1^{er} Cours : Introduction
MPRI 2015–2016

Michel Habib
habib@liafa.univ-Paris-Diderot.fr
http://www.liafa.univ-Paris-Diderot.fr/~habib

14 septembre 2015

Liafa

Schedule of this introduction course

Environment of research

Graph Representations

Geometric representations of particular classes of graphs

A set of important exercises

Elimination schemes

Academic year : 2015-2016

- Research team : Distributed algorithms and graphs, LIAFA :
- Part I
  Practical graph algorithms (Classical Algorithmic)
  Michel Habib (Pr Univ. Paris Diderot)
- Part II
  Treewith
  Structure theorems (minors theory and graph decompositions Robertson and Seymour)
  Pierre Charbit (MdC, Univ. Paris Diderot)
- contact : Prenom.Nom@liafa.univ-Paris-Diderot.fr

Organization of this first half

- There will be an extra course half-way : solution of exercises in order to be more familiar with the objects we play with
- All slides will be available and many papers.
- But listening to the course is not enough. Each course has to be worked at home (joint papers read carefully)
- Research internship subjects are on demand.

Application Themes

- Bioinformatics (originally phylogeny and now other graph problems such as computation of networks parameters)
- Networks and Distributed systems
- Analysis of huge graphs in social sciences (ranking, clustering or community detection)

On-going collaborations

GANG

An INRIA–LIAFA project on graphs an networks, dir. Laurent Viennot

SAE

Systématique, Adaptation, Evolution : A biological group at UPMC with Eric Bapteste and Philippe Lopez
An interesting project of forced evolution on lezards

Social sciences

ANR Project AlgoPol (Politics of Algorithms), a collaboration with Dominique Cardon (Orange Lab) and Christophe Prieur (LIAFA)

Theoretical side

Why such a course ?

Fagin's theorems in descriptive complexity

Characterizations of P and NP using graphs and logics fragments.

NP

The class of all graph-theoretic properties expressible in existential second-order logic is precisely NP.

P

The class of all graph-theoretic properties expressible in Horn existential second-order logic with successor is precisely P.

- The previous theorems belong to descriptive complexity :
- Descriptive complexity is a branch of computational complexity theory and of finite model theory that characterizes complexity classes by the type of logic needed to express the languages in them.
- Famous Courcelle's theorem is of this kind coupling monadic second order logic with linear algorithms on some graphs problems. (cf. Pierre Charbit, second half of this course)

Practical issues

1. Graphs made up with vertices and edges (or arcs) provide a very powerful tool to model real-life problems. Weights on vertices or edges can be added.
2. Many applications involve graph algorithms, in particular many facets of computer science !
3. Also many new leading economical applications Google PageRank and FaceBook are graph based The Facebook search engine is called GraphSearch !)
4. Analysis of Twitter sequences

Some applications

1. Shortest paths computations in Google maps
2. Community detections in Big Data (NSA, CIA, FBI ... but also Google, Facebook, Carrefour, ...)
3. (Un)Fortunately this course provides tools to solve these two practical problems

Program of the first half

1. Introduction, Elimination schemes, cographs
2. Diameter computations a problem almost solved
3. A theory of graph searches
4. Chordal graphs and phylogeny
5. Cographs and modular decomposition
6. Commun intervals
7. Graphs classes and geometric representations
8. Graphs and orders (co)-comparability graphs
9. Important graph parameters (treewidth, pathwidth ...)

Typical questions

- Recognition algorithms, if possible linear-time and **certifying**.
- Efficient decomposition algorithms
- Computations of compact encodings and representation
- Random generation and enumeration
- Routing protocols and diameter estimation
- Computation of some invariant (for example min coloring or max clique).

Goals

- Reach a good level of knowledge on graph algorithms (many students FAQ)
- Understand the greedy algorithms
- Try to understand why some simple heuristics works for most practical data
- Understand how to use structural results on graphs to design algorithms (gap between graph theory and graph algorithms)
- Panorama of research

Graph invariants, graph parameters

Definition

A graph invariant is a function $G \to f(G) \in N$, which is invariant under isomorphism (i.e. if G and H are isomorphic then $f(G) = f(H)$).

