Algorithms and Data Structures

Module 1

Lecture 3

Graphs: definitions, representations and basic operations

Adigeev Mikhail Georgievich

mgadigeev@sfedu.ru

adimg@yandex.ru

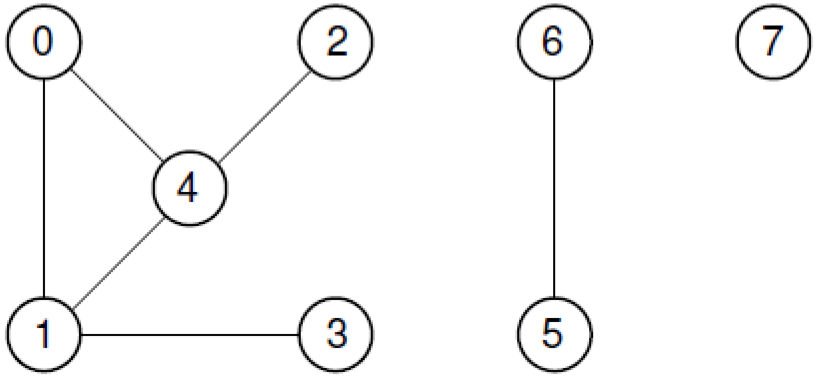
Graphs: definition

Graph G=(V,E)

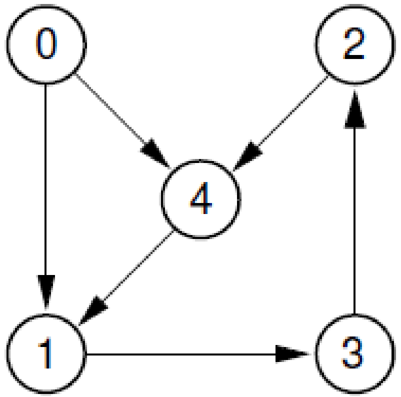
✓ V is a set of ***vertices*** ($v \in V$ – vertex, node). |V|=*n*.

✓ E is a set of ***edges*** ($e = (v, w)$: $v, w \in V$ – edge, arc). |E|=*m*

Undirected graph



Directed graph

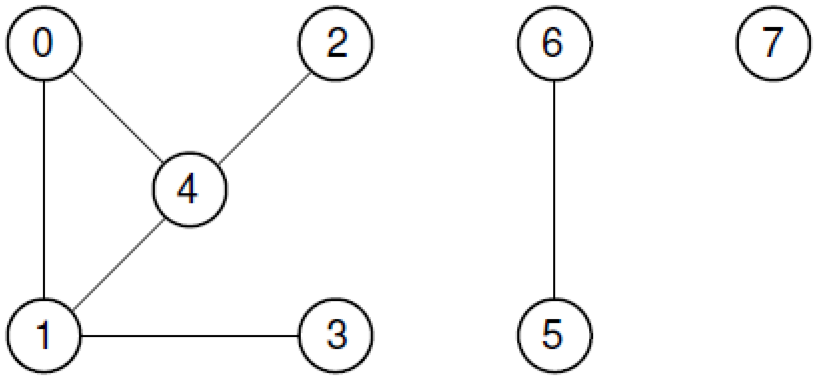


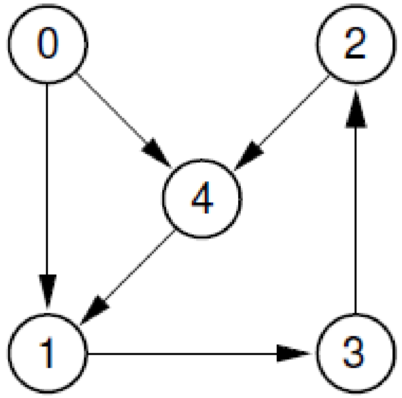
Graphs: definition

$e = (v, w): v, w \in V$

✓ e is ***incident*** to v and w ; v (w) is incident to e;

✓ v and w are ***adjacent*** ; they are *neighbours*.

