

Vistula, IT Faculty, 2014

# Algorithms and Complexity

Dmitry A. Zaitsev

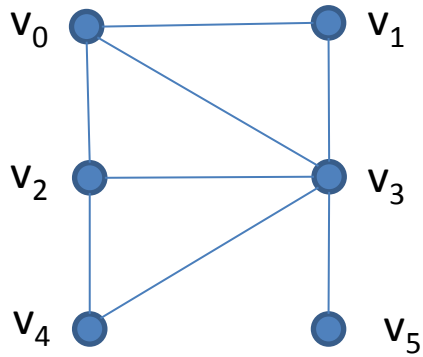
<http://daze.ho.ua>

## Lecture 7:

## Graphs and trees

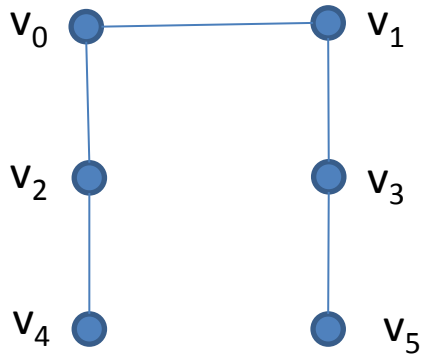
# Undirected graphs

$$G = (V, E), e \in E, e = \{x, y\}, x, y \in V$$



Adjacency matrix

i \ j	0	1	2	3	4	5
0	0	1	1	1	0	0
1	1	0	0	1	0	0
2	1	0	0	1	1	0
3	1	1	1	0	1	1
4	0	0	1	1	0	0
5	0	0	0	1	0	0



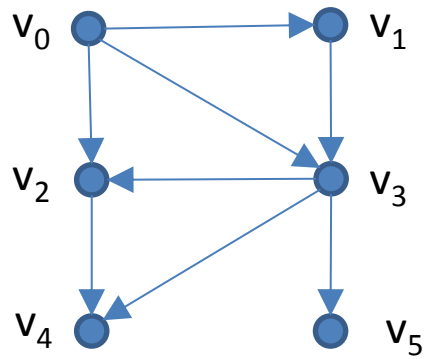
A tree – connected acyclic graph

$$|E| = |V| - 1$$

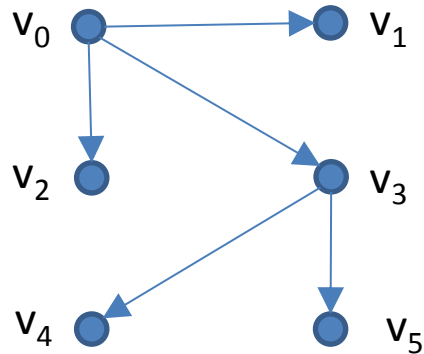
# Directed graphs

$$G = (V, E), E \subseteq V^2 = V \times V$$

Adjacency matrix

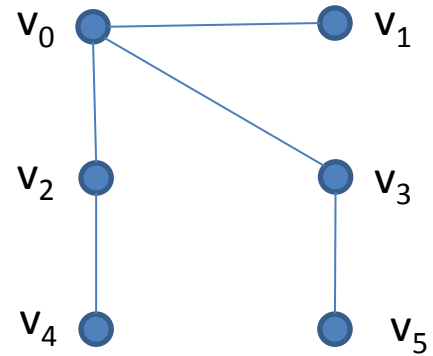


i \ j	0	1	2	3	4	5
0	0	1	1	1	0	0
1	1	0	0	1	0	0
2	1	0	0	1	1	0
3	1	1	1	0	1	1
4	0	0	1	1	0	0
5	0	0	0	1	0	0

