```
begin
    include("../shared/savemarked.jl")
    savemarked()
end
```

```
saved graphcol_1_data.jl
```

graphcol_1

```
begin
    md"""#### graphcol_1"""
end
```

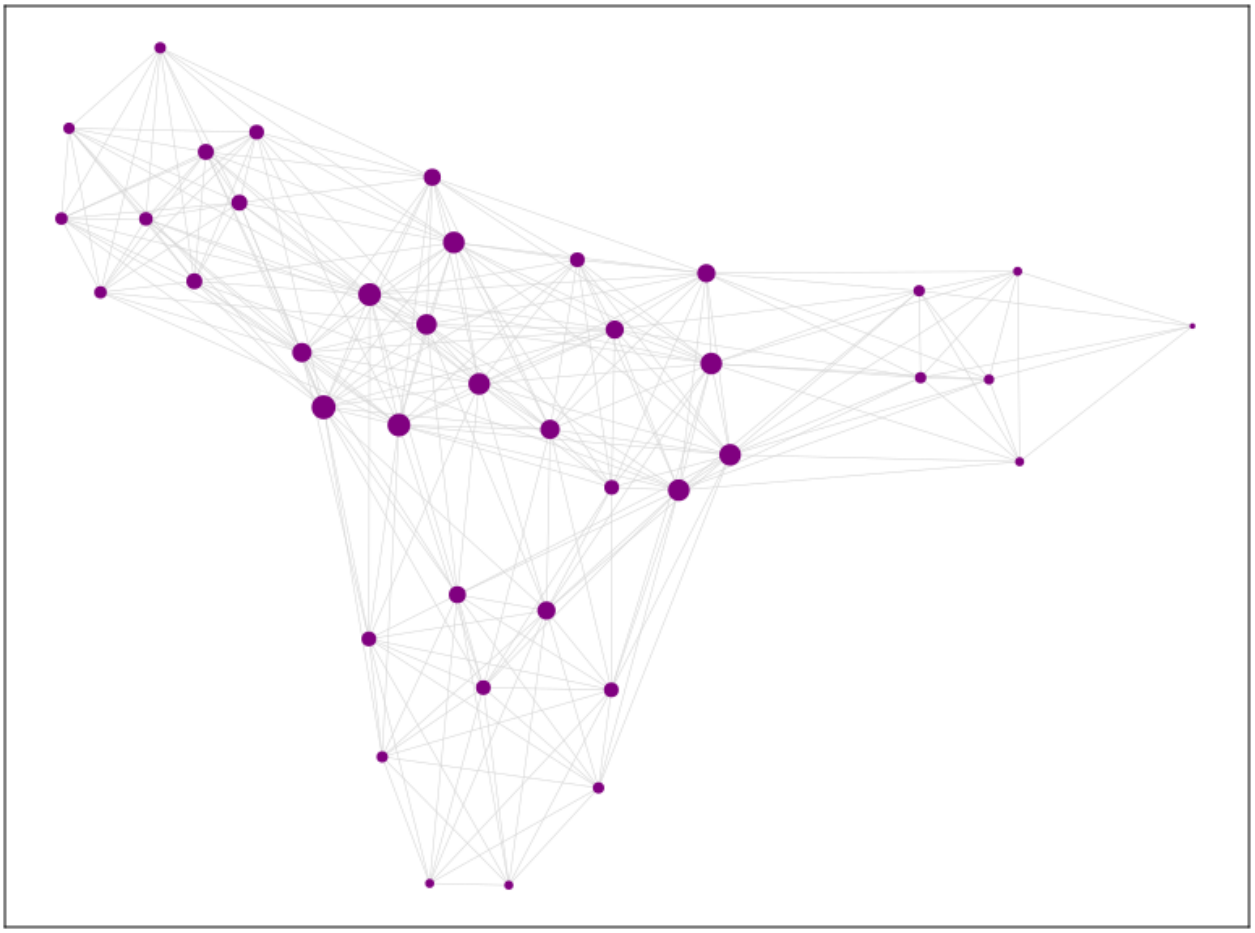
```
begin
    import Pkg
    Pkg.activate(".")


    Pkg.add.(
        [
            "DelimitedFiles",
            "Graphs", "Colors",
            "DataFrames", "StatsBase",
            "CairoMakie", "GraphMakie"
        ]
    )
    Pkg.instantiate()

    using
        DelimitedFiles,
        Graphs,Colors,
        DataFrames, StatsBase,
        CairoMakie, GraphMakie
end
```

```
  Activating project at `~/Asztal/git/plnotebooks/graphcol_1`
   Resolving package versions...
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Project.toml`
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Manifest.toml`
Precompiling project...
  ✓ CairoMakie
  1 dependency successfully precompiled in 42 seconds. 211 already precompi
led.
   Resolving package versions...
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Project.toml`
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Manifest.toml`
   Resolving package versions...
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Project.toml`
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Manifest.toml`
   Resolving package versions...
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Project.toml`
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Manifest.toml`
   Resolving package versions...
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Project.toml`
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Manifest.toml`
   Resolving package versions...
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Project.toml`
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Manifest.toml`
   Resolving package versions...
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Project.toml`
  No Changes to `~/Asztal/git/plnotebooks/graphcol_1/Manifest.toml`
```