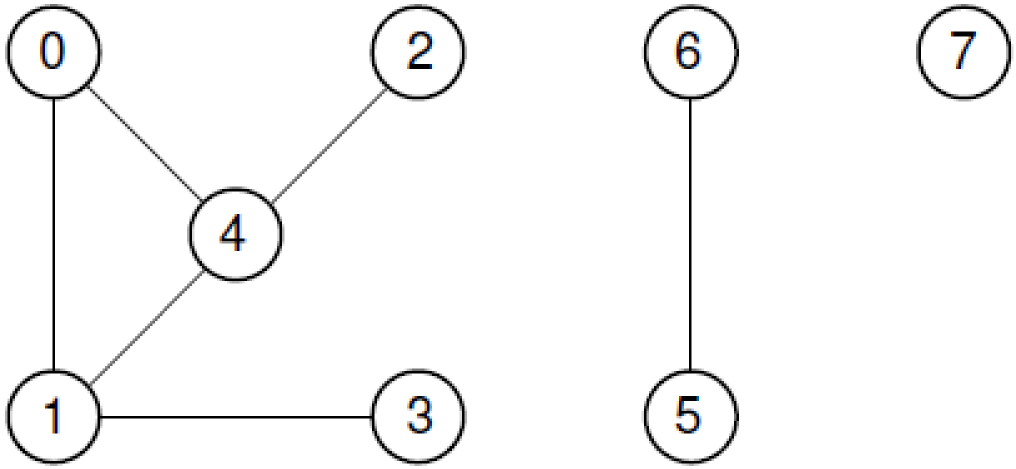




Graphs: definition

$v \in V$:

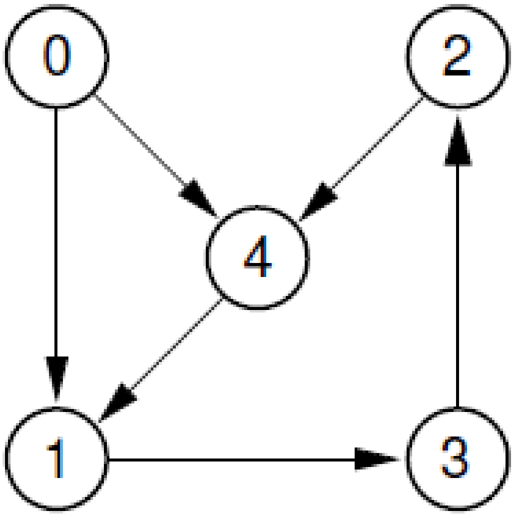
✓ $\deg(v)$ – ***degree*** of vertex v = number of edges incident to v .



Graphs: definition

$v \in V$:

- ✓ $\deg(v)$ – ***degree*** of vertex v = number of edges incident to v .
- ✓ $\text{outdeg}(v)$– out-degree of vertex v = number of edges which start from v .
- ✓ $\text{indeg}(v)$– in-degree of vertex v = number of edges which end at v .
- ✓ v is a ***source*** iff $\text{indeg}(v) = 0$
- ✓ v is a ***sink*** iff $\text{outdeg}(v) = 0$

