Shortest Path

- Generalize distance to weighted setting
- Digraph $G = (V,E)$ with weight function $W: E \rightarrow R$ (assigning real values to edges)
- Weight of path $p = v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k$ is

$$w(p) = \sum_{i=1}^{k-1} w(v_i, v_{i+1})$$

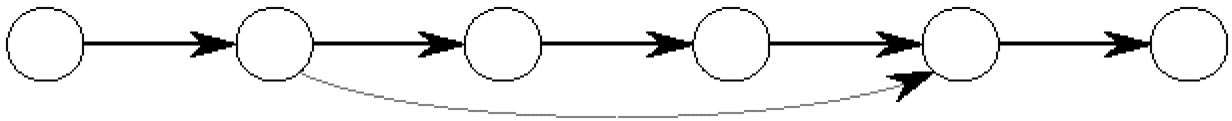
- Shortest path = a path of the minimum weight
- Applications
  - static/dynamic network routing
  - robot motion planning
  - map/route generation in traffic

Shortest-Path Problems

- Shortest-Path problems
  - **Single-source (single-destination).** Find a shortest path from a given source (vertex *s*) to each of the vertices. The topic of this lecture.
  - **Single-pair.** Given two vertices, find a shortest path between them. Solution to single-source problem solves this problem efficiently, too.
  - **All-pairs.** Find shortest-paths for every pair of vertices.
  - Unweighted shortest-paths.

Optimal Substructure

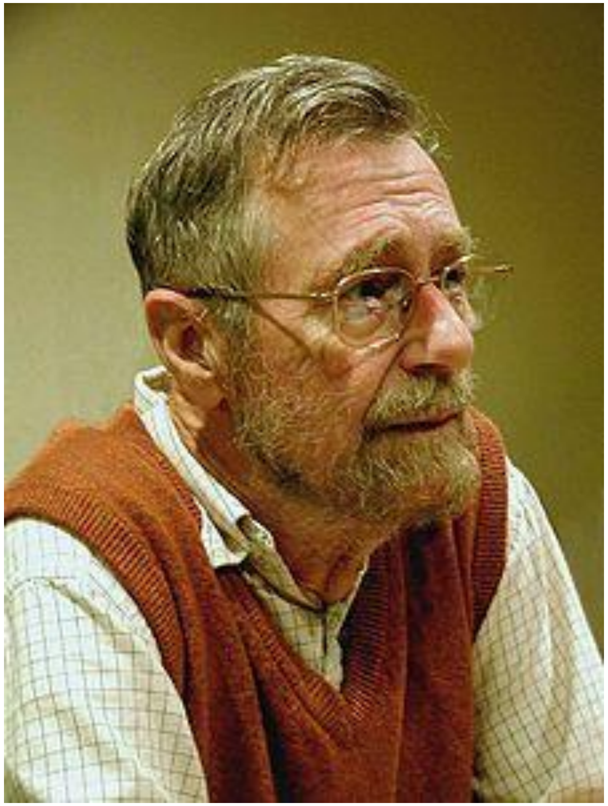
- *Theorem*: subpaths of shortest paths are shortest paths
- Proof ("cut and paste")
  - if some subpath were not the shortest path, one could substitute the shorter subpath and create a shorter total path



Negative Weights and Cycles?

- Negative edges are OK, as long as there are no *negative weight cycles* (otherwise paths with arbitrary small "lengths" would be possible)
- Shortest-paths can have no cycles (otherwise we could improve them by removing cycles)
  - Any shortest-path in graph G can be no longer than $n - 1$ edges, where n is the number of vertices

Edsger Wybe Dijkstra



Dijkstra’s ALgorithm

Solution to **Single-source (single-destination).**

Dijkstra's algorithm - is a solution to the single-source shortest path problem in graph theory.

Works on both directed and undirected graphs. However, all edges must have nonnegative weights.

Approach: Greedy

Input: Weighted graph G={E,V} and source vertex $v \in V$, such that all edge weights are nonnegative

Output: Lengths of shortest paths (or the shortest paths themselves) from a given source vertex $v \in V$ to all other vertices

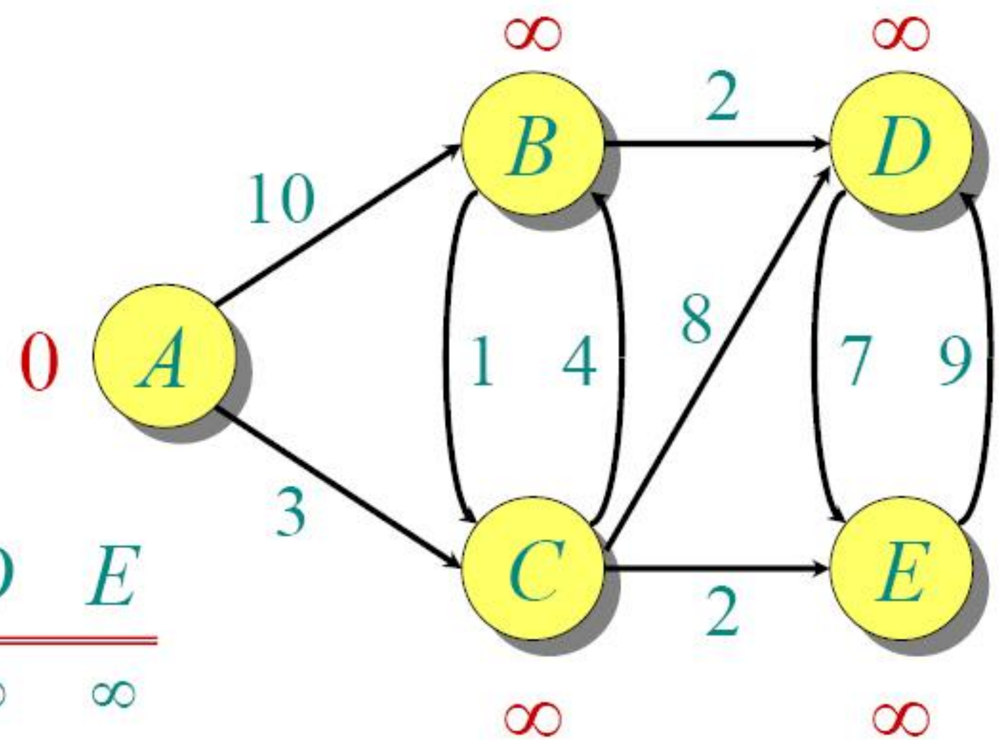
Dijkstra's Example

```
dist[s] ←0                                  (distance to source vertex is zero)
for  all v ∈ V–{s}
      do  dist[v] ←∞                        (set all other distances to infinity)
S←∅                                         (S, the set of visited vertices is initially empty)
Q←V                                         (Q, the queue initially contains all
vertices)
while Q ≠∅                                  (while the queue is not empty)
do   u ← mindistance(Q,dist)                (select the element of Q with the min. distance)
     S←S∪{u}                                (add u to list of visited vertices)
      for all v ∈ neighbors[u]
           do  if   dist[v] > dist[u] + w(u, v)              (if new shortest path found)
                   then     dist[v] ←dist[u] + w(u, v)       (set new value of shortest
path)
return dist
```

