

# Optimal Distance Labeling Scheme for Interval and Circular-arc Graphs

**C. Gavoille**

LaBRI, University Bordeaux I

**C. Paul**

CNRS - LIRMM, University of Montpellier II

# Graph Representations

How to **encode informations** about a graph  $G$  in such a way that an **oracle** is able to answer **queries** on that graph?

**Examples of queries** : adjacency, connectivity, **distance**, routing, connectivity ...

**Type of representation**: centralized, **distributed** or mixed

**Evaluation**: a tradeoff between

- **Space**: number of bits needed
- **Times**: complexity to answer queries and to compute the labels (usually the WORD-RAM model is used)

# Labeling Schemes

Let  $Q$  be a type of queries (adjacency, distance, ...).

For a family  $\mathcal{F}$  of graphs, a  $Q$ -labeling scheme is a pair of functions  $\langle L, f \rangle$  such that:

- $L(v, G)$  is a binary label associated to vertex  $v$  in graph  $G$
- $f(L(x, G), L(y, G))$  is a decoder function that answer the query between  $x$  and  $y$  in graph  $G$

The labeling scheme is said a  $l(n)$ - $Q$  labeling scheme if for every  $n$ -vertex graph  $G \in \mathcal{F}$ , the length of the labels is bounded by  $l(n)$  bits.

## Adjacency vs. Distance Labeling Schemes

**Implicit graph conjecture [Kannan, Naor & Rudich 1992]**

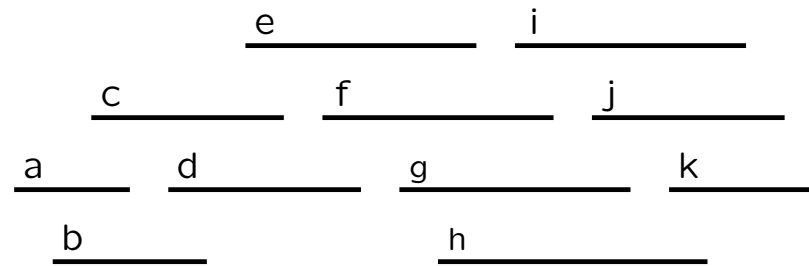
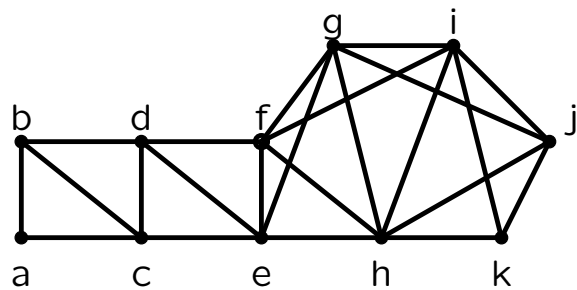
Any hereditary family  $\mathcal{F}$  containing no more than  $2^{n \cdot k(n)}$  graphs of  $n$  vertices enjoys a  $\mathcal{O}(k(n))$ -adjacency labeling scheme.

**Example of trees**  $\left\{ \begin{array}{l} 2 \log n\text{-adjacency labeling scheme} \\ \mathcal{O}(\log^2 n)\text{-distance labeling scheme} \end{array} \right.$

**Question:** does there exist a **non-trivial family** of graphs (**large enough**) for which the distance can be encoded within the same order of space as the adjacency?

# Interval Graphs

A graph is an **interval graph** if it is the intersection graph of a family of intervals on the real line.



Interval graphs enjoys a  $2 \log n$ -adjacency labeling scheme :  
any vertex  $x$  stores its left and right boundaries  $r(x), l(x)$

**[Katz, Katz & Peleg 2000]** The family of interval graphs enjoys an  $O(\log^2 n)$ -distance labeling scheme (**DLS** for short)

# Optimal Distance Labeling Schemes

An interval graph is *proper* iff there exists a layout without any interval inclusion.