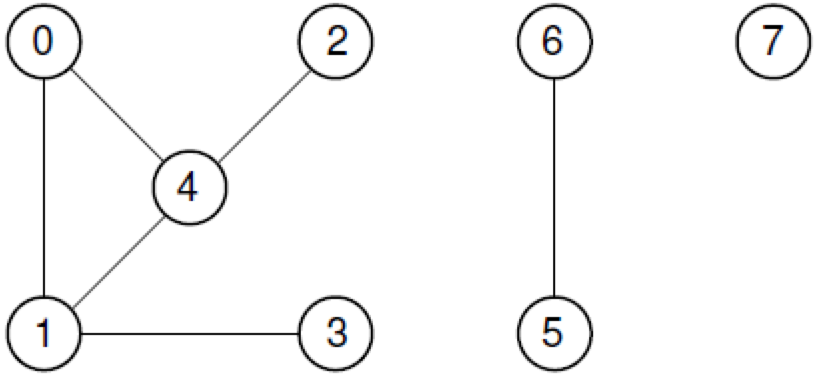


Graphs: representations

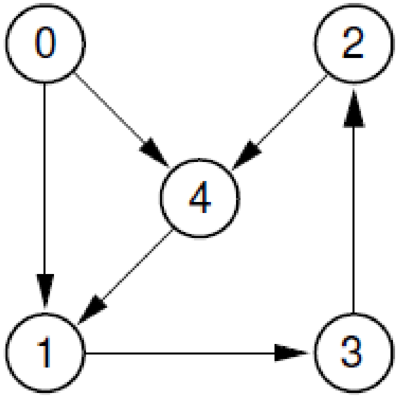
Edge list

$$E = \{e_1 = (u_1, v_1), \dots, e_m = (u_m, v_m)\}$$

0	1
0	4
1	3
1	4
2	4
5	6



0	1
0	4
1	3
4	1
2	4
3	2



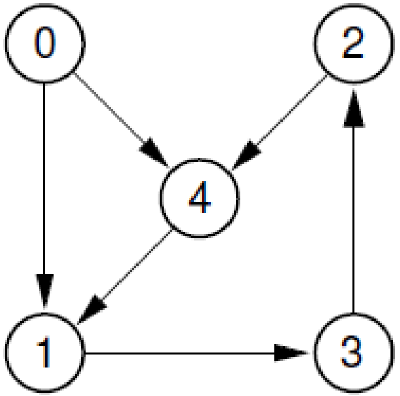
Graphs: representations

Edge list

$$E = \{e_1 = (u_1, v_1), \dots, e_m = (u_m, v_m)\}$$

Property	Complexity
out-deg(v)	$O(m)$
in-deg(v)	$O(m)$
deg(v)	$O(m)$
has_edge(v,w)	$O(m)$
is_source(v)	$O(m)$
is_sink(v)	$O(m)$

```
0 1
0 4
1 3
4 1
2 4
3 1
```

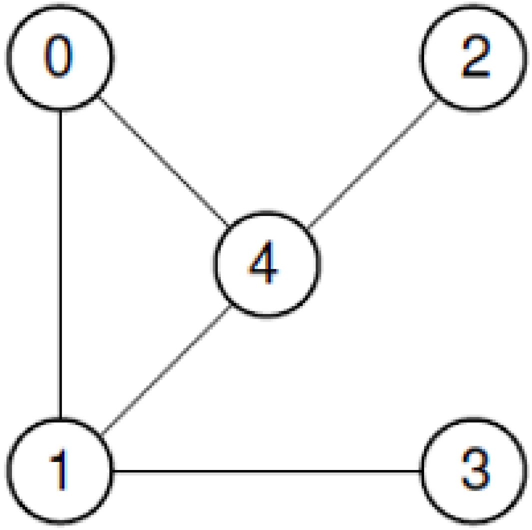


Graphs: representations

Adjacency matrix

$$A = \{a_{ij}\}_{i,j=1}^{n}: a_{ij} = \begin{cases} 1, if\ (i,j) \in E \\ 0,\ otherwise \end{cases}$$

	0	1	2	3	4
0	0	1	0	0	1
1	1	0	0	1	1
2	0	0	0	0	1
3	0	1	0	0	0
4	1	1	1	0	0

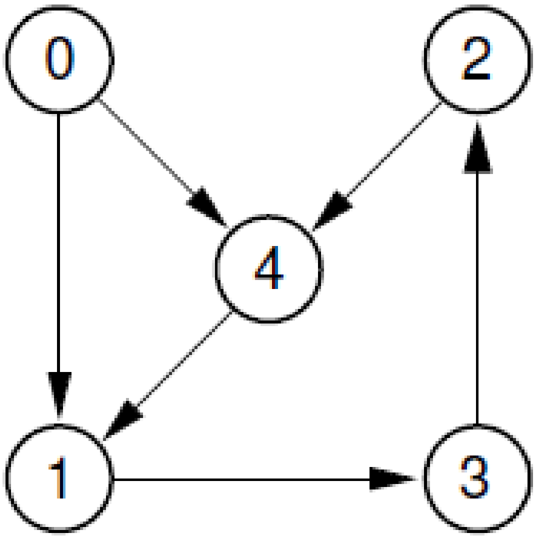


Graphs: representations

Adjacency matrix

$$A = \{a_{ij}\}_{i,j=1}^{n} : a_{ij} = \begin{cases} 1, if\ (i,j) \in E \\ 0,\ otherwise \end{cases}$$

