear equations.

x < 5

x+1 < 5

x < 4

3x < 6

x < 2

3x + 1 < 7

3x < 6

x < 2

x > 5

x+1>5

x > 4

3x>6

x>2

3x + 1 > 7

3x > 6

x > 2

- x < - 5

x > 5

|x|<5

|x| = x, if x > 0 or x = 0

|x| = -x, if x < 0

if x > 0 or x = 0, then |x| = x < 5

if x < 0, then |x| = -x < 5, x > -5

-5 < x < 5

|x – 1| < 4

-4+1 < x < 4+1

-3 < x < 5

|x + 3| < 8

-8 -3 < x < 8-3

-11 < x < 5

|5x|<10

|x|<2

-2 < x < 2

|5x - 1| < 9

-8 < 5x < 10

-8/5 < x < 2

|-x| = |x|

|3-5x| = |5x-3|

|3-5x| < 10

|5x-3| < 10

-10+3 < 5x < 10+3

-7 < 5x < 13

-7/5 < x < 13/5

|9000 - 20x| < 23009020

|20x-9000| < 23009020

-23009020 +9000 < 20x < 23009020+9000

(-23009020 +9000)/20 < x < (23009020+9000)/20

Solve the inequalities.

m2 = 0: |k - Tx| < s

m2 = 1: |-s + Lx| - |kx + T| < s

http://www.wolframalpha.com

x < y

Linear programming

**Linear programming** (**LP**), also called **linear optimization**, is a method to achieve the best outcome (such as maximum profit or lowest cost) in a [mathematical model](https://en.wikipedia.org/wiki/Mathematical_model) whose requirements are represented by [linear relationships](https://en.wikipedia.org/wiki/Linear_function#As_a_polynomial_function). Linear programming is a special case of mathematical programming (also known as [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization)).

en.wikipedia.org/wiki/Linear\_programming

youtube.com/watch?v=-32jcGMpD2Q

youtube.com/watch?v=Uo6aRV-mbeg

Example:

Each table takes 10 units of lumber,

5 hours of labour, make $180

Each bookcase takes 20 units of lumber,

4 hours of labour, make $200

200 units of lumber available

80 hours of labour available

I want to make as much money as possible.

Optimise: f(x,y) = 180x + 200y

Constraints:

5x + 4y 80

10x + 20y 200

x 0

y 0

Solving simultaneous linear equations

5x + 4y =80

10x + 20y = 200

Dim a(2, 2), r(2)

a(1, 1) = 5

a(1, 2) = 4

a(2, 1) = 10

a(2, 2) = 20

r(1) = 80

r(2) = 200

determinant = a(1, 1) \* a(2, 2) - a(1, 2) \* a(2, 1)

'MsgBox determinant

a(1, 1) = r(1)

a(2, 1) = r(2)

determinant1 = a(1, 1) \* a(2, 2) - a(1, 2) \* a(2, 1)

'MsgBox determinant1

x = determinant1 / determinant

MsgBox x

a(1, 1) = 5

a(1, 2) = 4

a(2, 1) = 10

a(2, 2) = 20

a(1, 2) = r(1)

a(2, 2) = r(2)

determinant2 = a(1, 1) \* a(2, 2) - a(1, 2) \* a(2, 1)

'MsgBox determinant2

y = determinant2 / determinant

MsgBox y

(0,0), (0,10), (16,0), (13.3,3.3)

x = 0

y = 0

x = 0

y = 10

x = 16

y = 0

x = 40 / 3

y = 10 / 3

First = 5 \* x + 4 \* y - 80

MsgBox First

Second2 = 10 \* x + 20 \* y - 200

MsgBox Second2

profit = 180 \* x + 200 \* y

MsgBox profit

youtube.com/watch?v=K7TL5NMlKIk

**Proof**

Question:

**Give a direct proof of the theorem:**

**“IF n is an odd integer, THEN n2 is odd”**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

**Give a direct proof of the theorem:**

**“IF m is an odd integer and n is even integer, THEN m + n is odd”**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

**Give a direct proof that:**

**“IF m and n are both perfect squares, THEN nm is also a perfect square”**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