2 important graph parameters

Maximum size of a clique in G : $\omega(G)$.
Coloration of the vertices of G with a minimum number of colors : $\chi(G)$
Trivially : $\omega(G) \leq \chi(G)$

Notations

Here we deal with finite loopless and simple undirected graphs.
For such a graph G
we denote by $V(G)$ the set of its vertices
and by $E(G)$ the edge set
By convention $|V| = n$ and $|E| = m$

Bound on the number of edges

Triangle free graphs (Turan's theorem)

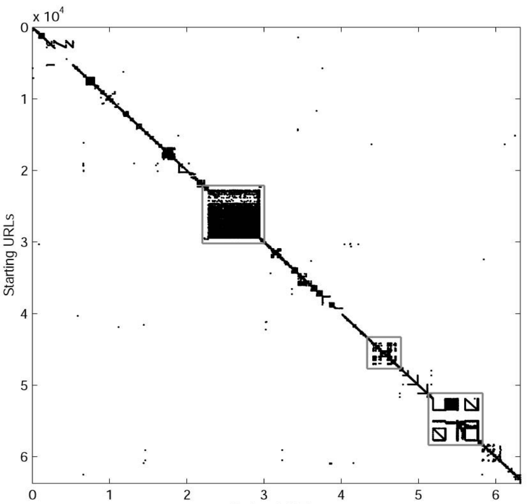
Show that if G has no triangle then :
$|E| \leq \frac{|V|^2}{4}$

Planar graphs

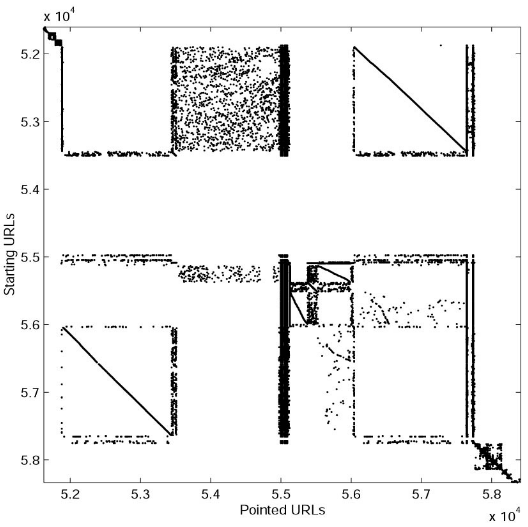
A graph is planar if it admits a non-crossing planar embedding in the plane.
Show that if G is a simple planar graph (i.e. without loop and parallel edge) then :
$|E| \leq 3|V| - 6$

Sparse graphs satisfy : $|E(G)| \in O(|V(G)|)$
Planar graphs are sparse, but also many graphs coming from applications are sparse.
For example the WEB graph is sparse.
For sparse graphs one has to use an adjacency lists representation.

Matrice ordonnée par l'ordre alphabétique des noms des URL



Un zoom autour de la diagonale



- Implicit hypothesis : the memory words have k bits with $k > \lceil log(|V|) \rceil$
- To be sure, consider the bit encoding level

- Adjacency lists
  $O(|V| + |E|)$ memory words
  Adjacency test : xy is an arc in $O(|N(x)|)$
- Basic one :
  The number of vertices and a list of edges. This often the format in which you cand find graphs in graph databases.
- Customized representations, a pointer for each arc ...
- All these representations are linearly equivalent.

Adjacency Matrix

Adjacency Matrix
$O(|V|^2)$ memory words (can be compressed)
Adjacency test : xy is an arc in $O(1)$

Comment

This representation is not linearly equivalent to the previous one (adjacency lists)

For some large graphs, the Adjacency matrix, is not easy to obtain and manipulate.
But the neighbourhood of a given vertex can be obtained. (WEB Graph or graphs is Game Theory)

Quicksands

- A sentence like :
  "To compute this invariant or this property of a given graph G one needs to "see"(or visit) every edge at least once".
- False statement as for example the computation of twins resp. connected components on $\overline{G}$ knowing G.

Exercise

Can the advantages of the 2 previous representations can be mixed in a unique new one ?
Adjacency lists : construction in $O(n+m)$
Incidence matrix : cost of the query : $xy \in E$? in $O(1)$

In other words

Using $O(n^2)$ space, but with linear time algorithms on graphs ?

Which graphs ?

- `http://snap.stanford.edu/data/`
  A common data basis of various graphs
  Format = List of edges
- From Social Networks or Biology
  Very hard or expensive to collect
- From Google (Web or geolocalisation spying ...)

Goals : to deal with very large graphs coming form :

- Humain brain 10^{23} vertices
- Dynamic of social networks
- Hot subject
  Classify huge graph families in order to predict events or to measure properties
  (level of democracy via the analysis of a social network)
- For each graph compute a descriptor i.e. a boolean vector of properties, and then apply some machine learning method.