	0	1	2	3	4
0	0	1	0	0	1
1	0	0	0	1	0
2	0	0	0	0	1
3	0	0	1	0	0
4	0	1	0	0	0



Graphs: representations

Adjacency matrix

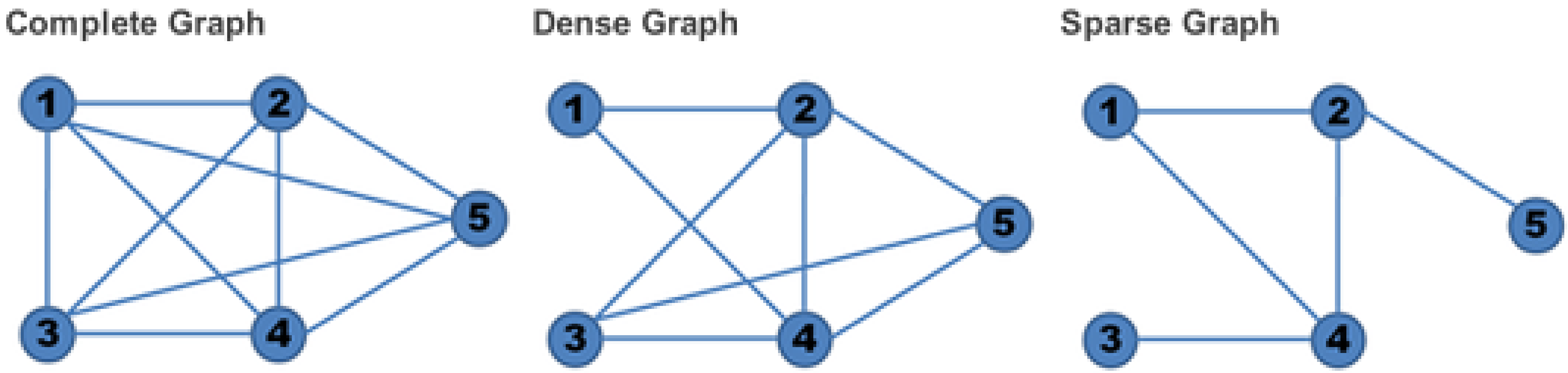
$A = \{a_{ij}\}_{i,j=1}^{n}$ contains $O(n^2)$ entries.

- Space-efficient for ***dense*** graphs $(m \sim O(n^2))$.
- Is space-inefficient for ***sparse*** graphs $(m \sim O(n))$.

	0	1	2	3	4
0	0	1	0	0	1
1	0	0	0	1	0
2	0	0	0	0	1
3	0	0	1	0	0
4	0	1	0	0	0

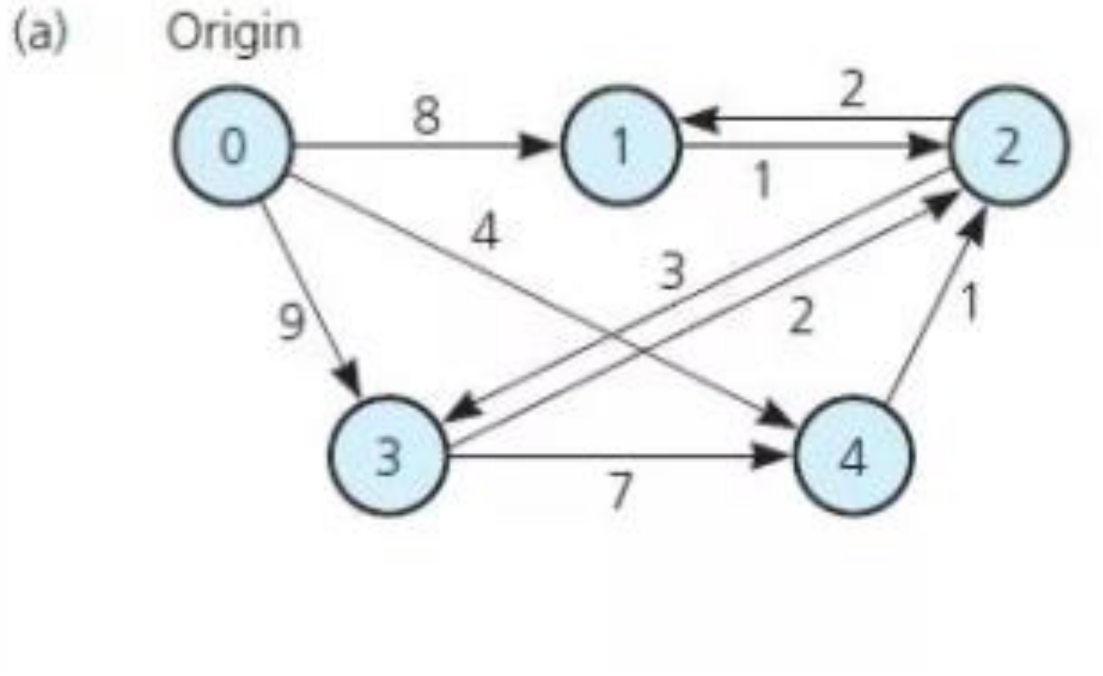
Graphs: representations

Adjacency matrix



Graphs: representations

Adjacency matrix



(b)