**Prove that:**

**“IF n is an integer and 3n + 2 is odd, THEN n is odd”**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

**Prove that:**

**“IF n = ab, where a and b are positive integers,**

**THEN a ≤ √n or b ≤ √n ”**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

**Prove that:**

**“IF 3n + 2 is odd, then n is odd”**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

**Prove that:**

**“IF n2 is even, then n is even”**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

Prove that

**1 + 2 + … + n = n(n+1)/2**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

Prove that

**1 + 3 + 5 + … + (2n -1) = n2**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

Prove that

**1 + 4 + 7 + … + (3n -2) = n2**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

Prove that

**1 + 2 + 22 + … + 2n = 2n+1 -1**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

Prove that

**1 + 22 + … +n2 = n(n+1)(2n+1)/6**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

Question:

Prove that

**n < 2n**

https://discrete4math.weebly.com/uploads/2/5/3/9/25393482/3proofs.ppt

**Boolean algebra**

Conjunctive normal form

Disjunctive normal form

Question:

Use conjunctive normal form and disjunctive normal form to express f(x,y,z) through x,y,z.

Conjunctive normal form:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | z | f(x,y,z) | (NOT x AND NOT y AND NOT z) OR (NOT x AND y AND z) OR (x AND y AND NOT z) |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |

Disjunctive normal form:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | z | f(x,y,z) | (x OR y OR NOT z) AND (x OR NOT y OR z) AND (NOT x OR y OR z) AND (NOT x OR y OR NOT z) AND (NOT x OR NOT y OR NOT z) |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |

The **Karnaugh map** (**KM** or **K-map**) is a method of simplifying [Boolean algebra](https://en.wikipedia.org/wiki/Boolean_algebra) expressions. [Maurice Karnaugh](https://en.wikipedia.org/wiki/Maurice_Karnaugh) introduced it in 1953 as a refinement of [Edward W. Veitch](https://en.wikipedia.org/wiki/Edward_W._Veitch)'s 1952 **Veitch chart**, which was a rediscovery of [Allan Marquand](https://en.wikipedia.org/wiki/Allan_Marquand)'s 1881 *logical diagram* aka **Marquand diagram** but with a focus now set on its utility for switching circuits. Veitch charts are also known as **Marquand–Veitch diagrams** or, rarely, as **Svoboda charts**, and Karnaugh maps as **Karnaugh–Veitch maps** (**KV maps**).

en.wikipedia.org/wiki/Karnaugh\_map

Question:

Simplify the expression for your *e*.

e = 0: A´BC + BC + AB´ + ABC + AC´ + BC´

e = 1: AB´C + B´C + A´B´ + ABC´ + AC´ + BC´

e = 2: B´C + B´C + A´B´C´ + ABC´ + AB´C´ + BC´

e = 3: BC´ + B´C + A´B´C´ + ABC´ + AB´C´ + B´C´

e = 4: A´BC + BC + AB´ + ABC + AC´ + BC´ + A´B´C

e = 5: A´BC + AB´ + ABC + AC´ + BC´ + A´B´C

e = 6: BC + AB´ + ABC + AC´ + BC´ + A´B´C

e = 7: A´BC´ + BC + AB´ + AC´ + BC´ + A´B´C

Use Karnaugh Map.

https://en.wikipedia.org/wiki/Karnaugh\_map

Question:

Find the function for your truth table for your *e*.

e = 0: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/e0\_truth\_table.docx

e = 1: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/e1\_truth\_table.docx

e = 2: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/e2\_truth\_table.docx

e = 3: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/e3\_truth\_table.docx

e = 4: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/e4\_truth\_table.docx

e = 5: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/e5\_truth\_table.docx

e = 6: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/e6\_truth\_table.docx

e = 7: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/e7\_truth\_table.docx

Graphs

Question:

Find adjacency and incidence matrixes for the graphs:

m6 = 0: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/student16number16graph16.docx

m6 = 1: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/trees16graphs2solve16.docx

m6 = 2: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/student4number4graph.docx

m6 = 3: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/student3number3graph.docx

m6 = 4: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/trees24graphs2solve.docx

m6 = 5: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/trees4graphs2solve.docx

Complete graph

In the [mathematical](https://en.wikipedia.org/wiki/Mathematics) field of [graph theory](https://en.wikipedia.org/wiki/Graph_theory), a **complete graph** is a [simple](https://en.wikipedia.org/wiki/Simple_graph) [undirected graph](https://en.wikipedia.org/wiki/Undirected_graph) in which every pair of distinct [vertices](https://en.wikipedia.org/wiki/Vertex_(graph_theory)) is connected by a unique [edge](https://en.wikipedia.org/wiki/Edge_(graph_theory)). A **complete digraph** is a [directed graph](https://en.wikipedia.org/wiki/Directed_graph) in which every pair of distinct vertices is connected by a pair of unique edges (one in each direction).