Dijkstra's Example

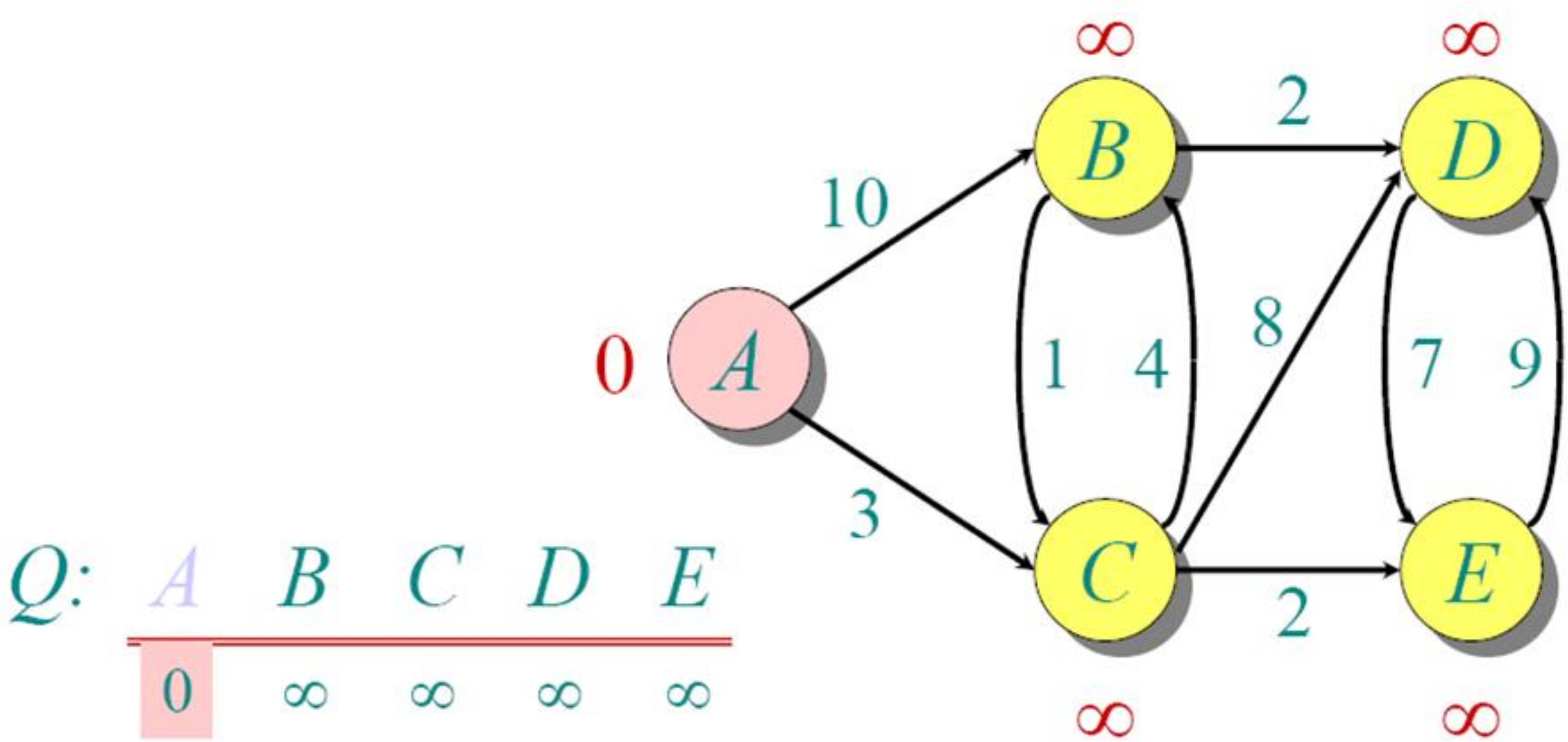
Initialize:

Q:	A	B	C	D	E
	0	∞	∞	∞	∞

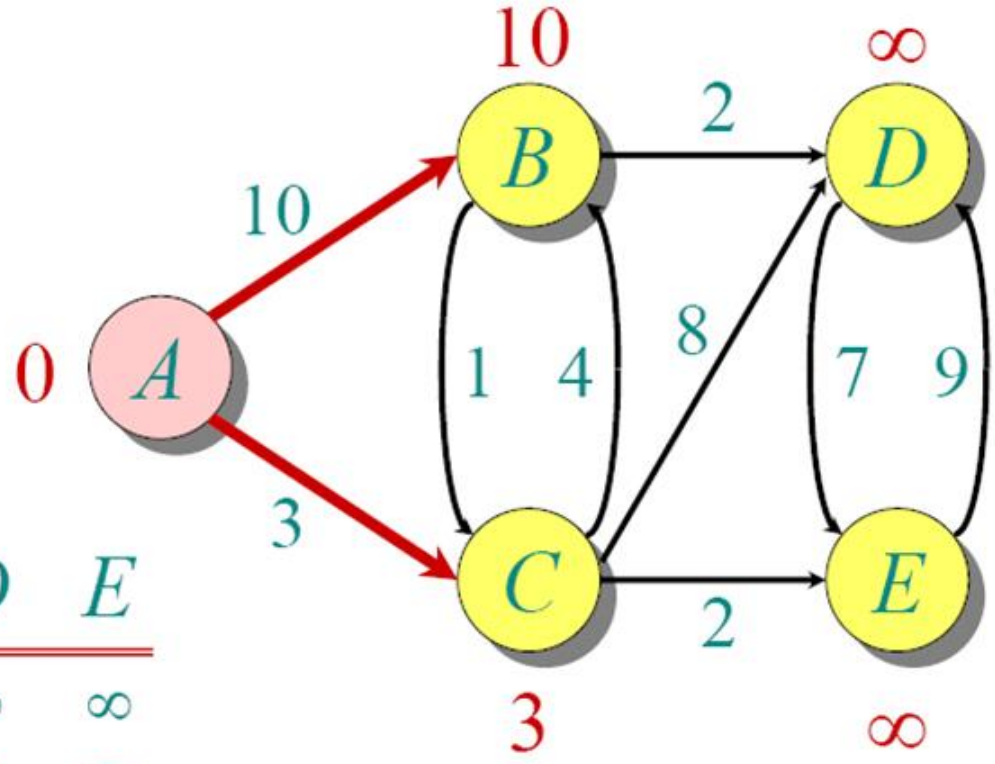


S: {}

Dijkstra’s Example



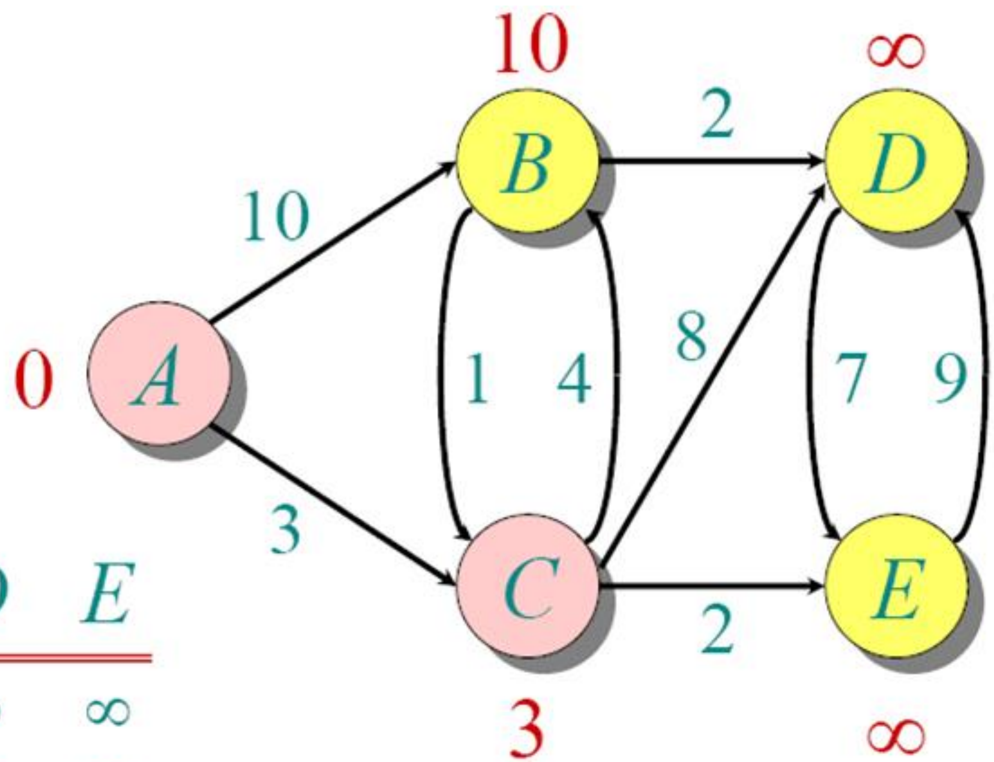
Dijkstra’s Example



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A }

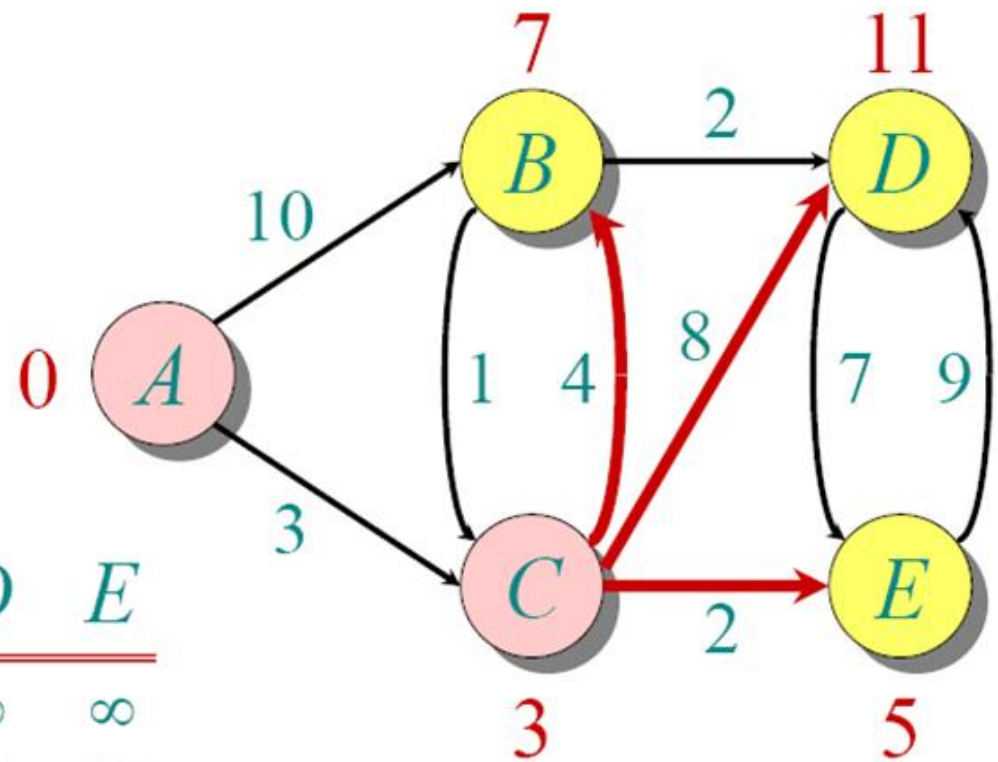
Dijkstra’s Example



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A, C }