	0	1	2	3	4
0	∞	8	∞	9	4
1	∞	∞	1	∞	∞
2	∞	2	∞	3	∞
3	∞	∞	2	∞	7
4	∞	∞	1	∞	∞

Graphs: representations

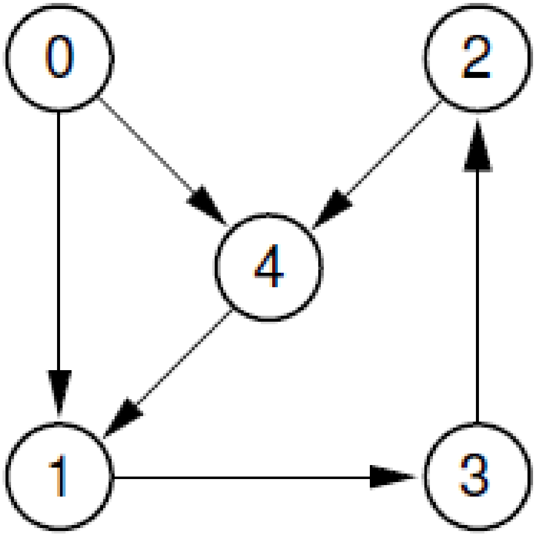
Adjacency matrix

Property	Complexity
out-deg(v)	$O(n)$
in-deg(v)	$O(n)$
deg(v)	$O(n)$
has_edge(v,w)	$O(1)$
is_source(v)	$O(n)$
is_sink(v)	$O(n)$

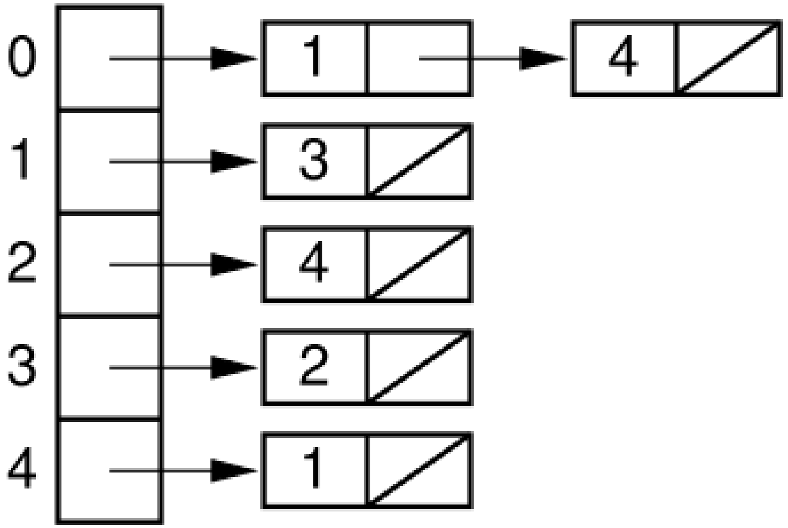
	0	1	2	3	4
0	0	1	0	0	1
1	0	0	0	1	0
2	0	0	0	0	1
3	0	0	1	0	0
4	0	1	0	0	0