Graph theory itself is typically dated as beginning with [Leonhard Euler](https://en.wikipedia.org/wiki/Leonhard_Euler)'s 1736 work on the [Seven Bridges of Königsberg](https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg). However, [drawings](https://en.wikipedia.org/wiki/Graph_drawing) of complete graphs, with their vertices placed on the points of a [regular polygon](https://en.wikipedia.org/wiki/Regular_polygon), had already appeared in the 13th century, in the work of [Ramon Llull](https://en.wikipedia.org/wiki/Ramon_Llull). Such a drawing is sometimes referred to as a **mystic rose**.

en.wikipedia.org/wiki/Complete\_graph

Question:

What is complete graph?

In the [mathematical](https://en.wikipedia.org/wiki/Mathematics) field of [graph theory](https://en.wikipedia.org/wiki/Graph_theory), a **bipartite graph** (or **bigraph**) is a [graph](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)) whose [vertices](https://en.wikipedia.org/wiki/Vertex_(graph_theory)) can be divided into two [disjoint](https://en.wikipedia.org/wiki/Disjoint_sets) and [independent sets](https://en.wikipedia.org/wiki/Independent_set_(graph_theory))

en.wikipedia.org/wiki/Bipartite\_graph

Question:

What is bipartite graph?

Konigsberg bridge problem

The **Seven Bridges of Königsberg** is a historically notable problem in mathematics. Its negative resolution by [Leonhard Euler](https://en.wikipedia.org/wiki/Leonhard_Euler) in 1736 laid the foundations of [graph theory](https://en.wikipedia.org/wiki/Graph_theory) and prefigured the idea of [topology](https://en.wikipedia.org/wiki/Topology).

en.wikipedia.org/wiki/Seven\_Bridges\_of\_Königsberg

Question:

Solve Konigsberg bridge problem.

In [graph theory](https://en.wikipedia.org/wiki/Graph_theory), an **Eulerian trail** (or **Eulerian path**) is a [trail](https://en.wikipedia.org/wiki/Trail_(graph_theory)) in a finite graph that visits every [edge](https://en.wikipedia.org/wiki/Edge_(graph_theory)) exactly once (allowing for revisiting vertices). Similarly, an **Eulerian circuit** or **Eulerian cycle** is an Eulerian trail that starts and ends on the same [vertex](https://en.wikipedia.org/wiki/Vertex_(graph_theory)).

en.wikipedia.org/wiki/Eulerian\_path

Question:

What is Eulerian path?

Hamiltonian path

In the [mathematical](https://en.wikipedia.org/wiki/Mathematics) field of [graph theory](https://en.wikipedia.org/wiki/Graph_theory), a **Hamiltonian path** (or **traceable path**) is a [path](https://en.wikipedia.org/wiki/Path_(graph_theory)) in an undirected or directed graph that visits each [vertex](https://en.wikipedia.org/wiki/Vertex_(graph_theory)) exactly once. A **Hamiltonian cycle** (or **Hamiltonian circuit**) is a [cycle](https://en.wikipedia.org/wiki/Cycle_(graph_theory)) that visits each vertex exactly once. A Hamiltonian path that starts and ends at adjacent vertices can be completed by adding one more edge to form a Hamiltonian cycle, and removing any edge from a Hamiltonian cycle produces a Hamiltonian path. Determining whether such paths and cycles exist in graphs (the [Hamiltonian path problem](https://en.wikipedia.org/wiki/Hamiltonian_path_problem) and [Hamiltonian cycle problem](https://en.wikipedia.org/wiki/Hamiltonian_cycle_problem)) are [NP-complete](https://en.wikipedia.org/wiki/NP-complete).

en.wikipedia.org/wiki/Hamiltonian\_path

Question:

What is Hamiltonian path?