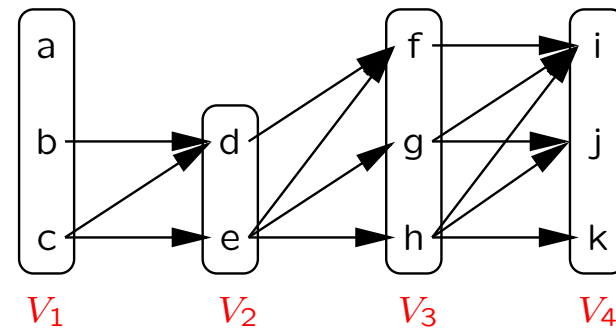
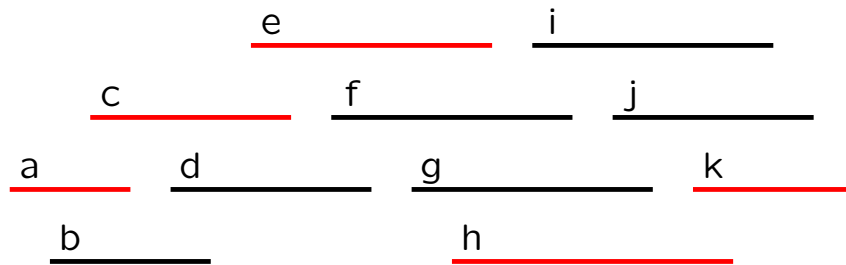
## Theorem 1 [Gavoille & Paul 2003]

- $n$  vertex *proper interval* graphs enjoys a  $2 \lceil \log n \rceil$ -DLS;
- $n$  vertex *interval* graphs enjoys a  $3 + 5 \lceil \log n \rceil$ -DLS;
- $n$  vertex *circular-arc* graphs enjoys a  $\mathcal{O}(\log n)$ -DLS.

The distance decoder has *constant time* decoder and given the sorted list of intervals, the labels can be computed in  $\mathcal{O}(n)$  time.

# Layer Partition

Let  $[x_0, \dots, x_k]$  be the **base-path** where  $x_0$  has  $r(x)$  minimum and  $\forall i > 0, x_i \in N(x_{i-1})$  such that  $r(x_i)$  is maximum



The **layer partition**  $V_1, \dots, V_k$  is defined by

$$V_i = \{v | l(v) < l(x_{i-1})\} \setminus \bigcup_{0 \leq j < i} V_j \text{ with } V_0 = \emptyset$$

## Graph of Errors

Let  $\lambda(x)$  be the integer such that  $x \in V_{\lambda(x)}$  and let  $H$  be the digraph on  $V$  composed of the arcs  $xy$  such that

$$\lambda(x) < \lambda(y) \text{ and } (x, y) \in E$$

The graph of errors is the transitive closure  $H^t$  of  $H$ , let

$$adj_{H^t}(x, y) = 1 \text{ iff } xy \text{ is an arc of } H^t$$



**Theorem 2** For all distinct vertices  $x, y$  such that  $\lambda(x) \leq \lambda(y)$ ,

$$dist_G(x, y) = \lambda(y) - \lambda(x) + 1 - adj_{Ht}(x, y)$$

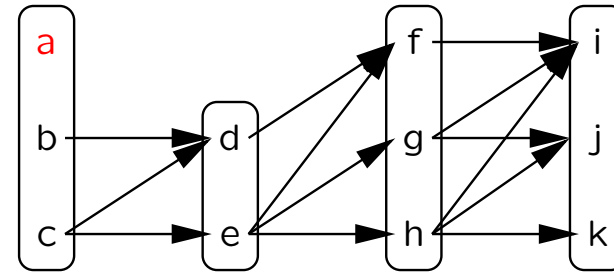
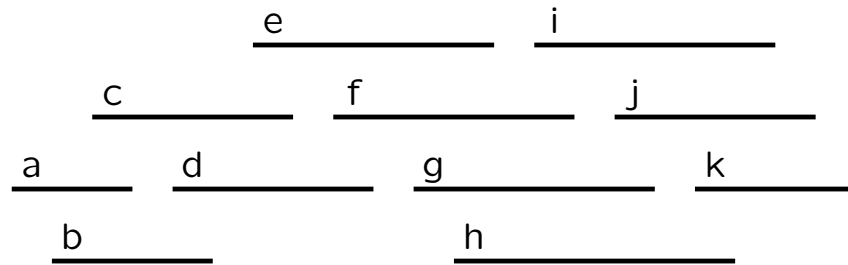
**Theorem 3** There exists a linear ordering  $\pi$  of the vertices, constructible in  $\mathcal{O}(n)$  time, such that

$$adj_{Ht}(x, y) = 1 \text{ iff } \lambda(x) < \lambda(y) \text{ and } \pi(x) > \pi(y)$$

$\pi$  is the pop ordering of a DFS on  $H$  using  $l(x)$  as a priority rule.

