BDMA - Massive Graph Management and Analytics

Jose Antonio Lorencio Abril

Fall 2023

Professor: Nacéra Seghouani

Student e-mail: jose-antonio.lorencio-abril@student-cs.fr

This is a summary of the course *Massive Graph Management and Analytics* taught at the Université Paris Saclay - CentraleSupélec by Professor Nacéra Seghouani in the academic year 23/24. Most of the content of this document is adapted from the course notes by Seghouani, [1], so I won't be citing it all the time. Other references will be provided when used.

Contents

1 Introduction

Graph-structured data is at the heart of complex systems and plays a major role in our daily life, science and economy. Examples of this data are the cooperation between billions of individuals, or communication infraestructures with billions of cell phones, computers and satellites, the interactions between thousands of genes and metabolites within our cells, and so on.

Therefore, understanding its mathematical foundations, description, prediction, and eventually being able to control them is one of the major scientific challenges of the 21st century.

2 Preliminaries

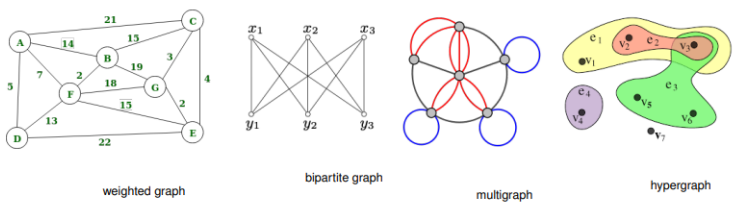
2.1 Graph Theory Preliminaries

A graph is a pair $G = (V, E)$, where V is the set of vertices and $E \subset V \times V$ is the set of edges. Usually, we denote $|V| = n$ and $|E| = m$.

There are different types of graphs:

- **Undirected**: $(u, v) \in E \implies (v, u) \in E$. That is, the edges goes in both directions.
- **Directed**: $(u, v) \in E \not\implies (v, u) \in E$. That is, the edges have direction, and it is possible that an edge goes from u to v, but not the other way.
- **Weigthed vertices**: the vertices have a weight. That is, there is a function $w_v : V \to \mathbb{R}$.
- **Weigthed edges**: the edges have a weight. That is, there is a function $w_e : E \to \mathbb{R}$.
- **Labeled vertices**: the vertices have a label, $L_v : V \to \mathcal{L}$, where $\mathcal{L}$ is the set of labels.
- **Labeled edges**: the edges have a label, $L_e : E \to \mathcal{L}$.
- **Bipartite**: a graph $G = (V, E)$ is bipartite if there is a partition of the vertices, $V = V_1 \cup V_2$, such that $V_1 \cap V_2 = \emptyset$ and $E = \{(v_i, v_j) \,|v_i \in V_1, v_j \in V_j\}$. That is, the vertices in V_1 only connect to vertices in V_2, and viceversa.
- **k-Partite**: a graph $G = (V, E)$ is k-partite if there is a k-partition of the vertices, $V = V_1 \cup V_2 \cup ... \cup V_k$, such that $V_i \cap V_j = \emptyset, \forall i \neq j$ and the is no edge $e = (u, v)$ such that $u, v \in V_i$, for the same i.
- **Multigraph** or **multidigraph**: in this case, there can be several edges between two vertices. For this, we define the edges as a separate set E, and a function $r : E \to V \times V$, that assigns the vertices related by that edge.
- **Hypergraph**: in this case, $E \subset 2^V$. That it, the edges can relate 0 or more vertices. In this case, it is more appropriate to interpret E as a set of classes or hierarchies, rather than edges.
- **Complete**: a graph is complete if $E = V \times V$.

Some examples are:



Continuing with definitions, let $G = (V, E)$ be a graph (directed or undirected). Let d_i^+ and d_i^- denote the number of edges coming out and coming to v_i, respectively. The **degree** of v_i is

$$d_i = d_i^+ + d_i^- .$$

Note that it counts double for undirected graphs.

Now, let N_i^+ and N_i^- the set of successors and predecessors of v_i, respectively. Then, the set of **neighbors** of v_i is

$$N_i = N_i^+ + N_i^- .$$

A **path** between two vertices, $u, v \in V$, denoted $u \rightsquigarrow v$, is a sequence of vertices $(u = v_0, v_1, ..., v_{k-1}, v_k = v)$, where $(v_{i-1}, v_i) \in E, \forall i = 1, ..., k$. The length of a path, $L(u \rightsquigarrow v)$, is the number of edges in the cycle, that is, k.

A **cycle** is a path from a vertex to itself, $u \rightsquigarrow u$.

The **distance** between two nodes, $d(u, v)$, is the shortest path length between them:

$$d(u, v) = \min_{u \rightsquigarrow v} L(u \rightsquigarrow v) .$$

The **eccentricity** of a node, $ecc(u)$, is the greatest distance between u and any other vertex in the graph:

$$ecc(u) = \max_{v \in V} d(u, v) .$$

Note that this could be infinity if we cannot reach some node from u. Usually, we consider only reachable nodes, because this can give us information about the graph, but a value of infinity is not very informative.

The **diameter** of a graph, $diam(G)$, is the greatest distance between two nodes in the graph:

$$diam(G) = \max_{u,v \in V} d(u, v) = \max_{u \in V} ecc(u) .$$

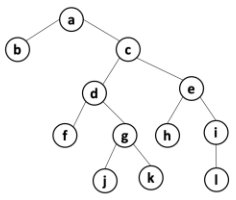
The **radius** of a graph, $rad(G)$, is the minimum eccentricity of any vertex in the graph:

$$rad(G) = \min_{u \in V} ecc(u) .$$

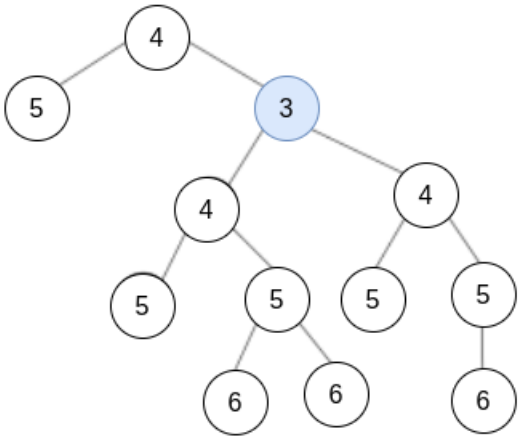
The **center** of a graph, $C(G)$, is the set of all vertices of minimum eccentricity, i.e., the graph radius:

$$C(G) = \{u : ecc(u) = rad(G)\} .$$

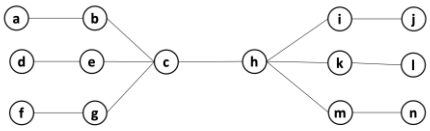
Example 2.1. Compute the diameter, radius and center of the following graphs:



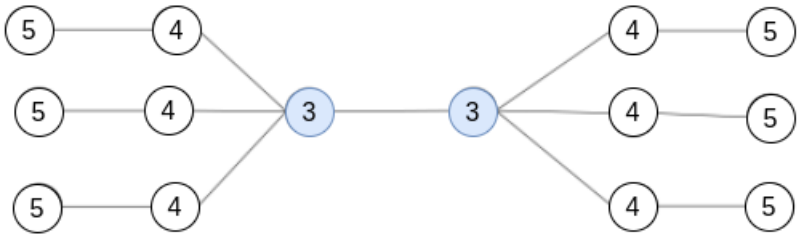
The solution is the following:



In each node, we show its eccentricity. The diameter is 6, the radius is 3 and the center is c (in blue).



Solution:



In this case, the diameter is 5, the radius is 3 and the center is $\{c, h\}$.

A **partial graph** of $G = (V, E)$ is a graph $G' = (V, E')$, where $E' \subset E$.