Djikstra algorithm

**Dijkstra's algorithm** ([/ˈdaɪkstrəz/](https://en.wikipedia.org/wiki/Help:IPA/English) [*DYKE-strəz*](https://en.wikipedia.org/wiki/Help:Pronunciation_respelling_key)) is an [algorithm](https://en.wikipedia.org/wiki/Algorithm) for finding the [shortest paths](https://en.wikipedia.org/wiki/Shortest_path_problem) between [nodes](https://en.wikipedia.org/wiki/Vertex_(graph_theory)) in a weighted [graph](https://en.wikipedia.org/wiki/Graph_(abstract_data_type)), which may represent, for example, [road networks](https://en.wikipedia.org/wiki/Road_network). It was conceived by [computer scientist](https://en.wikipedia.org/wiki/Computer_scientist) [Edsger W. Dijkstra](https://en.wikipedia.org/wiki/Edsger_W._Dijkstra" \o "Edsger W. Dijkstra) in 1956 and published three years later.

en.wikipedia.org/wiki/Dijkstra%27s\_algorithm

Question:

Explain Djikstra algorithm.

In [computer science](https://en.wikipedia.org/wiki/Computer_science), **Prim's algorithm** (also known as Jarník's algorithm) is a [greedy algorithm](https://en.wikipedia.org/wiki/Greedy_algorithm) that finds a [minimum spanning tree](https://en.wikipedia.org/wiki/Minimum_spanning_tree) for a [weighted](https://en.wikipedia.org/wiki/Weighted_graph) [undirected graph](https://en.wikipedia.org/wiki/Undirected_graph). This means it finds a subset of the [edges](https://en.wikipedia.org/wiki/Edge_(graph_theory)) that forms a [tree](https://en.wikipedia.org/wiki/Tree_(graph_theory)) that includes every [vertex](https://en.wikipedia.org/wiki/Vertex_(graph_theory)), where the total weight of all the [edges](https://en.wikipedia.org/wiki/Graph_theory) in the tree is minimized. The algorithm operates by building this tree one vertex at a time, from an arbitrary starting vertex, at each step adding the cheapest possible connection from the tree to another vertex.

en.wikipedia.org/wiki/Prim%27s\_algorithm

**Kruskal's algorithm** (also known as **Kruskal's method**) finds a [minimum spanning forest](https://en.wikipedia.org/wiki/Minimum_spanning_tree) of an undirected [edge-weighted graph](https://en.wikipedia.org/wiki/Weighted_graph). If the graph is [connected](https://en.wikipedia.org/wiki/Connectivity_(graph_theory)), it finds a [minimum spanning tree](https://en.wikipedia.org/wiki/Minimum_spanning_tree). (A minimum spanning tree of a connected graph is a subset of the [edges](https://en.wikipedia.org/wiki/Edge_(graph_theory)) that forms a tree that includes every [vertex](https://en.wikipedia.org/wiki/Vertex_(graph_theory)), where the sum of the [weights](https://en.wikipedia.org/wiki/Weighted_graph) of all the edges in the tree is minimized. For a disconnected graph, a minimum spanning forest is composed of a minimum spanning tree for each [connected component](https://en.wikipedia.org/wiki/Connected_component_(graph_theory)).) It is a [greedy algorithm](https://en.wikipedia.org/wiki/Greedy_algorithm) in [graph theory](https://en.wikipedia.org/wiki/Graph_theory) as in each step it adds the next lowest-weight edge that will not form a [cycle](https://en.wikipedia.org/wiki/Cycle_(graph_theory)) to the minimum spanning forest.

en.wikipedia.org/wiki/Kruskal%27s\_algorithm

In [computer science](https://en.wikipedia.org/wiki/Computer_science), **tree traversal** (also known as **tree search** and **walking the tree**) is a form of [graph traversal](https://en.wikipedia.org/wiki/Graph_traversal) and refers to the process of visiting (e.g. retrieving, updating, or deleting) each node in a [tree data structure](https://en.wikipedia.org/wiki/Tree_(data_structure)), exactly once. Such traversals are classified by the order in which the nodes are visited. The following algorithms are described for a [binary tree](https://en.wikipedia.org/wiki/Binary_tree), but they may be generalized to other trees as well.