Interval graphs

Definition

A graph G is an interval graph if there exists a set of closed intervals $\mathcal{I}$ of the real line such that :
there exists a bijection $f : V(G) \rightarrow \mathcal{I}$ such that :
$xy \in E(G)$ iff $f(x) \cap f(y)$

Uses of geometric representations

- Computing a maximum clique (of maximum size) using an interval embedding of an interval graph.
  First idea a greedy method :
  Just consider a set of successive vertical lines and compute their intersection with the intervals. In fact it suffices to study the $2n$ vertical lines corresponding to the end vertices of the intervals.

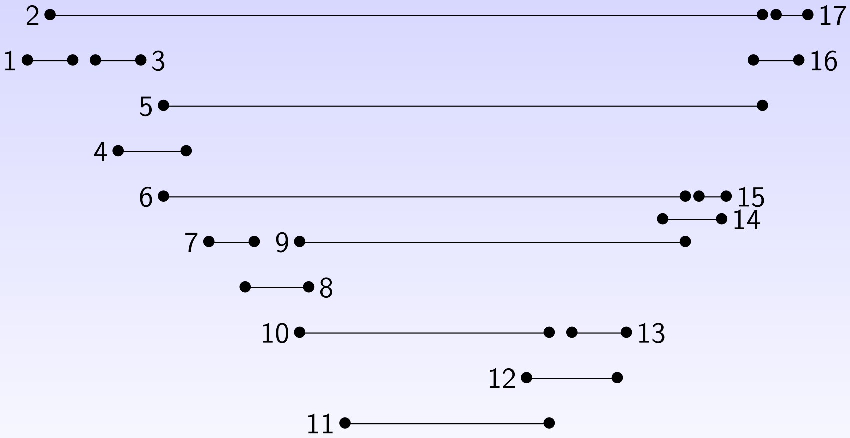


FIGURE: Ma's interval graph of order 17

Now consider a clique of maximum size of G, and their corresponding intervals.
They pairwise intersect.
We can prove by induction that they all intersect. (Helly property)

- True for 2, idem for 3 intervals
- Induction on $|\mathcal{I}|$.

Induction step

- Consider the n-1 first intervals $I_1, \ldots I_{n-1}$ by induction there exists a non empty set of points J_1 contained in all these intervals.
- Consider the n-1 last intervals $I_2, \ldots I_n$ by induction there exists a non empty set of points J_2 contained in all these intervals.
- If $J_1 \cap J_2 \neq \emptyset$, we have finished. Else we may state that J_1 is left to J_2.
  Since I_1 and I_n intersect in some point x. Three cases :
  1. x is in between J_1 and J_2.
     But $I_2, \ldots I_{n-1}$ contains J_1 and J_2 and therefore contain also x.
  2. x is left to J_1.
     But then I_n containing x and J_2 must intersects J_1 as all the intervals.
  3. x is right to J_2. Similarly all the intervals intersect J_2.

Helly Property

Definition

A ground set V, and a subset family $\{T_i\}_{i \in I}$ ($\forall i, T_i \subseteq V$)
The family satisfies Helly property if
$\forall J \subseteq I$ et $\forall i, j \in J$ $T_i \cap T_j \neq \emptyset$ implies $\cap_{i \in J} T_i \neq \emptyset$

We just proved

Theorem

The maximal cliques of an interval graph satisfy the Helly property.

Lines

Non-parallel lines in the plane, do not satisfy Helly property.

Greedy scan

Computing a maximum clique (of maximum size) using an interval embedding of an interval graph.
Just consider a set of successive $2n$ vertical lines corresponding to the end vertices of the intervals and compute their intersection with the intervals. Keep the maximum size of an intersection.

Theorem

The greedy scan finds a clique of maximum size in linear time.

Proof of the greddy scan algorithm

1. Each intersection found by the greedy scan generates a clique of the graph not necessarily maximal.
2. Furthermore using the Helly property any maximal clique $\mathcal{C}$ has a point in the line x. Consider $left(x)$ the rightmost left end of an interval of $\mathcal{C}$ finishing after x and $right(x)$ the leftmost right end of an interval of $\mathcal{C}$ that starts before x.
3. No interval of $\mathcal{C}$ starts in $[left(x), x]$ and similarly no interval of $\mathcal{C}$ ends in $[x, right(x)]$.
4. Therefore $left(x), x, right(x)$ have the same intersections with the intervals.