A **subgraph** of $G = (V, E)$ is a graph $G' = (V', E')$ where $V' \subset V$ and $E' \subset E$. Note that partial graphs are also subgraphs.

A graph $G = (V, E)$ is said to be **connected** if, and only if, $\forall u, v \in V, \exists u \rightsquigarrow v$.

A (strongly) **connected component** of $G = (V, E)$ is a subgraph $G_{cc} = (V_{cc}, E_{cc})$, where $\forall u, v \in V_{cc}, \exists u \rightsquigarrow v \in V_{cc}$. That it, a connected subgraph. It is called strongly when the paths are directed.

A graph $G = (V, E)$ is a **tree** if, and only if, G is a connected graph without cycles. In this case, the graph has $m = n - 1$ edges.

A graph $G = (V, E)$ is a **forest** if, and only if, all connected components of G are trees.

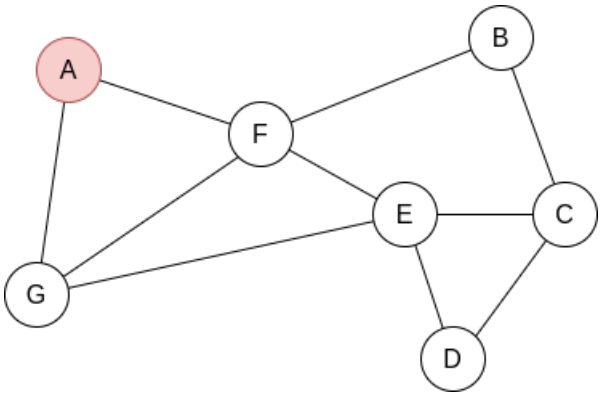
2.1.1 Breadth First Search (BFS)

BFS is a method to traverse the nodes of a graph, by starting at one node and traversing all its neighbours. Then, all neighbours of its neighbours, and so on.

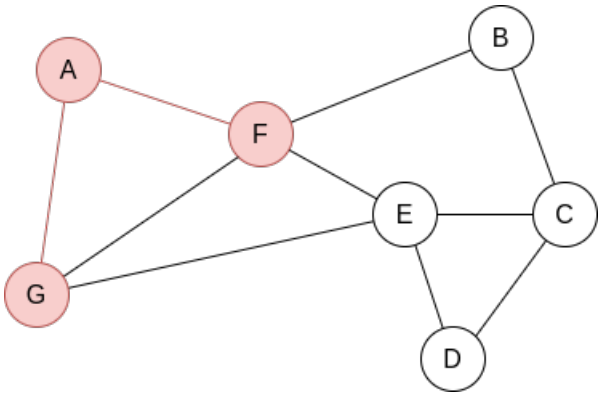
For this, we use a FIFO queue. The algorithm is:

```
procedure BFS(G=(V,E), r)
  Q <- emptyset
  enqueue(Q,r)
  r.label = True

  while Q is not empty do
    v <- dequeue(Q)
    for neig in neighbours(v) do
      if not neig.label then
        enqueue(Q,neig)
        neig.label = True
      end if
    end for
  end while
end procedure
```

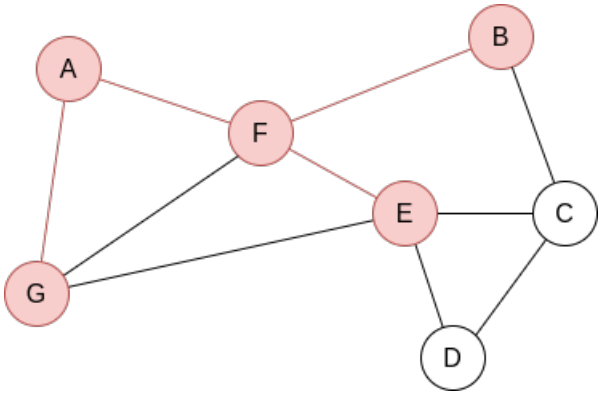
Example 2.2. Apply BFS in the following graph, starting at node A.



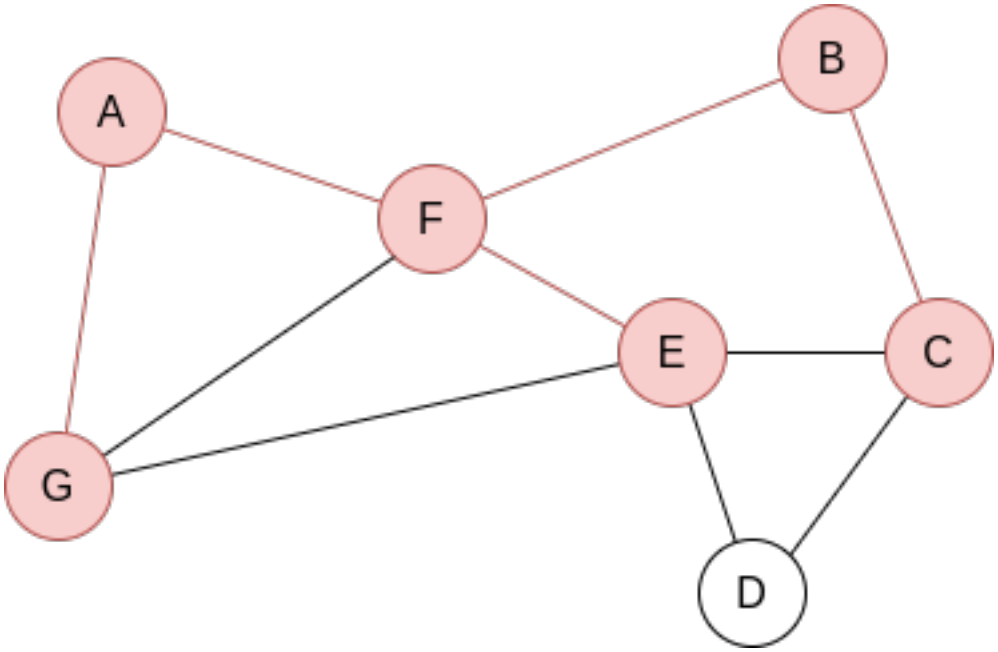
Q=[A]. We visit A's neighbours:



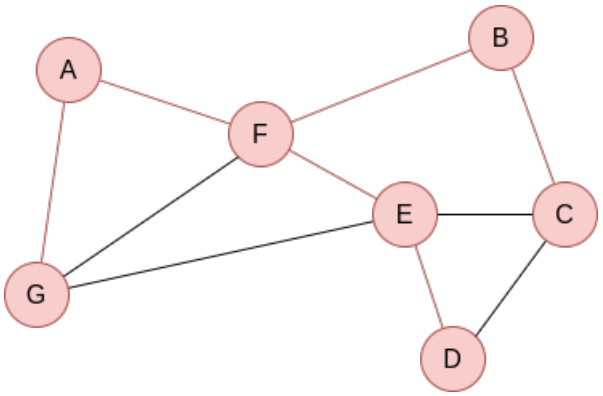
Q=[F,G]. Now, by lexycographical order, we visit F's neighbours:



Q=[G,B,E]. Now, we visit G's neighbours. Since it has no new unvisited neighbours, there is no change.
Q=[B,E]. Now, we visit B's neighbours:



Q=[E,C]. Now, we visit E's neighbours:



Q=[C,D]. Everything is visited, so the queue will be slowly emptied!

2.1.2 Depth First Search (DFS)

In the case of DFS, the objective is also to traverse the whole graph. The difference is that in this case we try to go as deep as we can in the graph before visiting more neighbours.

It can be implemented with a stack, let it be a explicit stack, or an implicit one.