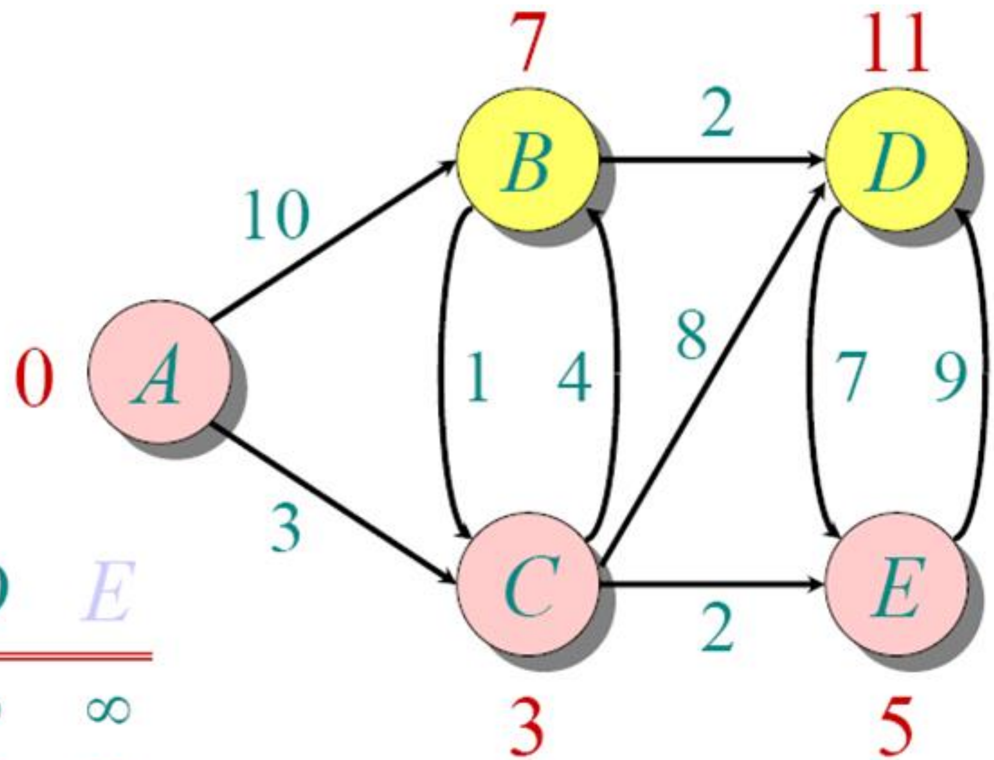
Dijkstra’s Example



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { A, C }

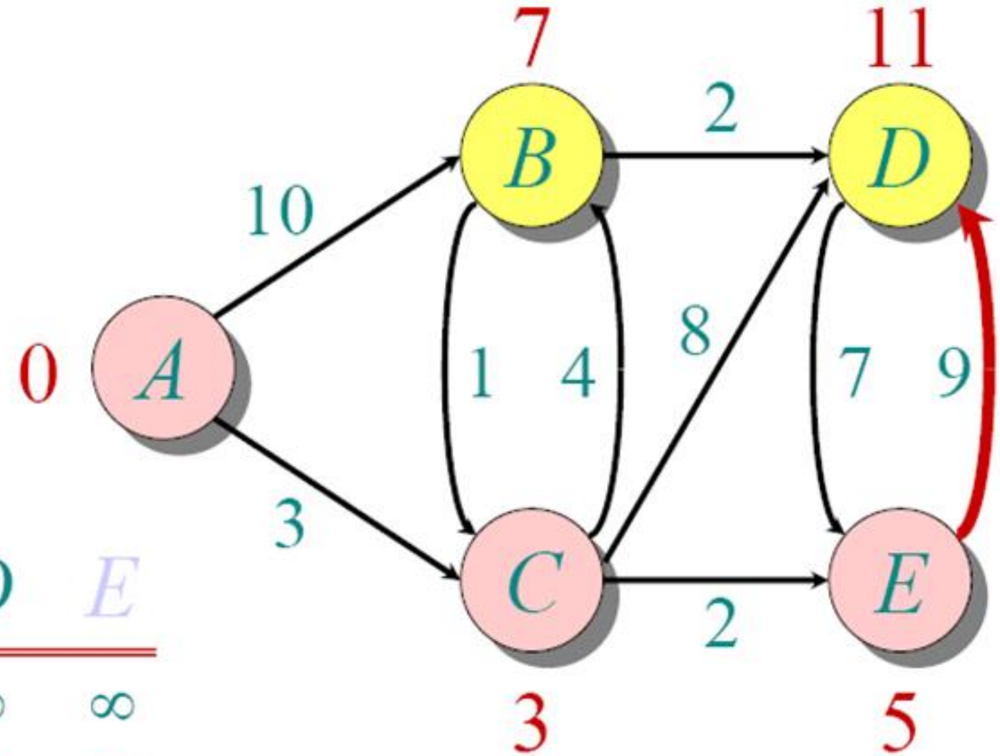
Dijkstra's Example



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { A, C, E }

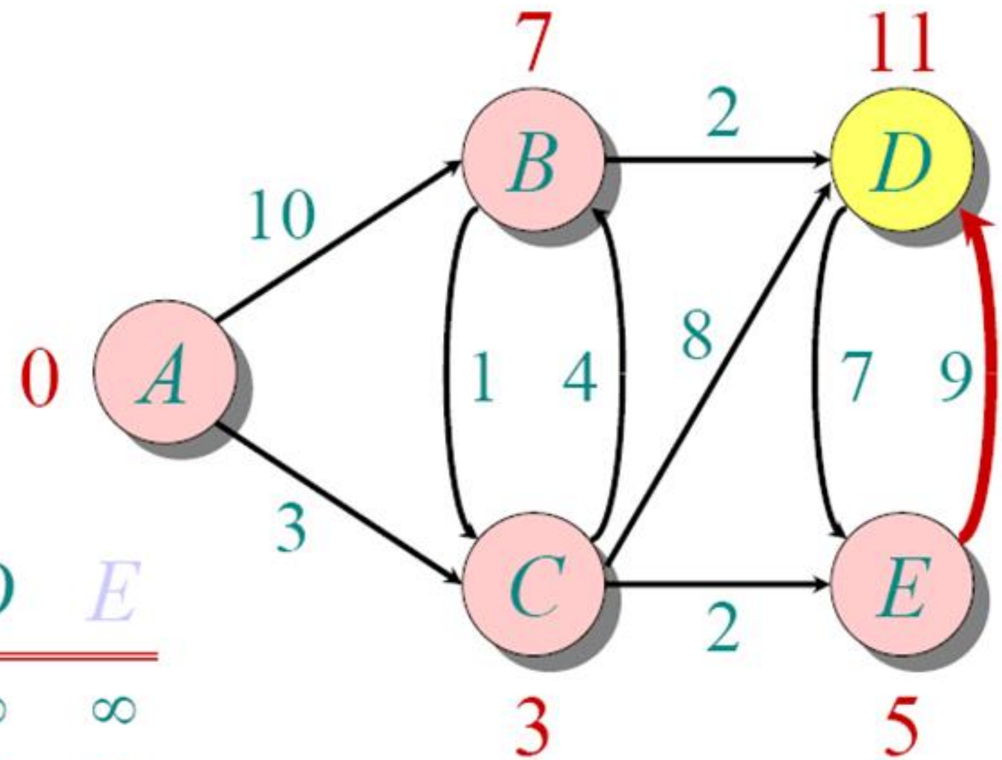
Dijkstra’s Example