Consequences

As a consequence of previous theorem :

- An interval graph has at most $2n$ different maximal cliques. (Could be exponential for some graphs).
- Exercise
  Compute a minimum coloring using an interval embedding of an interval graph.

More applications of Interval graphs

Let us consider the following Operation Research (OR) problem :

- Storage of products in fridges : each product is given with an interval of admissible temperatures.
  Find the minimum number of fridges needed to store all the products (a fridge is at a given temperature).
- A solution is given by computing a minimum partition into maximal cliques.
- Fortunately for an interval graph, this can be polynomially computed
- So knowing that a graph is an interval graph can help to solve a problem.

Interval graphs are used to modelize time (in scheduling) but also to analyze DNA sequences.

Boxicity dimension

A generalization of the fridge problem with many parameters : temperature, hygrometry . . .
Could also be viewed as a biological problem :
To each species on can associate the k principal parameters needed for its survival. Each parameter has a range of possible values.
For human beings these parameters could be : temperature, air pression, air quality (% of CO (carbon monoxide), . . .).
To each species one can associate a box in the k-dimensional space. Therefore two species may have lived together iff their boxes intersect.

Other geometric graphs

- Intersection graph of rectangles (resp. squares) in the plane (boxity 2 graphs)
- Intersection graph of hexagons (resp. circles) in the plane. Used for modelling GSM antennas and frequency assigment problems.
- Intersection graph of pairs of intervals (biology from DNA to ARN)
- Intersection graph of segments between two parallel lines (permutation graphs)
- Numerous algorithms on planar graphs use the existence of a dual graph. (Ex : flows transform into paths)
- . . .

Geometric representations

Let $\mathcal{I}$ be a set of intervals of the real line Δ, we define an infinite Hypergraph $H = (\Delta, \mathcal{I})$ as follows : the vertices are the points of Δ and the edges are simply the intervals.

Gallai's theorem 1962

$\tau(H) = \nu(H)$
$\tau(H)$ is the minimum cardinal of a set of points covering the edge set,
et $\nu(H)$ is the maximum cardinal of a matching, i.e. a set of edges pairwise disjoint.

First remark

Obviously : $\tau(H) \geq \nu(H)$.

An algorithmic proof

Repeat until $\mathcal{I}$ is empty
Find the p_i leftmost right end of an interval E_i in $\mathcal{I}$;
Delete in $\mathcal{I}$ the intervals that contain p_i ;

End of the proof

The set of selected points p_i clearly cover $\mathcal{I}$ by construction.
The intervals E_i provide a matching of $\mathcal{I}$. Suppose $E_i \cap E_j \neq \emptyset$ with $i < j$ but
when p_i has been considered we should have removed E_J, a contradiction.
Moreover they have the same cardinal which yields the optimality of the two parameters using the previous remark.

Exercises

X a finite set
$\mathcal{F}$ a family of subsets $S_1, \ldots, S_k$ de X.

A-1 Write an algorithm to check if there is no i, j such that : $S_i \subseteq S_j$

A-2 **Disjoint Set Problem :**
Find i, j such that $S_i \cap S_j = \emptyset$

A-3 Check if : $\forall i, j,\ i \neq j\ S_i \cap S_j = \emptyset$

A-4 Check if there exists i such that : S_i intersects all other subsets of $\mathcal{F}$

A-5 Compute the set of maximal (resp. minimal) for inclusion of $\mathcal{F}$

A-6 Check if $\mathcal{F}$ is laminar.

A-7 Check if $\mathcal{F}$ satisfies the Helly property

Exercises

G is a finite graph

B-1 Find if there is a triangle in G

B-2 Find if there is a 3-independent set in G

B-3 Find if there is an asteroidal triple in G

B-4 Compute the diameter of G

B-5 Compute a (resp. all) center(s) in G

Exercises on directed graphs

G is a finite directed acyclic graph

C-1 Compute the transitive closure of G

C-2 Compute the transitive reduction of G

Exercises related problems

D-1 Exact 3-sum :
3 sets A, B, C of integers
Question : $\exists$ a triple (a, b, c) with $a \in A$, $b \in B$, $c \in C$ such that $a + b + c = 0$?