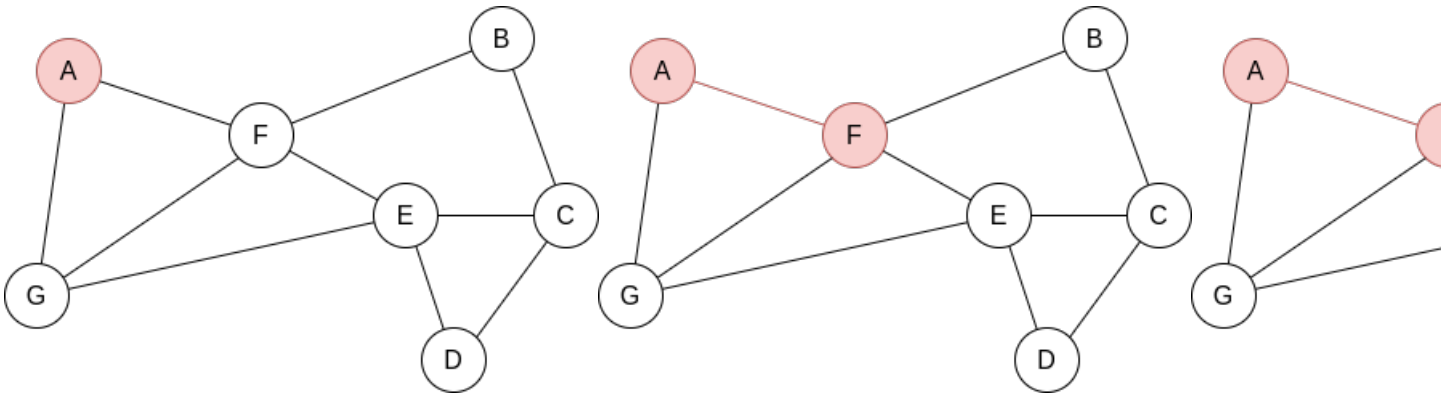
The implementation with an explicit stack is the following:

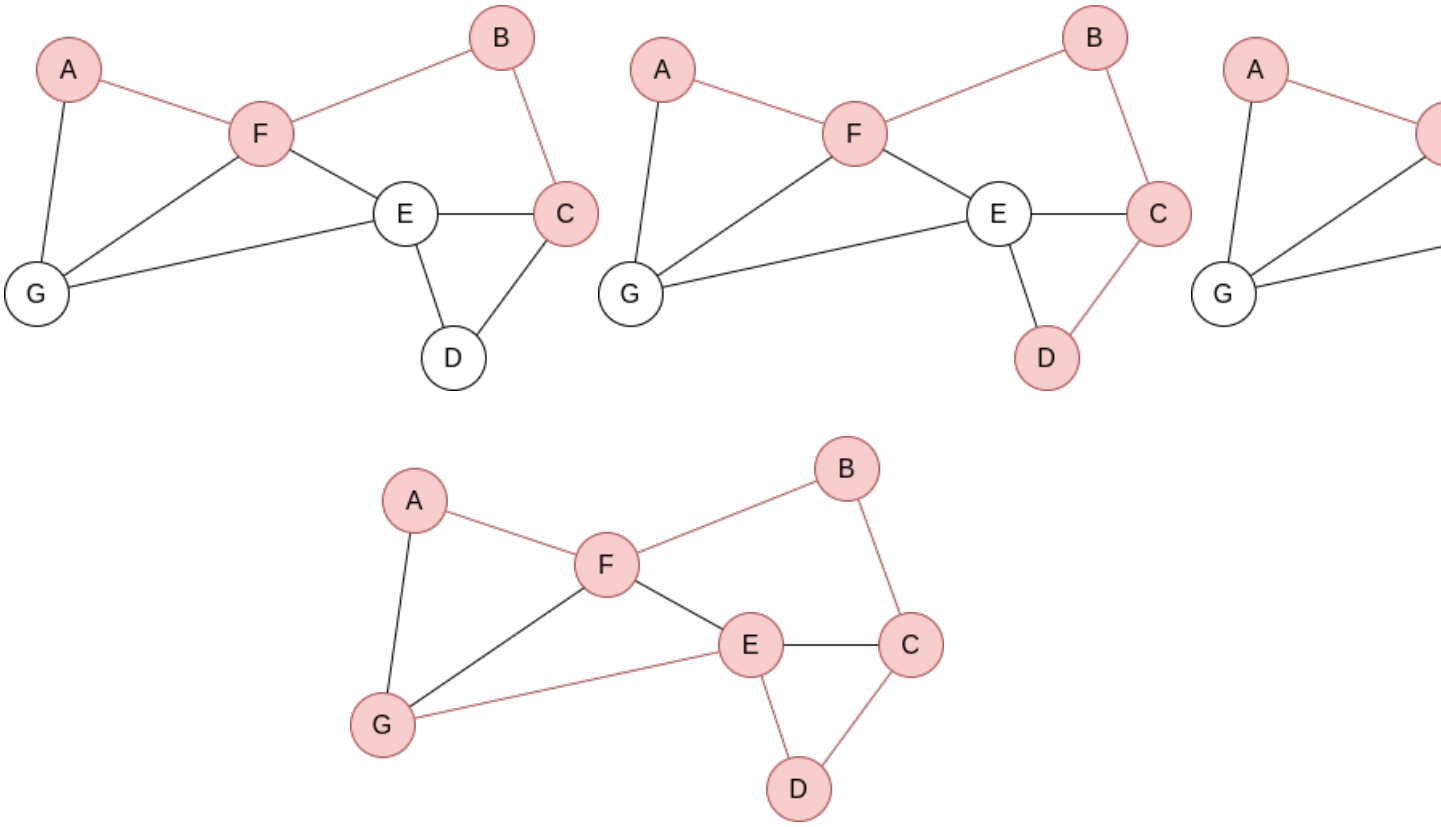
```
procedure DFS(G=(V,E),r)
  S <- emptyset
  push(S,r)
  while S is not empty do
    v <- pop(S)
    if not v.label then
      v.label = true
      for neig in neighbours(v) do
        push(S, neig)
      end for
    end if
  end while
end procedure
```

The implementation with an implicit stack is recursive, and is as follows:

```
procedure BFS(G=(V,E), r)
  Q <- emptyset
  enqueue(Q,r)
  r.label = True

  while Q is not empty do
    v <- dequeue(Q)
    for neig in neighbours(v) do
      if not neig.label then
        enqueue(Q,neig)
        neig.label = True
      end if
    end for
  end whileDFS*(G=(V,E), r)
  r.label = true
  for neig in neighbours(r) do
    if not neig.label then
      DFS*(G, neig)
    end if
  end for
end procedure
```

Example 2.3. Let's repeat the example, now using DFS:





2.1.3 Greaph Representations

A graph, $G = (V, E)$, with n vertices and m edges can be encoded using different structures:

- **Adjacency matrix**: a matrix $A \in \mathcal{M}_{n\times n}$, defined by

$$A_{ij} = \begin{cases} 1 & if,\ (v_i, v_j) \in E \\ 0 & otherwise \end{cases}.$$

  The adjacency matrix is symmetric for undirected graphs.

- **Adjacency list**: a list L of length n in which each vertex holds a list of its neighbours:

$$\forall u \in V, L_u = \{v |\, (u, v) \in E\}.$$

  If G is directed, the choice of the direction depends on the analytic needs.