en.wikipedia.org/wiki/Tree\_traversal

Question:

Apply Dijkstra’s, Prim's and Kruskal’s algorithms to the graphs. Traverse the trees.

m6 = 0: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/student16number16graph16.docx

m6 = 1: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/trees16graphs2solve16.docx

m6 = 2: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/student4number4graph.docx

m6 = 3: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/student3number3graph.docx

m6 = 4: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/trees24graphs2solve.docx

m6 = 5: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/trees4graphs2solve.docx

Game tree

In the context of [combinatorial game theory](https://en.wikipedia.org/wiki/Combinatorial_game_theory), which typically studies [sequential games](https://en.wikipedia.org/wiki/Sequential_game) with [perfect information](https://en.wikipedia.org/wiki/Perfect_information), a **game tree** is a graph representing all possible game states within such a game. Such games include well-known ones such as [chess](https://en.wikipedia.org/wiki/Chess), [checkers](https://en.wikipedia.org/wiki/Draughts), [Go](https://en.wikipedia.org/wiki/Go_(board_game)), and [tic-tac-toe](https://en.wikipedia.org/wiki/Tic-tac-toe). This can be used to measure the [complexity of a game](https://en.wikipedia.org/wiki/Game_complexity), as it represents all the possible ways a game can pan out. Due to the large game trees of [complex games](https://en.wikipedia.org/wiki/Game_complexity#Complexities_of_some_well-known_games) such as chess, algorithms that are designed to play this class of games will use partial game trees, which makes computation feasible on modern computers. Various methods exist to solve game trees. If a complete game tree can be generated, a [deterministic algorithm](https://en.wikipedia.org/wiki/Deterministic_algorithm), such as [backward induction](https://en.wikipedia.org/wiki/Backward_induction) or [retrograde analysis](https://en.wikipedia.org/wiki/Retrograde_analysis) can be used. [Randomized algorithms](https://en.wikipedia.org/wiki/Randomized_algorithm) and [minimax](https://en.wikipedia.org/wiki/Minimax) algorithms such as [MCTS](https://en.wikipedia.org/wiki/Monte_Carlo_tree_search) can be used in cases where a complete game tree is not feasible.

en.wikipedia.org/wiki/Game\_tree

Travelling salesman problem

The **travelling salesman problem** (**TSP**) asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?" It is an [NP-hard](https://en.wikipedia.org/wiki/NP-hardness) problem in [combinatorial optimization](https://en.wikipedia.org/wiki/Combinatorial_optimization), important in [theoretical computer science](https://en.wikipedia.org/wiki/Theoretical_computer_science) and [operations research](https://en.wikipedia.org/wiki/Operations_research).

en.wikipedia.org/wiki/Travelling\_salesman\_problem

Question:

How many options are there for traveling salesman problem?

Question:

Solve travelling salesman problem.

Planar graph

In [graph theory](https://en.wikipedia.org/wiki/Graph_theory), a **planar graph** is a [graph](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)) that can be [embedded](https://en.wikipedia.org/wiki/Graph_embedding) in the [plane](https://en.wikipedia.org/wiki/Plane_(geometry)), i.e., it can be drawn on the plane in such a way that its edges intersect only at their endpoints. In other words, it can be drawn in such a way that no edges cross each other.

en.wikipedia.org/wiki/Planar\_graph

Question:

Explain planar graph.

Kuratowski theorem

In [graph theory](https://en.wikipedia.org/wiki/Graph_theory), **Kuratowski's theorem** is a mathematical [forbidden graph characterization](https://en.wikipedia.org/wiki/Forbidden_graph_characterization) of [planar graphs](https://en.wikipedia.org/wiki/Planar_graph), named after [Kazimierz Kuratowski](https://en.wikipedia.org/wiki/Kazimierz_Kuratowski).

en.wikipedia.org/wiki/Kuratowski%27s\_theorem

Question:

Explain Kuratowski theorem.

Question:

How many edges are there in KT, K(a+1),(m+1)?

Question:

Find the number of faces for your graph.