["Biology of the Cell", "Molecular Biology", "Evolution", "Biochemistry", "Neurobiolo

```
begin
#--->graphcol_1_data

function graphcol_1_data()
    # read the data
    d0,h0=readdlm(
        "../data/synthetic_school_enrollment_data.csv",',';
        header=true
    )
    # convert the original data

    # drop out the first three columns (name,major/minor)
    # and convert it to a valid logical matrix
    data=map(
        x->if x=="True"
            true
        elseif x=="False"
            false
        else
            throw(error("unknown value"))
        end,
        d0[:,4:end]
    )

    header=h0[4:end]
    num_of_students,num_of_courses=size(data)

    # build the graph:
    # the nodes are the courses with an edge between them if there is a student visiting either.

    # first, collect the set of students visiting each courses
    S=[Set((1:num_of_students)[col]) for col in eachcol(data)]


    # then, use the sets
    G=Graph()
    add_vertices!(G,num_of_courses)
    for i in 1:num_of_courses-1, j in i+1:num_of_courses
        !isdisjoint(S[i],S[j]) && add_edge!(G,i,j)
    end
    (
        G=G,
        num_of_students=num_of_students,num_of_courses=num_of_courses,
        header=header
    )
end

#--->graphcol_1_data


data=graphcol_1_data()
G=data.G
num_of_students=data.num_of_students
num_of_courses=data.num_of_courses
header=data.header

end
```

```
end
```