- **Incidence matrix**: a matrix $B \in \mathcal{M}_{n\times m}$, defined by

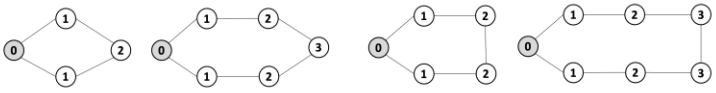
$$B_{ij} = \begin{cases} 1 & if\ e_j = (v_i, v_k) \in E \\ 0 & otherwise \end{cases}.$$

2.1.4 Exercises

1. Using graph traversal algorithms, propose an algorithm that computes the number of edges between a given vertex and all other vertices.

```
procedure n_edges(G=(V,E), r)
  Q <- emptyset
  enqueue(Q,r)
  r.n_edges = 0

  while Q is not empty do
    v <- dequeue(Q)
    for neig in neighbours(v) do
      if not neig.n_edges then
        enqueue(Q,neig)
        neig.n_edges = v.n_edges + 1
      end if
    end for
  end while
end procedure
```

2. Given the following cycles with even and odd lengths (with the distances or depths from the grey vertex), what do you think about the case of graphs with an odd cycle (in number of edges)? Is this a characteristic property? State the general case.



Proposition: a graph contains a cycle C with an odd number of edges if, and only if, $\exists\,(x,y) \in E | depth\,(x) = depth\,(y)$.

Proof: first, we know that all edges connect vertices of 'neighbouring' depths. That it, $\forall\,(x,y) \in E$, it holds $|depth\,(x) - depth\,(y)| \leq 1$.

$[\implies]$ By reduction ad absurdum, seeking a contradiction, suppose that $\forall\,(x,y) \in C$, with $depth\,(x) \neq depth\,(y)$. This means that $depth\,(x) = depth\,(y) \pm 1$. Therefore, there is, along the cycle, a node of even depth, followed by a node of odd lenght, and so on. When we close the cycle, the final node is the inicial one, so its depth is 0 (even). Therefore, we need an even number of edges, to conserve the parity.

$[\impliedby]$ If there is an edge $(x,y) \in E$ with $depth\,(x) = depth\,(y)$, then we can consider the path tree that was used to annotate the depths. In this tree, x and y have a first ancestor z in common, from which we can form an odd cycle of size $2 \cdot (depth\,(x) - depth\,(z)) + 1$ by adding the edge (x,y) to this subtree starting at z.

3. Propose an algorithm that determines if a graph contains an odd cycle.

```
procedure hasOddCycle(G=(V,E))
  v <- a vertex from V
  depths <- n_edges(G,v) #from the first exercise

  for (u,v) in E do
    if depth[u] == depth[v] then
      return True
    end if
  end for
end procedure
```

4. In a bipartite graph, can there be a cycle with an odd number of edges? Is this a characteristic property?

   No, it is not possible!

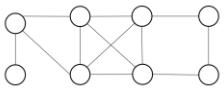
   Proposition: A graph is bipartite if, and only if, all cycles are of even size.

   $[\implies]$ If the graph is bipartite, any path alternates between each vertex of each partition to create a cycle ending by the initial vertex. Therefore, all cycles must be of even size.

   $[\impliedby]$ Consider the partition of vertices with even depth V_1, and the partition of vertices with odd depth V_2.

   Since there is no odd cycle, then, from question 2, we know that $\forall\,(u,v) \in E$ it is $depth\,(u) = depth\,(v) \pm 1$. Therefore, the graph is bipartite.

5. Propose an algorithm that allows to determine if a graph is bipartite. Test your algorithm in the following graph. Is it bipartite? Justify your answer.



The algorithm is the same as in exercise 3, because of exercise 4.

The proposed graph is clearly not bipartite, because there are several odd cycles.

6. Graph coloring is a way of coloring the vertices of a graph in such a way that no two adjacent vertices share the same color. A 2-colorable graph is a graph that can be colored with only 2 colors.

   (a) What is the link with the previous exercise? Justify your answer.

   Proposition: a graph is 2-colorable if, and only if, it is bipartite.

   Proof: [$\Longrightarrow$] If it is 2-colorable, with colors red and blue. Then we take $V_1 = \{u | color\,(u) = blue\}$ and $V_2 = \{u | color\,(u) = red\}$. G is clearly bipartite with this partition.

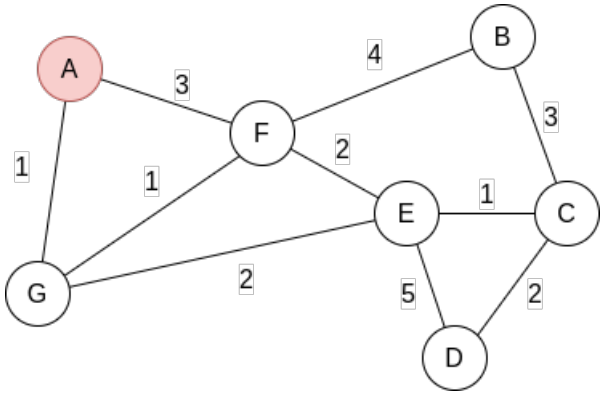
   [$\Longleftarrow$] If it is bipartite, with partition V_1 and V_2, then we can color all nodes in V_1 in blue, and all nodes in V_2 in red. The graph is 2-colorable.

   (b) We want to write an algorithm, inspired by DFS search, which takes as input a graph, $G = (V, E)$, and which returns a pair $(result, color)$ where $result$ is True if the graph is colorable, False otherwise, and $color$ is a dictionary associating a color 0 or 1 to each vertex. This algorithm should stop as soon as possible when the graph is not 2-colorable.

```
procedure coloring(G=(V,E), r)
  color <- {r: 0}
  stack <- emptyset
  push(stack, r)

  while stack is not empty do
    v <- pop(stack)
    for neig in neighbours(v) do
      if neigh is not in color.keys then
        push(stack, neigh)
        color[neig] = 1 - color[v]
      elif color[neig] = color[v] then
        return False, color
      end if
    end for
  end while
  return True, color
end procedure
```

7. Compute the shortest path in the following graph using Dijkstra's algorithm, starting at A:

```
procedure dijkstra(G=(V,E), r)
  dist <- {r:0}
  P <- emptyset

  for v in V-{r} do
    dist[v] = infinity
  end for

  while V-P is not empty do
    w <- select(v in V-P and dist[v]=min_u dist[u])
    P <- P union {w}
    for neig in neighbours(w)-P do
      if dist[w]+weight(neig,w) < dist[neig] then
        dist[neig] <- dist[w]+wight(neig,w)
      end if
    end for
  end while
end procedure
```



We start with: dist $=$

A	B	C	D	E	F	G
0	∞	∞	∞	∞	∞	∞

, and $P = \emptyset$.

Now, $w = A$ and $P = \{A\}$. We update dist $=$

A	B	C	D	E	F	G
0	∞	∞	∞	∞	3	1

.

Now, $w = G$ and $P = \{A, G\}$. We update dist $=$

A	B	C	D	E	F	G
0	∞	∞	∞	3	2	1

.

Now, $w = F$ and $P = \{A, F, G\}$. We update dist $=$

A	B	C	D	E	F	G
0	6	∞	∞	3	2	1

.

Now, $w = E$ and $P = \{A, E, F, G\}$. We update dist $=$

A	B	C	D	E	F	G
0	6	4	8	3	2	1

.

Now, $w = C$ and $P = \{A, C, E, F, G\}$. We update dist $=$

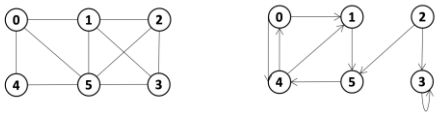
A	B	C	D	E	F	G
0	6	4	6	3	2	1

.

Now, $w = B$ and $P = \{A, B, C, E, F, G\}$. dist does not change.

Finally, $w = D$ and $P = \{A, B, C, D, E, F, G\}$. dist does not change.