D-2 Boolean matrix multplication

D-3 SAT

- Of course the idea is to find the best algorithm for each of these problems or to obtain a non trivial lower bound.
- I have already met each of these problems when dealing with graph algorithms, mainly when X is the vertex set of the graph and $\mathcal{F}$ is the family obtained by considering the vertex neighbourhoods.
- Hint : You could first search the relationships between these problems

Another algorithmic exercise

- Which is the best algorithm to merge k sorted lists ?
- What is the best complexity ?
- Is there a lower bound ?

For social sciences applications :

- Compute graph parameters such as betweenness centrality in order to discover structures in social networks.
- A typical problem :
  How to compute : for every x, y $\frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|}$
- Can we compute $|N(x) \cap N(y)|$ in the size of the smallest ?
- Even with some preprocessing.

Notations

- $G = (V, E)$ undirected graph, $|V| = n$, $|E| = m$.
- $N(v)$ strict neighborhood of v in G.
- $v_1, \ldots, v_n$: a given ordering of vertices of G.
- $V_i = \{v_i, v_{i+1}, \ldots, v_n\}$.
- $G_i = G[V_i \cap N(v_i)]$.

Definition

- P property.
- Ordering $v_1, \ldots, v_n$ elimination scheme with respect to P if $\forall i$, $P(v_i)$ true on G_i.

First example

If $G = (V, E)$ is a simple (i.e. no loop and no multiple edges) connected planar graph with more than 3 vertices, then :

- $\exists v \in V$, degree$(v) \leq 5$.
  (For a simple connected planar graph $2m = \Sigma_f degre(f) \leq 3f$ thus $m \leq 3/2f$.
  Using Euler theorem : $n - m + f = 2$ we obtain $m \leq 3n - 6$.
  But $\forall x$, $d(x) \geq 6$ implies $2m = \Sigma_x degre(x) > 6n$, i.e., $m > 3n$ a contradiction.)
- Any induced subgraph of a planar graph is planar.

First example

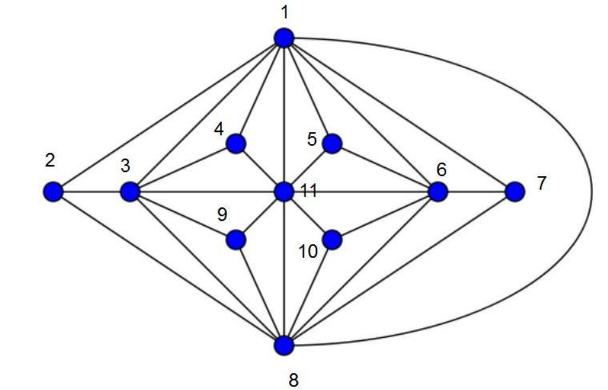
- P property

  $\rightarrow P(v)$: "$\exists v \in V$ with degree ≤ 5"

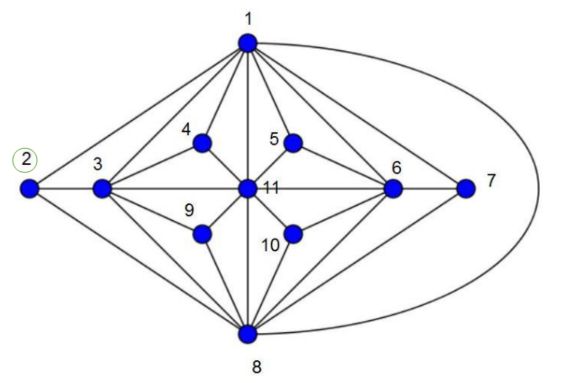
- $v_1, \ldots, v_n$ elimination scheme with respect to P if $\forall i$, $P(v_i)$ true on G_i.

  $\rightarrow$ build inductively elimination scheme w.r.t P by deleting vertex degree ≤ 5