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E }

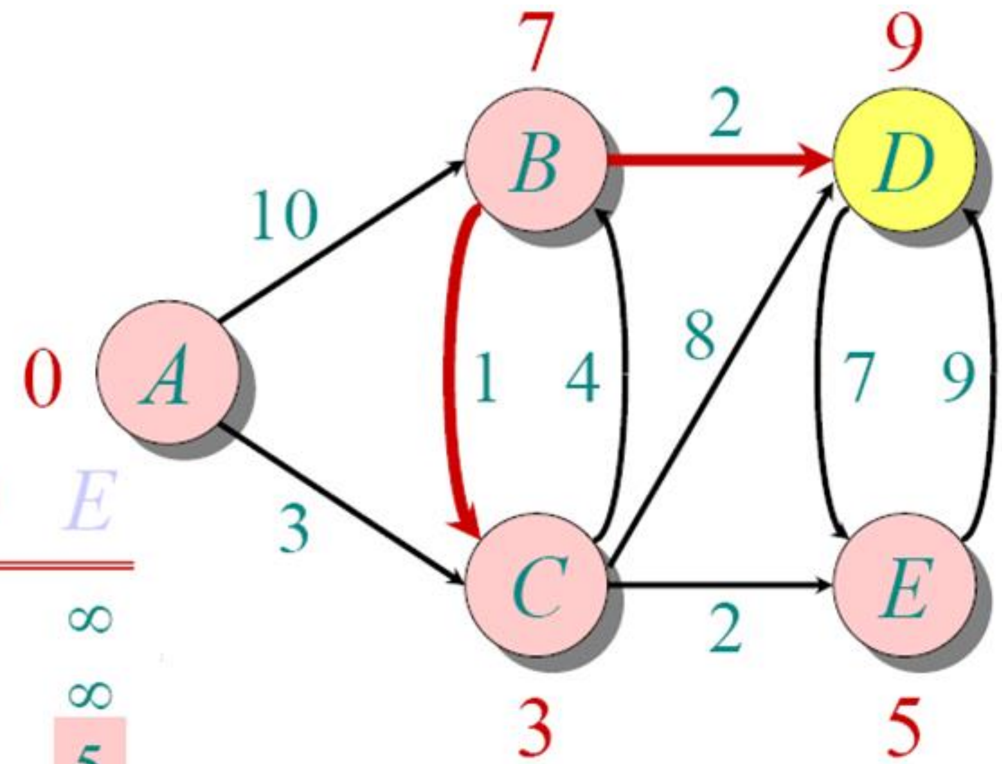
Dijkstra’s Example



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E, B }

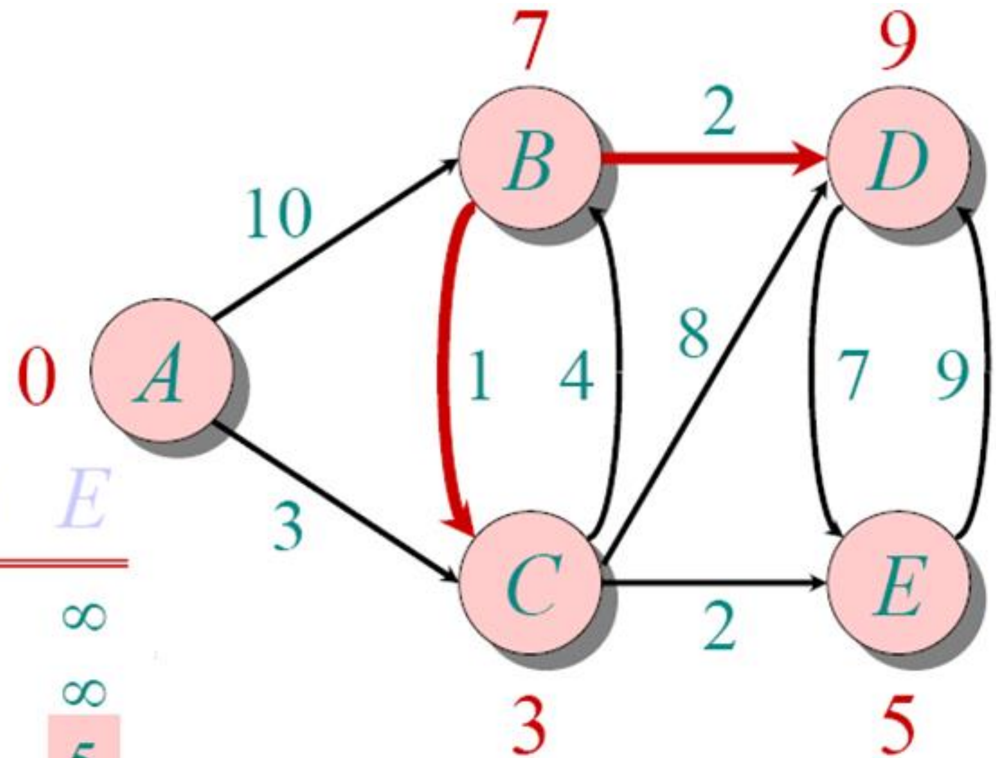
Dijkstra’s Example



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	

S: { A, C, E, B }

Dijkstra’s Example



Q:

A	B	C	D	E
0	∞	∞	∞	∞
	10	3	∞	∞
	7		11	5
	7		11	
			9	

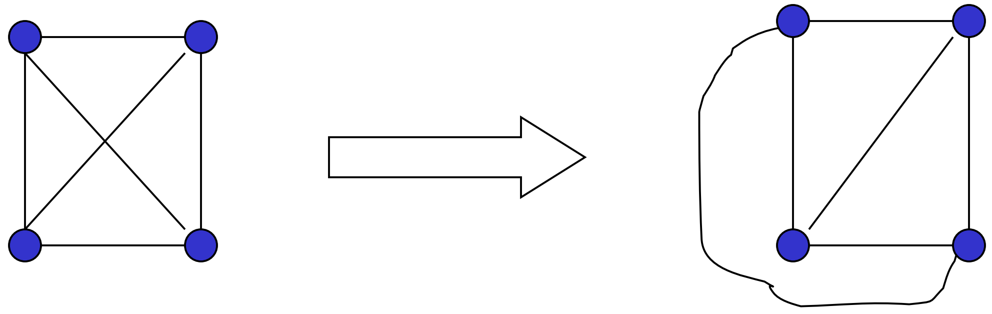
S: { A, C, E, B, D }

Planar Graphs

- A graph (or multigraph) *G* is called *planar* if *G* can be drawn in the plane with its edges intersecting only at vertices of *G,* such a drawing of *G* is called an *embedding* of *G* in the plane.

  Application Example: VLSI design (overlapping edges requires extra layers), Circuit design (cannot overlap wires on board)

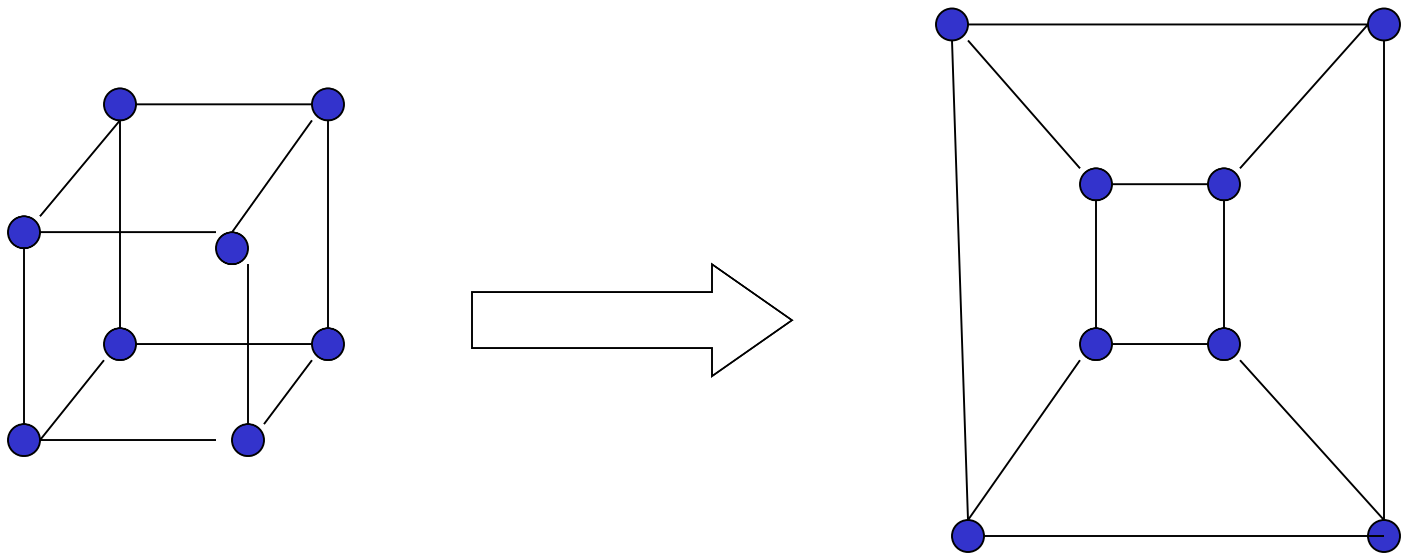
  Representation examples: $K1$,$K2$,$K3$,$K4$ are planar, Kn for $n>4$ are non-planar



K_4

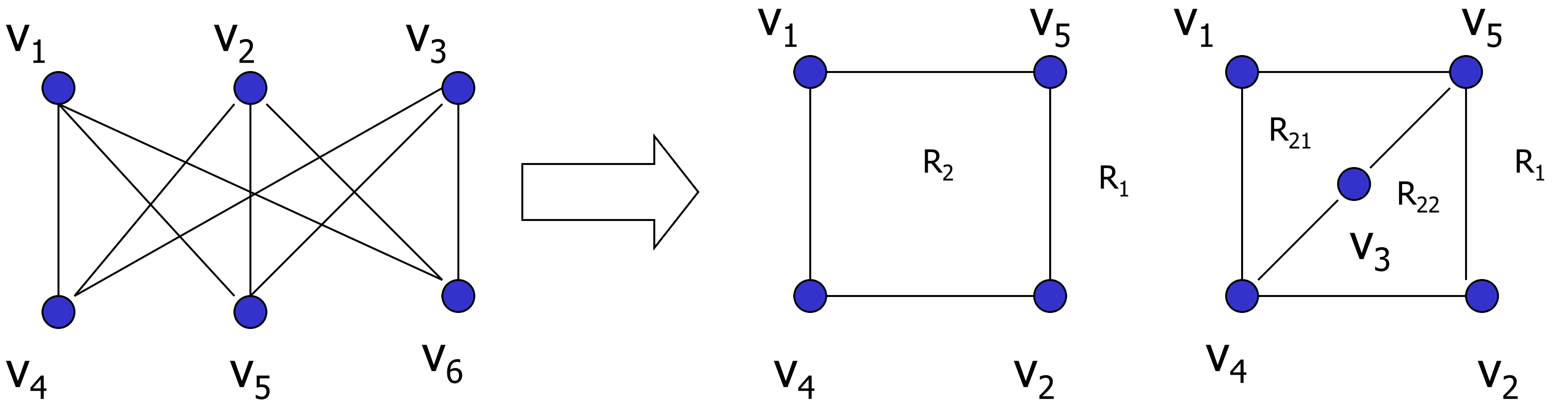
Planar Graphs

- Representation examples: Q_3



Planar Graphs

- Representation examples: $K_{3,3}$ is Nonplanar



Planar Graphs

Theorem : *Euler's planar graph theorem*

For a **connected** planar graph or multigraph:

$$v - e + r = 2$$

number
of vertices

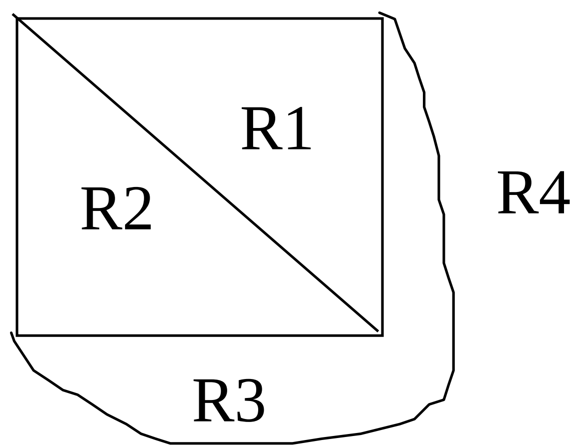
number
of edges

number
of regions

Planar Graphs

Example of Euler's theorem

K_4



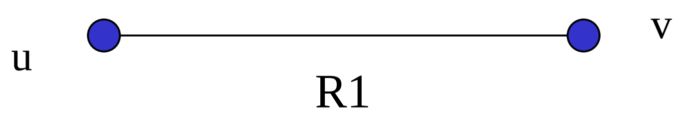
A planar graph divides the plane into several regions (faces), one of them is the infinite region.

$v=4, e=6, r=4, v-e+r=2$

Planar Graphs

- Proof of Euler's formula: By Induction

<u>Base Case:</u> for G1 , $e_1 = 1$, $v_1 = 2$ and $r_1 = 1$

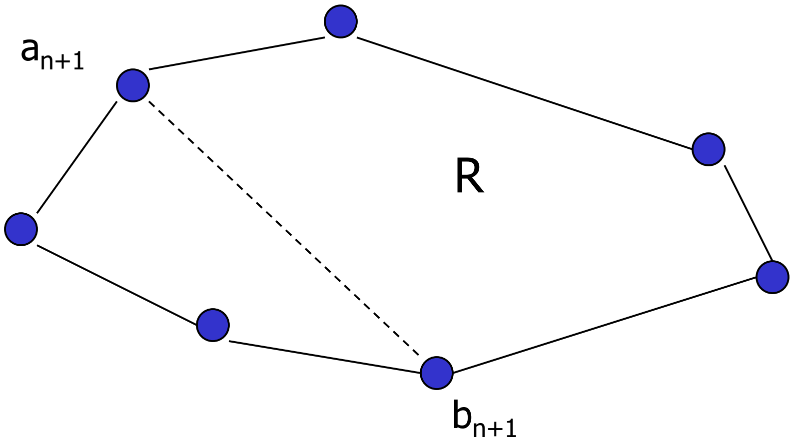


<u>n+1 Case:</u> Assume, $r_n = e_n - v_n + 2$ is true. Let {an+1, bn+1} be the edge that is added to Gn to obtain Gn+1 and we prove that $r_n = e_n - v_n + 2$ is true. Can be proved using two cases.

Planar Graphs

- Case 1:

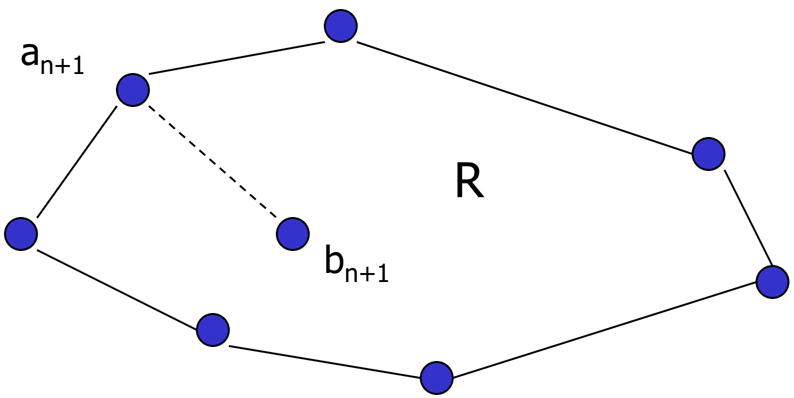
$$r_{n+1} = r_n + 1,\ e_{n+1} = e_n + 1,\ v_{n+1} = v_n \Rightarrow r_{n+1} = e_{n+1} - v_{n+1} + 2$$



Planar Graphs

- <u>Case 2:</u>

$r_{n+1} = r_n, e_{n+1} = e_n + 1, v_{n+1} = v_n + 1 \Rightarrow r_{n+1} = e_{n+1} - v_{n+1} + 2$



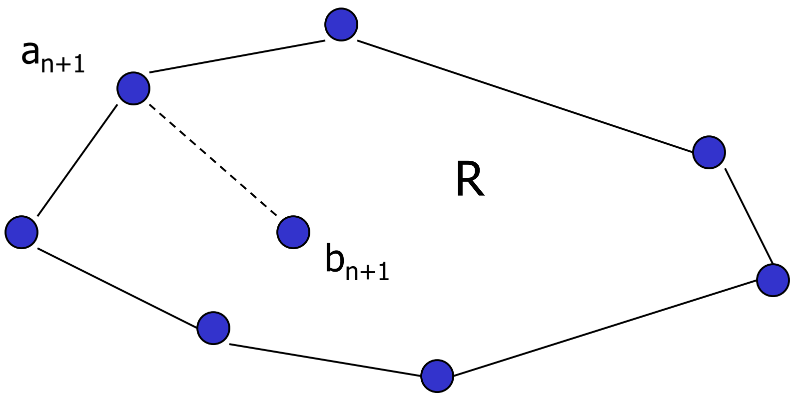
Planar Graphs

Corollary 1: Let G = (V, E) be a connected simple planar graph with |V| = v, |E| = e > 2, and r regions. Then $3r \leq 2e$ and $e \leq 3v - 6$

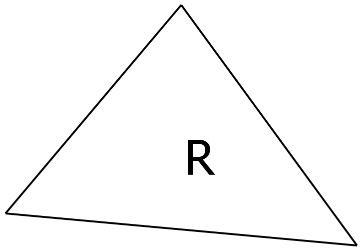
Proof: Since G is loop-free and is not a multigraph, the boundary of each region (including the infinite region) contains at least three edges. Hence, each region has degree ≥ 3.