8. Given the following graphs:



(a) Give the different representations of these graphs.

$$A_1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}, L_1 = \begin{array}{c} 0 : \{1, 4, 5\} \\ 1 : \{0, 2, 3, 5\} \\ 2 : \{1, 3, 5\} \\ 3 : \{1, 2, 5\} \\ 4 : \{0, 5\} \\ 5 : \{0, 1, 2, 3, 4\} \end{array}$$

$$B_1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccccccc} 01 & 04 & 05 & 12 & 13 & 15 & 23 & 25 & 35 & 45 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

$$A_2 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}, \quad L_2 = \begin{array}{c} 0 : \{1, 4\} \\ 1 : \{5\} \\ 2 : \{3, 5\} \\ 3 : \{3\} \\ 4 : \{0, 1\} \\ 5 : \{4\} \end{array},$$

$$B_1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 01 & 04 & 15 & 23 & 25 & 33 & 40 & 41 & 54 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

(b) Compute A^2, A^3. What does A^r_{ij} represents?
A^r_{ij} represents the number of paths of length r from node i to node j.

(c) What is the complexity of A^r? Is it possible to reduce it?
Computing A^r is $O\left(rn^3\right)$, since it requires r products of complexity $O\left(n^3\right)$.
However, we can reuse some results to reduce the complexity:

- If r is even, we can do $A^r = \left(A^{\frac{r}{2}}\right)^2$.
- If r is odd, we can do $A^r = A\left(A^{\frac{r-1}{2}}\right)^2$.

Therefore, we can obtain A^r in $O\left(\log r \cdot n^3\right)$.

2.2 Linear Algebra Preliminaries

A **norm** is a function f that measures the size of a vector. It must satisfy the following properties:

- $f(x) = 0 \iff x = 0$.
- Linear on scale factors:
$$f(\alpha x) = |\alpha| f(x), \forall \alpha \in \mathbb{R}.$$
- Triangle inequality:
$$f(x + y) \leq f(x) + f(y).$$

A widely use family of norms are the p-norms:

$$\|x\|_p = \sqrt[p]{\sum_i |x_i|^p},$$

with the most common one being the Euclidean norm, for $p = 2$:

$$\|x\| = \sqrt{\sum_i x_i^2}.$$

The **determinant** of a square matrix is equal to the hypervolume of the parallelotope defined by the vectors of the matrix. It is 0 if, and only if, the set of vectors is colinear.

The determinant can be used for many things:

- We can represents linear systems with matrices as $Y = AX$, and there are many methods to solve this efficiently.

- With the determinant we can compute the **characteristic polynomial** of A, whose roots are the eigenvalues of A.

Some properties of the determinant are:

- $|I| = 1$, where I is the identity matrix.
- $|A| = 0$ if A is singular (not invertible).
- $|AB| = |A|\,|B|$.
- $\left|A^T\right| = |A|$.
- $|cA| = c^n\,|A|$, where n is the dimension of A.

A square matrix, A, is **invertible** (non-singular, non-degenerate), with inverse denoetd A^{-1}, if $\exists B$ such that

$$AB = BA = I,$$

in this case, $A^{-1} = B$.

Proposition 2.1. *For a square matrix, A, the following properties are equivalent:*

- *A is invertible.*
- *All vectors in A are linearly independent.*
- $|A| \neq 0$.
- A^T *is invertible.*
- *0 is not an eigenvalue of A.*

Properties of the inverse:

- $\left(A^{-1}\right)^{-1} = A$.
- $\left(A^T\right)^{-1} = \left(A^{-1}\right)^T$.
- $(AB)^{-1} = B^{-1}A^{-1}$.
- $(cA)^{-1} = \frac{1}{c}A^{-1}$ for $c \neq 0$.
- $\left|A^{-1}\right| = \frac{1}{|A|}$.

An **eigenvector** or characteristic vector of a linear transformation, T, is a non-zero vector that changes by a escalar factor, λ, when transformed by T. That is, v is an eigenvector of the linear transformation T if

$$T(v) = \lambda v.$$

There is a direct correspondence between $n \times n$ matrices and linear transformation in the n-dimenstional vector space into itself. That is, every linear transformation T can be represented as a matrix A_T (the matrix depends on the chosen base). Therefore, we can say that A_T has an eigenvector v if

$$A_T v = \lambda v.$$

The scale factors of the eigenvectors are called **eigenvalues**.

We can find the eigenvalues by solving a polynomial function on λ called the **characteristic polynomial** of A_T:

$$(A - \lambda I)\, v = 0.$$

Now, this equation has non-zero solution if, and only if,

$$|A - \lambda I| = 0.$$

Therefore, we can compute $|A - \lambda I|$ and find all values of λ that makes it equal to 0.

Once we have the eigenvalues, we can use them to find the corresponding eigenvectors.

Example 2.4. Compute the eigenvalues and eigenvectors of $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$.

$$|A - \lambda I| = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = (2-\lambda)^2 - 1 = \lambda^2 - 4\lambda + 3.$$

This has as solutions

$$\lambda = \frac{4 \pm \sqrt{16 - 12}}{2} = \frac{4 \pm 2}{2} = 2 \pm 1.$$

Therefore, we have $\lambda_1 = 1, \lambda_2 = 3$.

To find the eigenvectors, we solve

$$Av = \lambda v \iff \begin{cases} 2x + y &= \lambda x \\ x + 2y &= \lambda y \end{cases}.$$

For $\lambda_1 = 1$, it is

$$\begin{cases} 2x + y &= x \\ x + 2y &= y \end{cases} \iff x = -y,$$

so the eigenvector associated to $\lambda_1 = 1$ is

$$v_{\lambda_1} = \begin{pmatrix} t \\ -t \end{pmatrix}.$$

For $\lambda_2 = 3$, it is

$$\begin{cases} 2x + y &= 3x \\ x + 2y &= 3y \end{cases} \iff x = y,$$

so the eigenvector associated to $\lambda_2 = 3$ is

$$v_{\lambda_2} = \begin{pmatrix} t \\ t \end{pmatrix}.$$

We call the **algebraic multiplicity**, t_i, of the eigenvalue λ_i to its multiplicity as root of the characteristic polynomial:

$$P(A) = |A - \lambda I| = (\lambda - \lambda_1)^{t_1} (\lambda - \lambda_2)^{t_2} \cdot \ldots \cdot (\lambda - \lambda_k)^{t_k}.$$

Note that A can have at most n distinct eigenvalues, although some of them may be complex.

Proposition 2.2. *If the eigenvalues of A are all different, then the corresponding eigenvectors are linearly independent.*

The **eigenspace** of an eigenvalue, λ, is the spave generated by the eigenvectors associated to λ.

The dimension of the eigenspace of λ is the **geometric multiplicity** of λ. The geometric multiplicity of an eigenvalue is, at most, its algebraic multiplicity.

Example 2.5. Let's get some eigenspaces:

$A = \begin{pmatrix} -1 & 1 & 0 \\ -4 & 3 & 0 \\ 1 & 0 & 2 \end{pmatrix}$, so

$$\begin{aligned} |A - \lambda I| &= \begin{vmatrix} -1-\lambda & 1 & 0 \\ -4 & 3-\lambda & 0 \\ 1 & 0 & 2-\lambda \end{vmatrix} = (-1-\lambda)(3-\lambda)(2-\lambda) + 4(2-\lambda) \\ &= (2-\lambda)\left[(-1-\lambda)(3-\lambda) + 4\right] = (2-\lambda)\left(-3 + \lambda - 3\lambda + \lambda^2 + 4\right) \\ &= (2-\lambda)\left(\lambda^2 - 2\lambda + 1\right) \\ &= (2-\lambda)(\lambda - 1)^2 . \end{aligned}$$

This has roots $\lambda_1 = 1$, with algebraic multiplicity 2, and $\lambda_2 = 2$, with algebraic multiplicity 1.