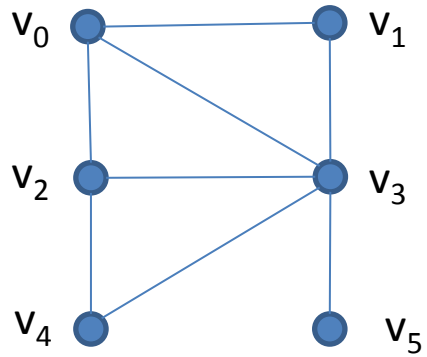


A directed tree

## Breadth-first search

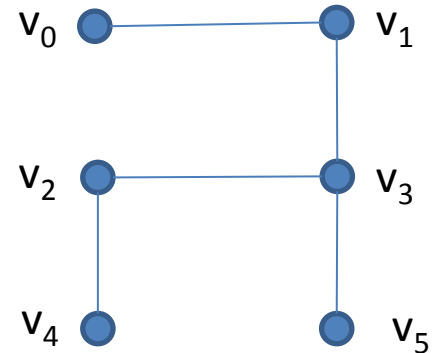


Next – all adjacent



Next – first adjacent then return

## Depth-first search

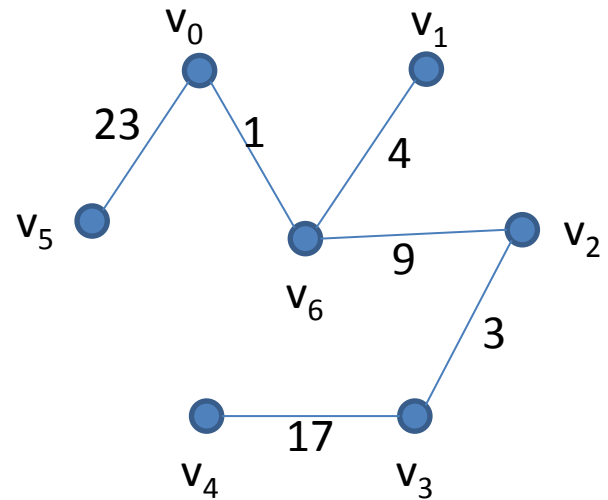
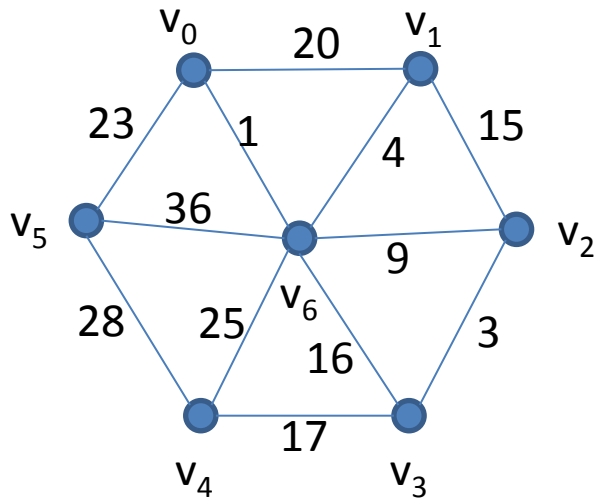


# Breadth-first search

```
while(go)
{
    a_const(nv1, n, 0); // new new vertices
    go=0;
    for(v1=0;v1<n;v1++) // first vertex
    {
        if(nv[v1]==1) // it is new
        {
            for(v2=0;v2<n;v2++) // second vertex
            {
                if(a[v1][v2] && ! vv[v2] ) // an edge leading to a new vertex
                {
                    vv[v2]=1; nv1[v2]=1; go=1;
                    b[v1][v2]=1; b[v2][v1]=1; // comment for undirected
                }
            }
        }
    }
    a_copy(nv1, nv, n);
}
```

# Minimum cost spanning tree

$$G = (V, E), S = (V, T), T \subseteq E, W: E \rightarrow \mathbb{N}$$



## Minimum cost spanning tree – Kruskal's algorithm

- Start from isolated vertices
- Sort edges by the growth of their weights
- Add current edge in case it does not create cycles
- Proceed adding edges till  $|V|-1$  edges added

## Minimal paths – algorithm of Dejkstra

- Labels of vertices:

$l'(s)=0$  – constant ,  $l(v)=\infty$  - temporary,  $p=s$

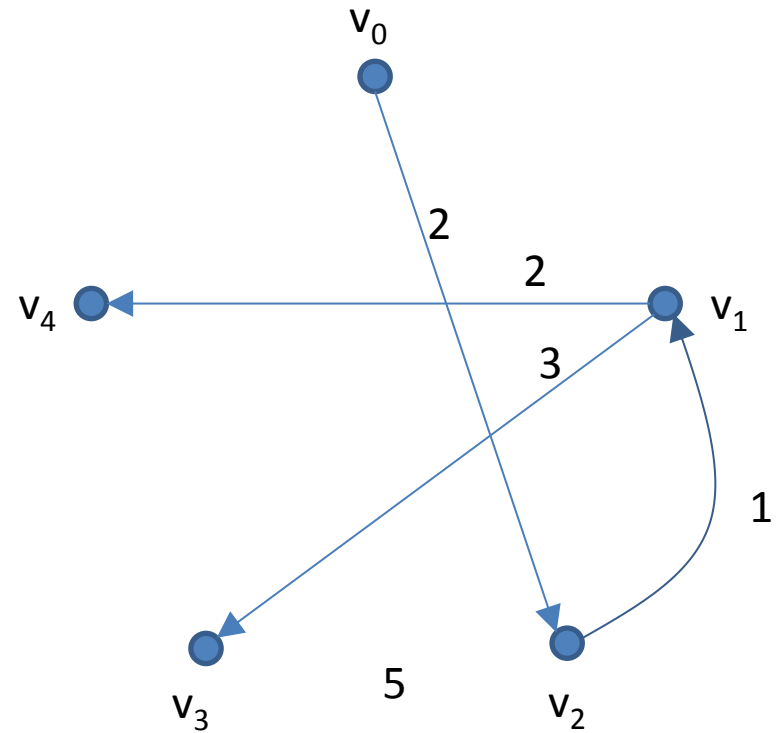
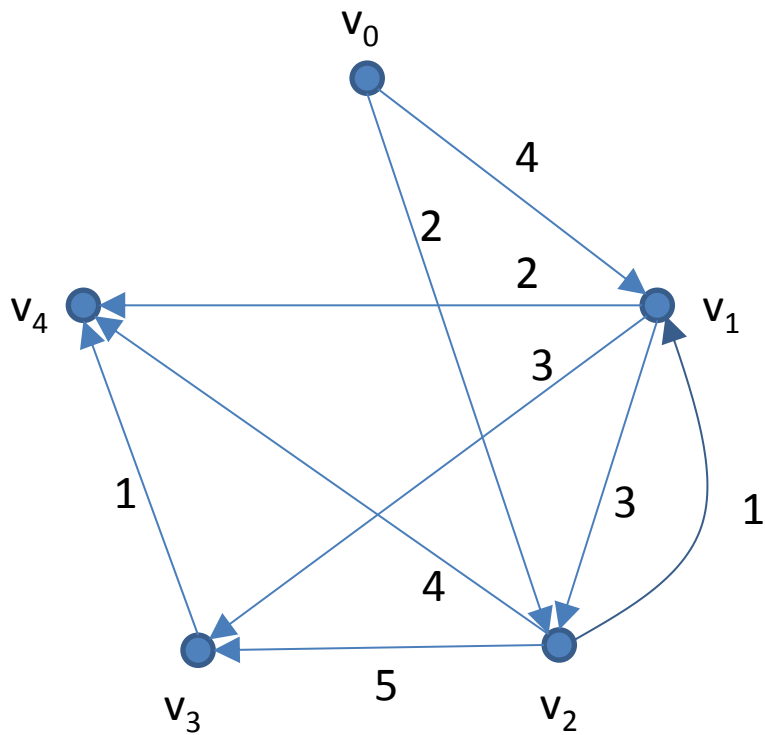
- Recalculate temporary labels:

$v \in G(p): l(v)=\min(l(v), l(p)+w(p,v))$

- For temporary labels:  $l(u)=\min(l(v))$
- Make it constant label:  $l'(u)$
- Set  $p=u$  and repeat recalculating labels

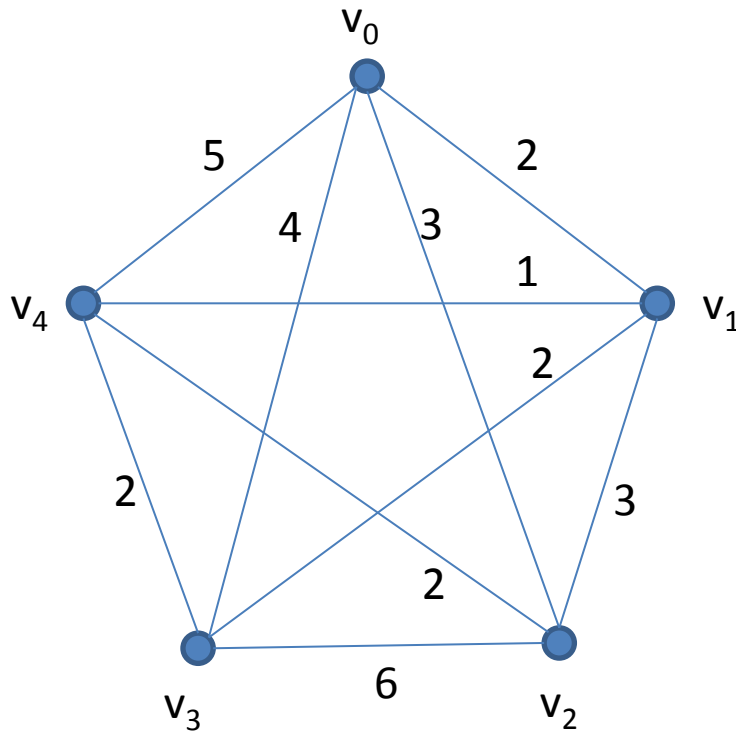


## Example of minimal paths



# Travelling salesman problem

Find the shortest route that passes through each of the nodes in a graph exactly once and returns to the starting node



Shortest route:

$$r = v_0 v_1 v_3 v_4 v_2 v_0$$

$$w(r) = 2 + 2 + 2 + 2 + 3 = 11$$

Exhaustive search:

$$(|V| - 1)! \text{ Paths}$$

$$O(2^n)$$