Question:

Do graceful labeling of your graph.

http://azspcs.com/Contest/GracefulGraphs

Question:

Give Euler’s, Hamiltonian’s cycles, and paths in the graphs:

m4 = 0: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/student16number16graph16.docx

m4 = 1: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/trees16graphs2solve16.docx

m4 = 2: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/euler6cycle.ppt

m4 = 3: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/question2euler.ppt

Question:

Is the graph planar? Why?

m7 = 0: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/graph22jun16.docx

m7 = 1: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/graph7am22jun16.docx

m7 = 2: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/graph8am22jun16.docx

m7 = 3: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/graph1pm22june16.docx

m7 = 4: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/student16number16graph16.docx

m7 = 5: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/trees16graphs2solve16.docx

m7 = 6: http://discrete4math.weebly.com/uploads/2/5/3/9/25393482/planar4graphs.ppt

Question:

What is the number of faces for your graceful graph?

Question:

Is your graceful graph planar? Why?

https://en.wikipedia.org/wiki/Kuratowski%27s\_theorem

Question:

Find the graceful labeling of

ii = FreeFile

Open "D:\bipartate33k.txt" For Output As #ii

Dim e(9)

e(1) = 9

Max = 8

For v1 = 1 To Max - 3

For v2 = v1 + 1 To Max - 2

For v3 = v2 + 1 To Max - 1

For v4 = v3 + 1 To Max

e(2) = v4

e(3) = v3

e(4) = Abs(9 - v1)

e(5) = Abs(v4 - v1)

e(6) = Abs(v3 - v1)

e(7) = Abs(9 - v2)

e(8) = Abs(v4 - v2)

e(9) = Abs(v3 - v2)

If e(1) <> e(2) And e(1) <> e(3) And e(1) <> e(4) And e(1) <> e(5) Then GoTo 11

GoTo 10

11 If e(1) <> e(6) And e(1) <> e(7) And e(1) <> e(8) And e(1) <> e(9) Then GoTo 111

GoTo 10

111 If e(2) <> e(3) And e(2) <> e(4) And e(2) <> e(5) And e(2) <> e(6) And e(2) <> e(7) And e(2) <> e(8) Then GoTo 22

GoTo 10

22 If e(2) <> e(9) Then GoTo 222

GoTo 10

222 If e(3) <> e(4) And e(3) <> e(5) And e(3) <> e(6) And e(3) <> e(7) And e(3) <> e(8) And e(3) <> e(9) Then GoTo 33

GoTo 10

33 If e(4) <> e(5) And e(4) <> e(6) And e(4) <> e(7) And e(4) <> e(8) And e(4) <> e(9) Then GoTo 44

GoTo 10

44 If e(5) <> e(6) And e(5) <> e(7) And e(5) <> e(8) And e(5) <> e(9) Then GoTo 55

GoTo 10

55 If e(6) <> e(7) And e(6) <> e(8) And e(6) <> e(9) Then GoTo 66

GoTo 10

66 If e(7) <> e(8) And e(7) <> e(9) Then GoTo 77

GoTo 10

77 If e(8) <> e(9) Then GoTo 1

10 Next v4

Next v3

Next v2

Next v1

GoTo 2

1 Print #ii, v1, v2, v3, v4

2 End Sub

Question:

Color your graphs using as few colors as possible. Find the chromatic numbers of the graphs.

Question:

Color the map of the country number T using as few colors as possible.

http://www.worldometers.info/geography/alphabetical-list-of-countries/

Question:

Find the number of regions for the graph with L+20 edges and e+10 vertices.

Petersen graph:

Question:

m6 = 0: Does Petersen graph satisfy the condition e < 2v – 4?

m6 = 1: Does Petersen graph satisfy the condition e < 3v – 6?

m6 = 2: Is this graph planar?

m6 = 3: Why?

m6 = 4: Which important graph is Petersen graph similar to?

m6 = 5: Do graceful labeling of Petersen graph.

http://azspcs.com/Contest/GracefulGraphs

https://en.wikipedia.org/wiki/Petersen\_graph

https://en.wikipedia.org/wiki/Planar\_graph

https://en.wikipedia.org/wiki/Kuratowski%27s\_theorem