Now, we get the eigenvectors associated to them:

$$Av = \lambda v \iff \begin{cases} -x + y & = \lambda x \\ -4x + 3y & = \lambda y \\ x + 2z & = \lambda z \end{cases} .$$

For λ_1 this is

$$\begin{cases} -x + y & = x \\ -4x + 3y & = y \\ x + 2z & = z \end{cases} \iff \begin{cases} y & = 2x \\ -4x + 3y & = y \\ x & = -z \end{cases} ,$$

so $v_{\lambda_1} = \begin{pmatrix} t \\ 2t \\ -t \end{pmatrix}$, with dimension 1 (it could be 2).

For λ_2 this is

$$\begin{cases} -x + y & = 2x \\ -4x + 3y & = 2y \\ x + 2z & = 2z \end{cases} \iff \begin{cases} y & = 3x \\ -4x & = -y \\ x + 2z & = \lambda z \end{cases} \iff \begin{cases} x = 0 & \\ y = 0 & \\ 2z & = 2z \end{cases} ,$$

so $v_{\lambda_2} = \begin{pmatrix} 0 \\ 0 \\ t \end{pmatrix}$, with dimension 1 (it could not be differently).

$B = \begin{pmatrix} 4 & 6 & 0 \\ -3 & -5 & 0 \\ -3 & -6 & 1 \end{pmatrix}$, so

$$\begin{aligned} |B - \lambda I| &= \begin{vmatrix} 4-\lambda & 6 & 0 \\ -3 & -5-\lambda & 0 \\ -3 & -6 & 1-\lambda \end{vmatrix} = (4-\lambda)(-5-\lambda)(1-\lambda) + 18(1-\lambda) \\ &= (1-\lambda)\left[(4-\lambda)(-5-\lambda) + 18\right] = (1-\lambda)\left(-20 - 4\lambda + 5\lambda + \lambda^2 + 18\right) \\ &= (1-\lambda)\left(\lambda^2 + \lambda - 2\right) = (1-\lambda)^2(-2-\lambda) . \end{aligned}$$

This has roots $\lambda_1 = 1$, with algebraic multiplicity 2, and $\lambda_2 = -2$, with algebraic multiplicity 1.

Now, we get the eigenvectors associated to them:

$$Av = \lambda v \iff \begin{cases} 4x + 6y & = \lambda x \\ -3x - 5y & = \lambda y \\ -3x - 6y + z & = \lambda z \end{cases} .$$

For $\lambda_1 = 1$, we have

$$\begin{cases} 4x + 6y & = x \\ -3x - 5y & = y \\ -3x - 6y + z & = z \end{cases} \iff \begin{cases} x & = -2y \\ & \\ z & = z \end{cases}.$$

Therefore, the eigenspace associated to λ_1 is

$$E(\lambda_1) = \left\{ \begin{pmatrix} -2t \\ t \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ u \end{pmatrix} \right\}.$$

For $\lambda_2 = 2$, we have

$$\begin{cases} 4x + 6y & = -2x \\ -3x - 5y & = -2y \\ -3x - 6y + z & = -2z \end{cases} \iff \begin{cases} x & = -y \\ & \\ -3y + z & = -2z \end{cases} \iff \begin{cases} x & = -y \\ & \\ y & = z \end{cases}.$$

Thus, the eigenspace associated to $\lambda_2$is

$$E(\lambda_2) = \begin{pmatrix} -t \\ t \\ t \end{pmatrix}.$$

$$C = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix},$$

$$\begin{aligned} |C - \lambda I| &= \begin{vmatrix} 1-\lambda & -1 & 0 \\ -1 & 2-\lambda & 1 \\ 0 & 1 & 1-\lambda \end{vmatrix} = (1-\lambda)^2 (2-\lambda) - 2(1-\lambda) \\ &= (1-\lambda)\left[(1-\lambda)(2-\lambda) - 2\right] \\ &= (1-\lambda)\left(2 - \lambda - 2\lambda + \lambda^2 - 2\right) \\ &= (1-\lambda)\left(\lambda^2 - 3\lambda\right) \\ &= (1-\lambda)(\lambda - 3)\lambda. \end{aligned}$$

$$Cv = \lambda v \iff \begin{cases} x - y & = \lambda x \\ -x + 2y + z & = \lambda y \\ y + z & = \lambda z \end{cases}.$$

$\lambda_1 = 0$:

$$\begin{cases} x - y & = 0 \\ -x + 2y + z & = 0 \\ y + z & = 0 \end{cases} \iff \begin{cases} x = y & \\ y + z & = 0 \end{cases} \iff \begin{cases} x = y \\ y = -z \end{cases},$$

so

$$E(\lambda_1) = \begin{pmatrix} t \\ t \\ -t \end{pmatrix}.$$

$\lambda_2 = 1$:

$$\begin{cases} x - y & = x \\ -x + 2y + z & = y \\ y + z & = z \end{cases} \iff \begin{cases} y = 0 & \\ -x + z & = \\ z & = z \end{cases} \iff \begin{cases} y = 0 \\ x = z \end{cases},$$

so

$$E(\lambda_2) = \begin{pmatrix} t \\ 0 \\ t \end{pmatrix}.$$

$\lambda_3 = 3$:

$$\begin{cases} x - y & = 3x \\ -x + 2y + z & = 3y \\ y + z & = 3z \end{cases} \iff \begin{cases} y & = -2x \\ y & = 2z \end{cases},$$

so

$$E(\lambda_3) = \begin{pmatrix} -t \\ 2t \\ t \end{pmatrix}.$$

$$D = \begin{pmatrix} 1 & -1 & 4 \\ 3 & 2 & -1 \\ 2 & 1 & -1 \end{pmatrix},$$

$$\begin{aligned} |D - \lambda I| &= \begin{vmatrix} 1-\lambda & -1 & 4 \\ 3 & 2-\lambda & -1 \\ 2 & 1 & -1-\lambda \end{vmatrix} \\ &= (1-\lambda)(2-\lambda)(-1-\lambda) + 12 + 2 - 8(2-\lambda) + 1 - \lambda + 3(-1-\lambda) \\ &= \left(2 - 3\lambda + \lambda^2\right)(-1-\lambda) - 4 + 4\lambda \\ &= -2 - 2\lambda + 3\lambda + 3\lambda^2 - \lambda^2 - \lambda^3 - 4 + 4\lambda \\ &= -\lambda^3 + 2\lambda^2 + 5\lambda - 6 \end{aligned}$$

To obtain the roots, we can use Ruffini:

	-1	2	5	-6
1		-1	1	6
	-1	1	6	0

So $\lambda_1 = 1$ is a root and we have now $-\lambda^2 + \lambda + 6 = 0$, obtaining

$$\lambda = \frac{-1 \pm \sqrt{1+24}}{-2} = \frac{-1 \pm 5}{-2},$$

and we get $\lambda_2 = -2$ and $\lambda_3 = 3$.

$$Dv = \lambda v \iff \begin{cases} x - y + 4z & = \lambda x \\ 3x + 2y - z & = \lambda y \\ 2x + y - z & = \lambda z \end{cases}.$$

$\lambda_1 = 1$:

$$\begin{cases} x - y + 4z & = x \\ 3x + 2y - z & = y \\ 2x + y - z & = z \end{cases} \iff \begin{cases} y & = 4z \\ 3x + 3z & = 0 \\ 2x + 2z & = 0 \end{cases} \iff \begin{cases} y = 4z \\ x = -z \end{cases}.$$

Then, $E(\lambda_1) = \begin{pmatrix} -t \\ 4t \\ t \end{pmatrix}$.

$\lambda_2 = -2$:

$$\begin{cases} x - y + 4z & = -2x \\ 3x + 2y - z & = -2y \\ 2x + y - z & = -2z \end{cases} \iff \begin{cases} 3x - y + 4z & = 0 \\ 3x + 4y - z & = 0 \\ 2x + y + z & = 0 \end{cases} \iff \begin{cases} 5y - 5z & = 0 \\ 2x + y + z & = 0 \end{cases}$$

$$\iff \begin{cases} y = z \\ 2x + 2y & = 0 \end{cases} \iff \begin{cases} y = z \\ x = -y \end{cases}.$$

Then, $E(\lambda_2) = \begin{pmatrix} -t \\ t \\ t \end{pmatrix}$.