Graphs: representations

Adjacency list



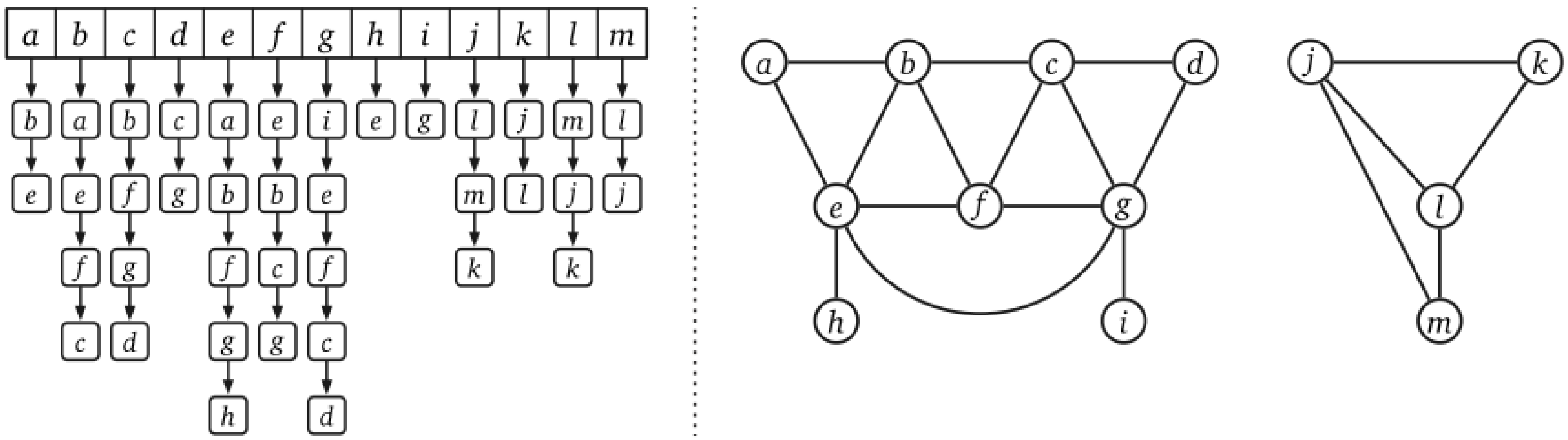
	0	1	2	3	4
0	0	1	0	0	1
1	0	0	0	1	0
2	0	0	0	0	1
3	0	0	1	0	0
4	0	1	0	0	0



Space complexity: $O(n + m)$

Graphs: representations

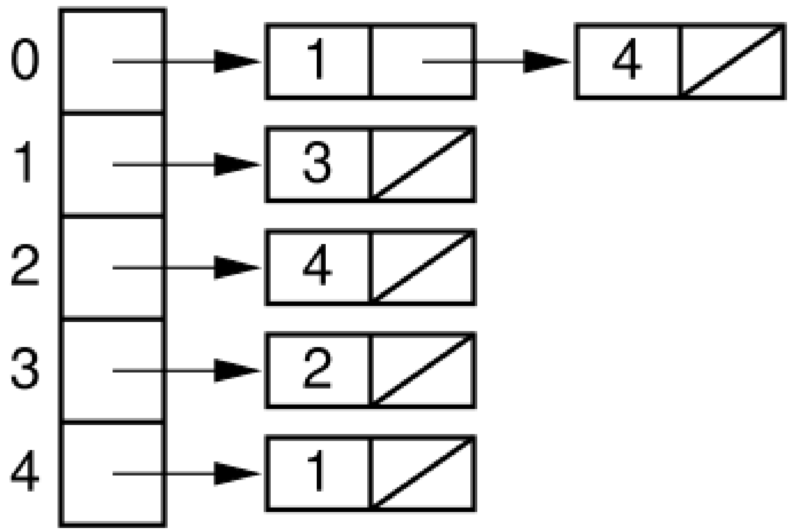
Adjacency list



Graphs: representations

Adjacency list

Property	Complexity
out-deg(v)	$O(\max deg) = O(n)$
in-deg(v)	$O(n + m) = O(m)$
deg(v)	$O(m)$
has_edge(v,w)	$O(\max deg) = O(n)$
is_source(v)	$O(m)$
is_sink(v)	$O(1)$



Space complexity: $O(n + m)$

Graphs: representations

Property	Edge list	Adjacency matrix	Adjacency list
out-deg(v)	$O(m)$	$O(n)$	$O(\max deg) = O(n)$
in-deg(v)	$O(m)$	$O(n)$	$O(n + m) = O(m)$
deg(v)	$O(m)$	$O(n)$	$O(m)$
has_edge(v,w)	$O(m)$	$O(1)$	$O(\max deg) = O(n)$
is_source(v)	$O(m)$	$O(n)$	$O(m)$
is_sink(v)	$O(m)$	$O(n)$	$O(1)$
memory	$O(m)$	$O(n^2)$	$O(n + m)$