Degree of region: No. of edges on its boundary; 1 edge may occur twice on boundary -> contributes 2 to the region degree.

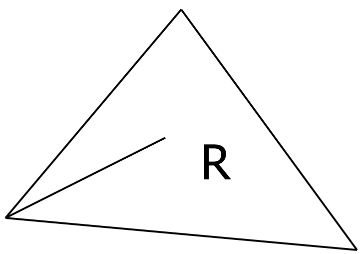
Each edge occurs exactly twice: either in the same region or in 2 different regions



Region Degree



Degree of R = 3



Degree of R = ?

Planar Graphs

Each edge occurs exactly twice: either in the same region or in 2 different regions

$\Rightarrow$2e = sum of degree of r regions determined by 2e

$\Rightarrow$2e $\geq$ 3r. (since each region has a degree of at least 3)

$\Rightarrow$r $\leq$ (2/3) e

$\Rightarrow$From Euler's theorem, 2 = v – e + r

$\Rightarrow$2 $\leq$ v – e + 2e/3

$\Rightarrow$2 $\leq$ v – e/3

$\Rightarrow$So 6 $\leq$ 3v – e

$\Rightarrow$or e $\leq$ 3v – 6

Planar Graphs

Corollary 2: Let $G = (V, E)$ be a connected simple planar graph then G has a vertex degree that does not exceed 5

Proof: If G has one or two vertices the result is true

If G has 3 or more vertices then by Corollary 1, $e \leq 3v - 6$

$\Rightarrow 2e \leq 6v - 12$

If the degree of every vertex were at least 6:

by Handshaking theorem: $2e = \text{Sum}(\deg(v))$

$\Rightarrow 2e \geq 6v$. But this contradicts the inequality $2e \leq 6v - 12$

$\Rightarrow$There must be at least one vertex with degree no greater than 5

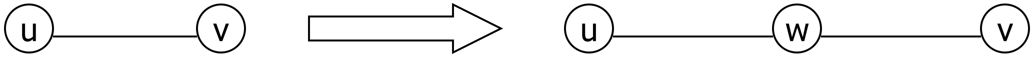
Planar Graphs

Corollary 3: Let G = (V, E) be a connected simple planar graph with v vertices ($v \geq 3$) , e edges, and no circuits of length 3 then $e \leq 2v - 4$

Proof: Similar to Corollary 1 except the fact that no circuits of length 3 imply that degree of region must be at least 4.

Planar Graphs

- **Elementary sub-division:** Operation in which a graph are obtained by removing an edge {u, v} and adding the vertex w and edges {u, w}, {w, v}

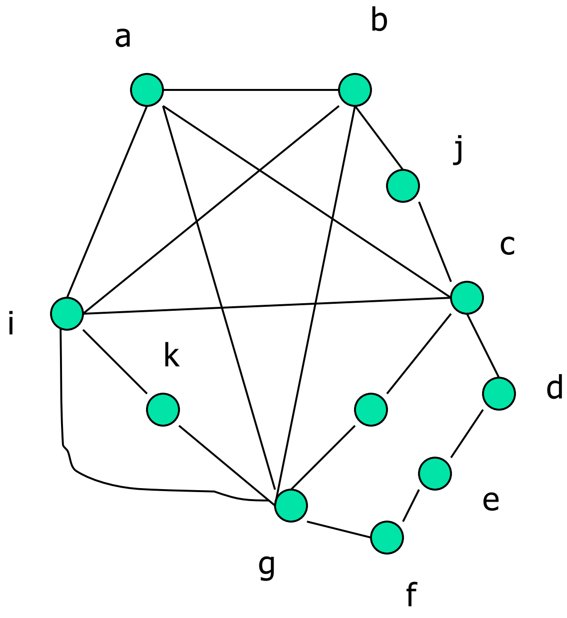


- **Homeomorphic Graphs:** Graphs G1 and G2 are termed as homeomorphic if they are obtained by sequence of elementary sub-divisions.

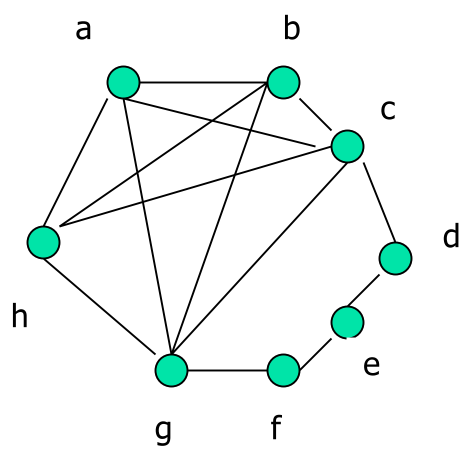
Planar Graphs

- **Kuwratoski's Theorem:** A graph is non-planar if and only if it contains a subgraph homeomorephic to $K_{3,3}$ or K_5

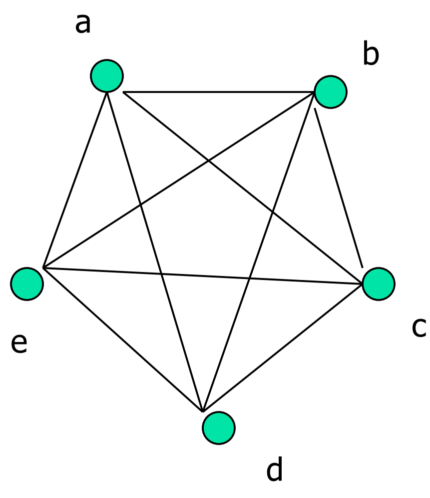
  Representation Example: G is Nonplanar



G



H



K_5