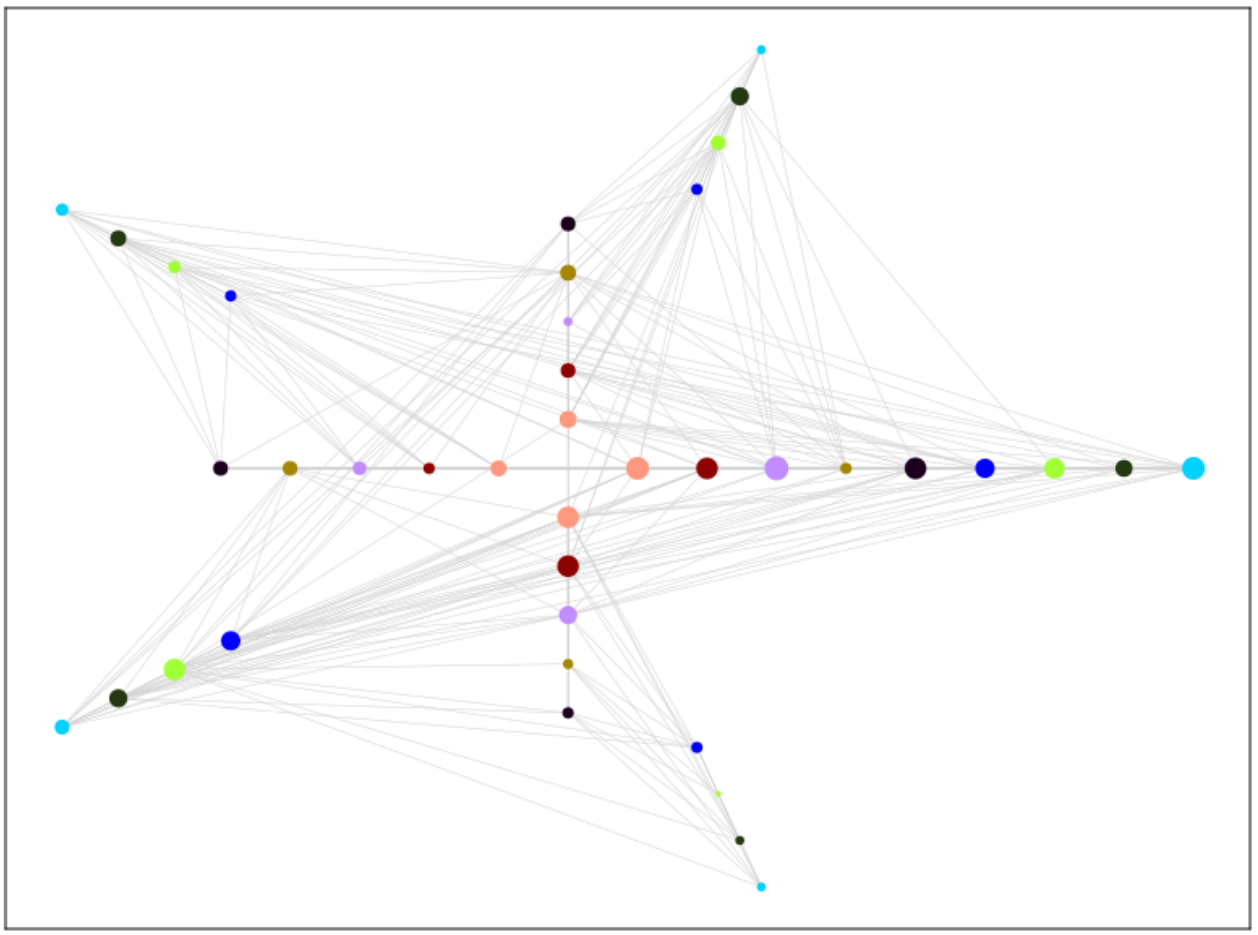


```
begin
    # plot the graph
    deg=degree(G)
    scene=graphplot(
        G,
        node_size=deg,
        node_color="Purple",
        edge_color="LightGray",
        edge_width=0.5,
    )
    hidedecorations!(scene.axis)
    scene
end
```

```
Coloring(9, [5, 1, 4, 3, 9, 6, 2, 7, 9,    more ,5])
```

```
begin
    # as in networkX in Graph.jl there is a "builtin" method
    # greedy_color(G; reps) to generate
    # colorings, therefore we'll use it
    # it returns an object w/ num_colors and colors fields
    # we need col.num_colors dates for the exams
    @time the_coloring=greedy_color(G; reps=100)

end
```

```
  0.000469 seconds (4.20 k allocations: 423.828 KiB)
```



```
begin
    # plotting the graph w/ colors assigned
    # shell layout would be better but
    # see https://github.com/JuliaGraphs/GraphPlot.jl/pull/186
    dc=distinguishable_colors(the_coloring.num_colors, colorant"blue")

    # first is the innermost
    the_shells=[[] for c in 1:the_coloring.num_colors]
    for v in vertices(G)
        push!(the_shells[the_coloring.colors[v]],v)
    end
    sort!(the_shells, by=x->length(x))

    colored_G=graphplot(
        G,
        layout=GraphMakie.Shell(
            ;
            nlist=the_shells
        ),
        node_size=deg,
        node_color=dc[the_coloring.colors],
        edge_color="LightGray",
        edge_width=0.5,
    )
    hidedecorations!(colored_G.axis)
    colored_G
end
```

	Exam	Room-1	Room-2	Ro
1	"Exam-1"	"Molecular Biology"	"Classical Mechanics"	"Linear Algeb
2	"Exam-2"	"Genetics"	"Linear Algebra for the Sciences"	"Statistics I
3	"Exam-3"	"Biochemistry"	"Complex Systems"	"Statistics I
4	"Exam-4"	"Evolution"	"Thermodynamics"	"Geometry"
5	"Exam-5"	"Biology of the Cell"	"Material Science"	"Calculus II"
6	"Exam-6"	"Animal Behavior"	"Nanotechnologies"	"Programming
7	"Exam-7"	"Bioinformatics"	"Programming for Physics"	"Calculus I"
8	"Exam-8"	"Robotics"	"Probability II"	"Data Science
9	"Exam-9"	"Neurobiology"	"Quantum Mechanics"	"Probability

```
begin
    # we need maxcolsize rooms
    cm=the_coloring.colors|>countmap
    mincolsize,maxcolsize=extrema(nc for (c,nc) in cm)

    # build the final table
    # exams for courses with the color 'k' will be held on the 'k'-th date given
    table=fill("-",the_coloring.num_colors,maxcolsize) # indices for filling in
    idx=fill(0,the_coloring.num_colors)
    for i in 1:num_of_courses
      ri=the_coloring.colors[i]
      ci=(idx[ri]+=1)
      table[ri,ci]=header[i]
    end

    df=DataFrame(
      hcat("Exam-".*string.(1:the_coloring.num_colors),table),
      vcat("Exam","Room-".*string.(1:maxcolsize)))
end
```

InterruptException:

```
begin
    md"""
    #### Note
    * inspecting the data and the result in the original tutorial more closely one can found that the 2021-06-15 18:00 Bioinformatics and Data Science exams share a student, namely Katrina Scott (Computer Science major/no minor). So, the tutorial's program has some error (which explains why we see different number of edges in the graphs)
    """
end
```