$\lambda_3 = 3$:

$$\begin{cases} x - y + 4z &= 3x \\ 3x + 2y - z &= 3y \\ 2x + y - z &= 3z \end{cases} \iff \begin{cases} -2x - y + 4z &= 0 \\ 3x - y - z &= 0 \\ 2x + y - 4z &= 0 \end{cases} \iff \begin{cases} -2x - y + 4z &= 0 \\ 5x - 5z &= 0 \end{cases}$$

$$\begin{cases} y = 2z \\ x = z \end{cases} .$$

Then, $E(\lambda_3) = \begin{pmatrix} t \\ 2t \\ t \end{pmatrix}$.

$$E = \begin{pmatrix} 6 & -2 & 2 \\ -2 & 3 & -1 \\ 2 & -1 & 3 \end{pmatrix},$$

$$\begin{aligned} |E - \lambda I| &= \begin{vmatrix} 6-\lambda & -2 & 2 \\ -2 & 3-\lambda & -1 \\ 2 & -1 & 3-\lambda \end{vmatrix} \\ &= (6-\lambda)(3-\lambda)^2 + 4 + 4 - 4(3-\lambda) - (6-\lambda) - 4(3-\lambda) \\ &= (6-\lambda)\left(9 - 6\lambda + \lambda^2\right) + 2 - 8(3-\lambda) + \lambda \\ &= 54 - 36\lambda + 6\lambda^2 - 9\lambda + 6\lambda^2 - \lambda^3 - 22 + 8\lambda + \lambda \\ &= -\lambda^3 + 12\lambda^2 - 36\lambda + 32. \end{aligned}$$

Again, we can use the Ruffini rule:

	-1	12	-36	32
2		-2	20	-32
	-1	10	-16	0

So $\lambda_1 = 2$ is a root, and we now have $-\lambda^2 + 10\lambda - 16 = 0$, which gives us

$$\lambda = \frac{-10 \pm \sqrt{100 - 64}}{-2} = \frac{-10 \pm 6}{-2} = 5 \pm 3.$$

Therefore, λ_1 is a double root and the other root is $\lambda_2 = 8$.

$$Ev = \lambda v \iff \begin{cases} 6x - 2y + 2z &= \lambda x \\ -2x + 3y - z &= \lambda y \\ 2x - y + 3z &= \lambda z \end{cases} .$$

$\lambda_1 = 2$:

$$\begin{cases} 6x - 2y + 2z &= 2x \\ -2x + 3y - z &= 2y \\ 2x - y + 3z &= 2z \end{cases} \iff \begin{cases} 4x - 2y + 2z &= 0 \\ -2x + y - z &= 0 \\ 2x - y + z &= 0 \end{cases} \iff \begin{cases} 4x - 2y + 2z &= 0 \\ 2x - y + z &= 0 \end{cases}$$

$$\iff 2x - y + z = 0$$

If $x = 0$: $y = z$.

If $x = t \neq 0$: $-y + z = -2t$, working for $y = t$ and $z = -t$.

So

$$E(\lambda_1) = \left\{ \begin{pmatrix} 0 \\ t \\ t \end{pmatrix}, \begin{pmatrix} t \\ t \\ -t \end{pmatrix} \right\}.$$

$\lambda_2 = 8$:

$$\begin{cases} 6x - 2y + 2z &= 8x \\ -2x + 3y - z &= 8y \\ 2x - y + 3z &= 8z \end{cases} \iff \begin{cases} -2x - 2y + 2z &= 0 \\ -2x - 5y - z &= 0 \\ 2x - y - 5z &= 0 \end{cases} \iff \begin{cases} -3y - 3z &= 0 \\ 2x - y - 5z &= 0 \end{cases}$$

$$\iff \begin{cases} y = -z \\ 2x - 4z &= 0 \end{cases} \iff \begin{cases} y = -z \\ x = 2z \end{cases}.$$

Therefore,

$$E(\lambda_2) = \begin{pmatrix} 2t \\ -t \\ t \end{pmatrix}.$$

$$F = \begin{pmatrix} 0 & -1 & -1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix},$$

$$\begin{aligned} |F - \lambda I| &= \begin{vmatrix} -\lambda & -1 & -1 \\ 1 & 2-\lambda & 1 \\ 1 & 1 & 2-\lambda \end{vmatrix} \\ &= -\lambda(2-\lambda)^2 \cancel{-2+2-\lambda+\lambda} + 2 - \lambda \\ &= (2-\lambda)\left[-\lambda(2-\lambda)+1\right] \\ &= (2-\lambda)\left(-2\lambda + \lambda^2 + 1\right) \\ &= (2-\lambda)(1-\lambda)^2. \end{aligned}$$

One root is $\lambda_1 = 1$ with algebraic dimension 2, and $\lambda_2 = 2$ with algebraic dimension 1.

$$Fv = \lambda v \iff \begin{cases} -y - z &= \lambda x \\ x + 2y + z &= \lambda y \\ x + y + 2z &= \lambda z \end{cases}.$$

$\lambda_1 = 1$:

$$\begin{cases} -y - z &= x \\ x + 2y + z &= y \\ x + y + 2z &= z \end{cases} \iff \begin{cases} x + y + z &= 0 \end{cases}$$

If $x = 0$: $y = -z$.

If $x = t \neq 0$: $y + z = -t$. This works for $y = t, z = -2t$.

Therefore,

$$E(\lambda_1) = \left\{ \begin{pmatrix} 0 \\ t \\ -t \end{pmatrix}, \begin{pmatrix} t \\ t \\ -2t \end{pmatrix} \right\}.$$

$\lambda_2 = 2$:

$$\begin{cases} -y - z &= 2x \\ x + 2y + z &= 2y \\ x + y + 2z &= 2z \end{cases} \iff \begin{cases} -y - z &= 2x \\ x + z &= 0 \\ x + y &= 0 \end{cases} \iff \begin{cases} x = -z \\ x = -y \end{cases}.$$

$$\iff 2x - y + z = 0$$

So

$$E(\lambda_2) = \begin{pmatrix} t \\ -t \\ -t \end{pmatrix}.$$

Another way to represent eigenvalues and eigenvectors is

$$AV = V\Lambda,$$

where $V = [v_1, ..., v_n]$ is the matrix formed by putting each eigenvector as a column, and

$$\Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$$

is the diagonal matrix formed by all eigenvalues.

A matrix A is **diagonalizable** if there exist n linearly independent eigenvectors. That is, if the matrix V is invertible:

$$\Lambda = V^{-1}AV.$$

This leads naturally to the **eigen-decomposition** of the matrix,

$$A = V\Lambda V^{-1}.$$

A real matrix, U, is **orthogonal** if $U^TU = UU^T = I$.

Proposition 2.3. *The following statements are equivalent:*

- U^T *is orthogonal.*
- $U^T = U^{-1}$.
- $|U| = 1$.
- U*'s eigenvectors are orthonormal (the pairwise dot product is 0 and the norm is 1).*

Example 2.6. Some examples of orthogonal matrices:

- Identity: I
- Permutation of coordinates: $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- Rotation: $\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$.
- Reflection: $\begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix}$.

A matrix A is said to be **positive semi-definite** when it can be obtained as the product of a matrix by its transpose:

$$\exists X | A = XX^T.$$

Positive semi-definite matrices are always symmetric, because

$$A^T = \left(XX^T\right)^T = XX^T = A.$$

A symmetric matrix A is positive semi-definite if all its eigenvalues are non-negative.