First example

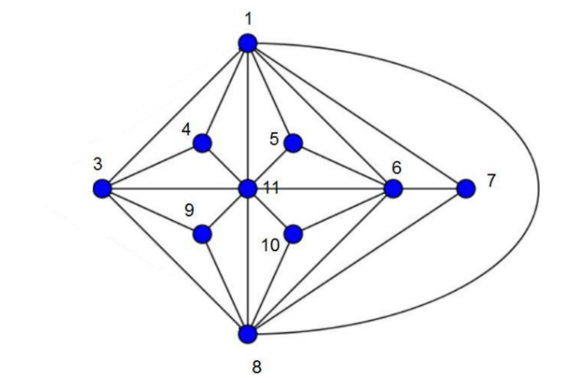


First example



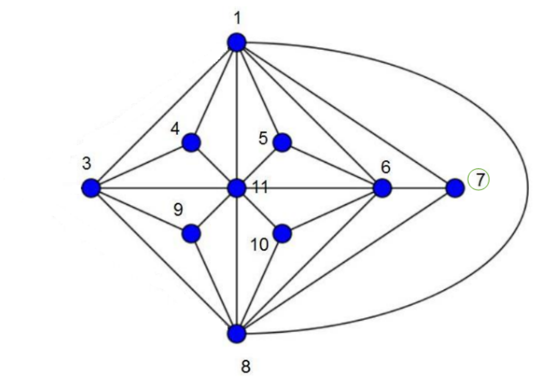
2

First example



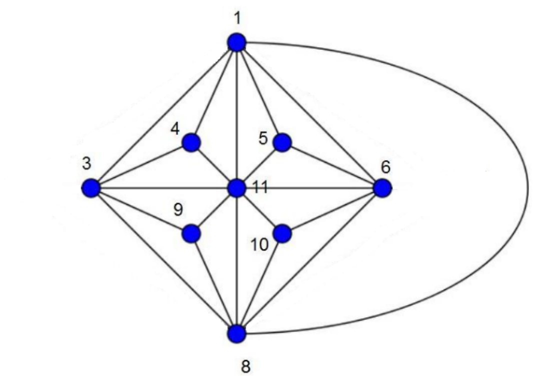
2

First example



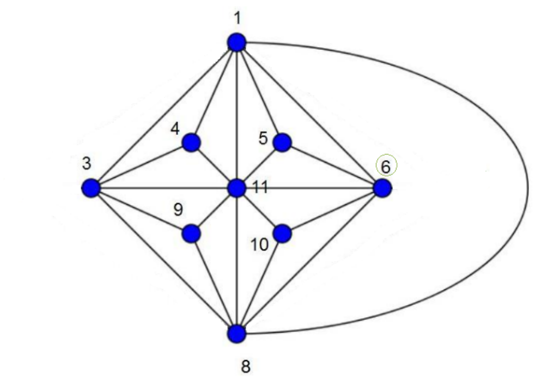
$2, 7$

First example



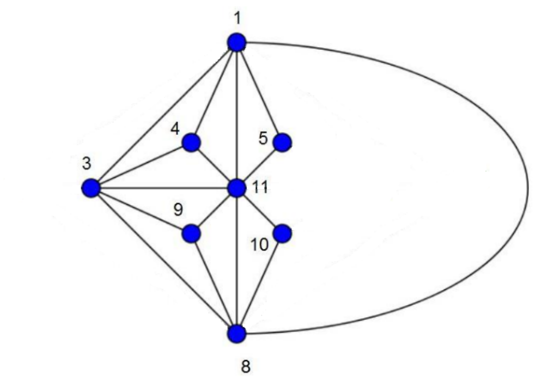
$2, 7$

First example



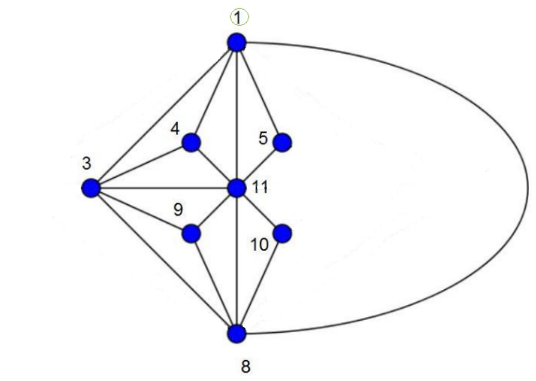
$2, 7, 6$

First example



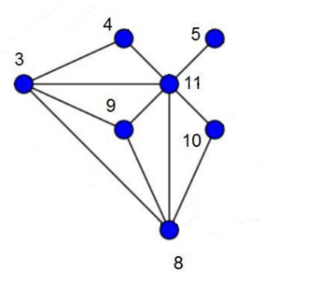
$2, 7, 6$

First example



$2, 7, 6, 1$

First example



2, 7, 6, 1

.

.

Finally : 2, 7, 6, 1, 8, 3, 4, 5, 9, 10, 11

Consequences

1. Every planar graph can be encoded using $5 logn$ bits per vertex, which is a very efficient encoding for a graph.
2. But we used Euler Theorem !
3. This is why geometric graph representations are of great importance. Since we can use geometric, topological tools for designing our algorithms.