Proposition 2.4. *Let A be a positive semi-definite matrix. Then:*

- $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ *and its eigenvectors are pairwise orthogonal when their eigenvalues are different.*
- *The eigenvalues are composed of real values.*
- *The multiplicity of an eigenvalue is the dimension of its eigenspace.*

In this case, since eigenvectors are orthogonal, it is possible to store all the eigenvectors in an orthogonal matrix.

Therefore, the eigen-decomposition of a positive semi-definite matrix, A, could be

$$A = U\Lambda U^T,$$

with U an orthogonal matrix.

As a consequence, the eigen-decomposition of a positive semi-definite matrix is often referred to as its diagonalization.

An alternative definition for positive semi-definite matrix is:

A is positive semi-definite if $x^T Ax \geq 0, \forall x$.

If it is $x^T Ax > 0, \forall x$, then it is positive definite.

If it is $x^T Ax \leq 0, \forall x$, then it is negative semi-definite.

If it is $x^T Ax < 0, \forall x$, then it is negative definite.

The **rank** of a matrix is the dimension of the vector space generated by its columns (or rows). This corresponds to the maximum number of linearly independent columns of A. A matrix whose rank is equal to its size is called a **full rank matrix**. Only full rank matrices have an inverse.

Proposition 2.5. *The sum of the eigenvalues of a matrix is the sum of the elements of its main diagonal. The product of the eigenvalues is equal to the determinant of the matrix.*

We can now define the **Laplacian matrix** for undirected graphs, as

$$L_{ij} = \begin{cases} -1 & , (v_i, v_j) \in E \\ 0 & , (v_i, v_j) \notin E \\ d_i & , i = j \end{cases},$$

or, equivalently,

$$L = D - A,$$

where A is the degree is the matrix of G, and A its adjacency matrix.

2.2.1 Exercises

1. What could you say about these matrices?
   (a) $A = \begin{pmatrix} -1 & \frac{3}{2} \\ 1 & -1 \end{pmatrix}$, $det\,(A) = -\frac{1}{2}$, A is invertible. Its eigenvalues are $\lambda_1 = -1+\frac{\sqrt{6}}{2}$ and $\lambda_2 = -1-\frac{\sqrt{6}}{2}$, with $v_{\lambda_1} = \begin{pmatrix} \frac{\sqrt{6}}{2}t \\ t \end{pmatrix}$ and $v_{\lambda_2} = \begin{pmatrix} -\frac{\sqrt{6}}{2}t \\ t \end{pmatrix}$.
   (b) $B = \begin{pmatrix} -1 & \frac{3}{2} \\ \frac{2}{3} & -1 \end{pmatrix}$. The second row is equal to the first row multiplied by $-\frac{2}{3}$. Therefore, it is not invertible.

(c) I: its determinant is 1. It is symmetric, orthogonal, its own inverse. Triple eigenvalue 1, with eigenspace the whole space.

2. Show that $A^n = X\Lambda X^{-1}$.

   First, this is only true if A is diagonalizable. If that is the case, then we can proceed by induction on n:

   $n = 1$: Obvious.

   $n = 2$:

   $$A^2 = \left(X\Lambda X^{-1}\right)^2 = X\Lambda X^{-1} X\Lambda X^{-1} = X\Lambda^2 X^{-1}.$$

   Suppose it is true for $n-1$:

   $$A^{n-1} = X\Lambda^{n-1} X^{-1}.$$

   Then, for n, we have:

   $$A^n = AA^{n-1} = X\Lambda X^{-1} X\Lambda^{n-1} X^{-1} = X\Lambda^n X^{-1}.$$

3. Find the eigenvalues and unit eigenvectors of A^TA and AA^T with $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ the Fibonnaci matrix.

   First of all, notice that A is symmetric, so $A^TA = AA^T = A^2 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$.

   $\begin{vmatrix} 2-\lambda & 1 \\ 1 & 1-\lambda \end{vmatrix} = (2-\lambda)(1-\lambda) - 1 = 2 - 3\lambda + \lambda^2 - 1 = \lambda^2 - 3\lambda + 1$. The roots of this polynomial are

   $$\lambda = \frac{3 \pm \sqrt{9-4}}{2} = \frac{3 \pm \sqrt{5}}{2}.$$

   Now,

   $$A^2 v = \lambda v \iff \begin{cases} 2x + y &= \lambda x \\ x + y &= \lambda y \end{cases} \iff \left\{ x = (\lambda - 1)\, y \right.$$

   Therefore

   $$E(\lambda_1) = \begin{pmatrix} \frac{1+\sqrt{5}}{2} t \\ t \end{pmatrix}$$

   with unit eigenvector $v_1 = \frac{1}{\sqrt{4-\sqrt{5}}} \begin{pmatrix} \frac{1+\sqrt{5}}{2} \\ 1 \end{pmatrix}$.

   And

   $$E(\lambda_2) = \begin{pmatrix} \frac{1-\sqrt{5}}{2} t \\ t \end{pmatrix}$$

   with unit eigenvector $v_2 = \frac{1}{\sqrt{4-\sqrt{5}}} \begin{pmatrix} \frac{1-\sqrt{5}}{2} \\ 1 \end{pmatrix}$.

4. Without multiplying

   $$S = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 5 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix},$$

   find the determinant, the eigenvalues and eigenvectors. Why S is positive definite?

   We have $S = U\Lambda U^T$ with U orthogonal. Therefore, the eigenvalues of S are 2 and 5. Its determinant is 10. The eigenvectors are the eigenvectors of Λ rotated as well, that is:

   $$V = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}.$$

   S is positive definite because

   $$xSx^T = x \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 5 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} x^T,$$

now note that

$$\left(x \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\right)^T = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} x^T,$$

so

$$xSx^T = y \begin{pmatrix} 2 & 0 \\ 0 & 5 \end{pmatrix} y^T \geq 0,$$

because $\begin{pmatrix} 2 & 0 \\ 0 & 5 \end{pmatrix}$ is positive semi-definite (symmetric with positive eigenvalues).

5. For what numbers c and d are the following matrices positive definite?

   (a) $A = \begin{pmatrix} c & 1 & 1 \\ 1 & c & 1 \\ 1 & 1 & c \end{pmatrix}$: all principal minors must be positive. That is:

   - $c > 0$.
   - $\begin{vmatrix} c & 1 \\ 1 & c \end{vmatrix} = c^2 - 1 > 0$. Combined with the previous one, this is $c > 1$.
   - $\begin{vmatrix} c & 1 & 1 \\ 1 & c & 1 \\ 1 & 1 & c \end{vmatrix} = c^3 + 2 - 3c$. Roots: 1, $\begin{array}{c|cccc} 1 & 1 & 0 & -3 & 2 \\ & & 1 & 1 & -2 \\ \hline & 1 & 1 & -2 & 0 \end{array}$, and we have $c^2 + c - 2$, with roots $c = \frac{-1 \pm \sqrt{5}}{2}$. We are only interested in the interval $(1, \infty)$, in which $c^3 - 3c + 2 > 0$.

   Therefore, it is $c > 1$.

   (b) $B = \begin{pmatrix} 1 & 2 & 3 \\ 2 & d & 4 \\ 3 & 4 & 5 \end{pmatrix}$:

   - $1 > 0$.
   - $\begin{vmatrix} 1 & 2 \\ 2 & d \end{vmatrix} = d - 4 > 0 \iff d > 4$.
   - $\begin{vmatrix} 1 & 2 & 3 \\ 2 & d & 4 \\ 3 & 4 & 5 \end{vmatrix} = 5d + 24 + 24 - 9d - 16 - 20 = -4d + 12 > 0 \iff -4d > -12 \iff d < 3$.

   Therefore, there is no value for d for which B is positive.

3 Random Walks on Graphs

References

[1] NacÃ©ra Seghouani. Massive graph management and analytics. Lecture Notes.