The label of the vertex  $x$  is  $L(x, G) = \langle \lambda(x), \pi(x) \rangle$

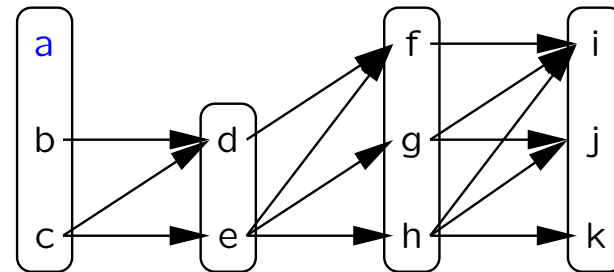
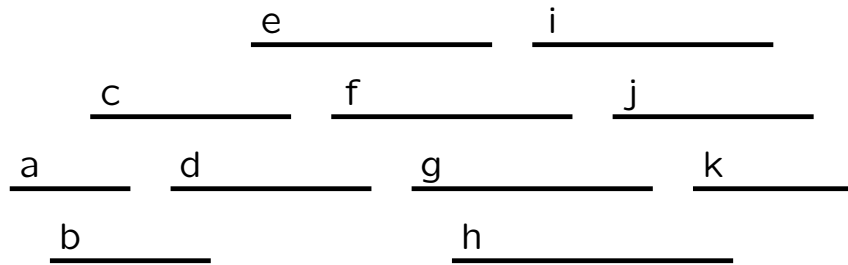
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$											

Stack: a

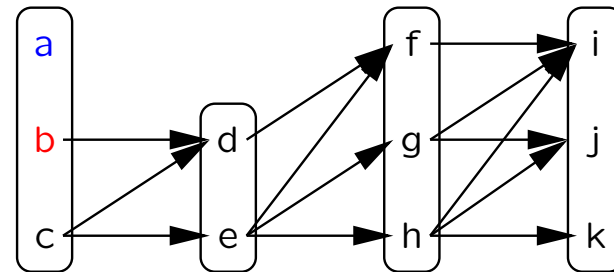
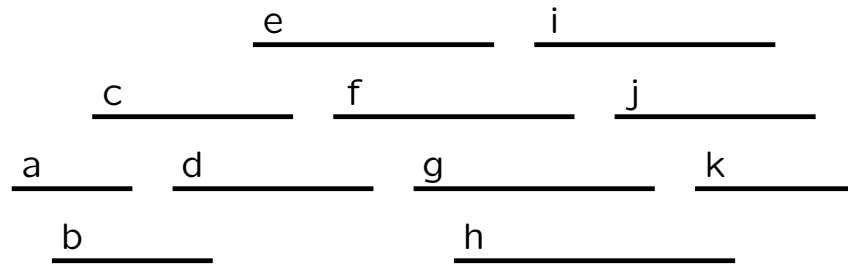
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1										

Stack: a

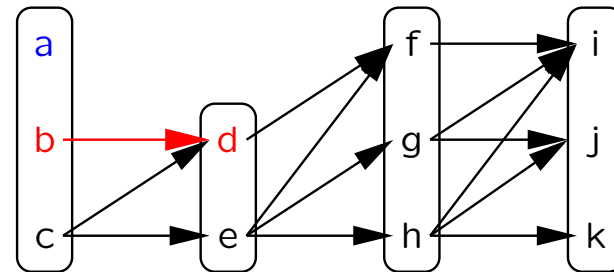
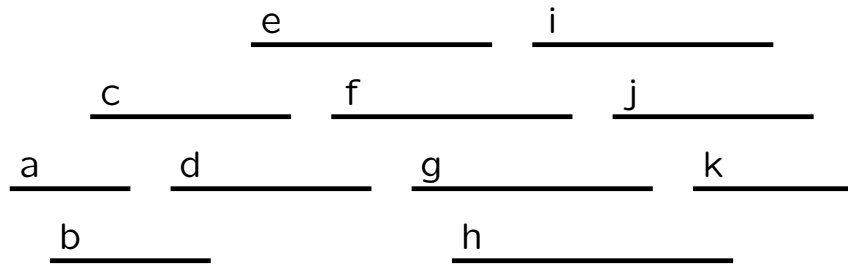
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1										

Stack: ~~a~~ b

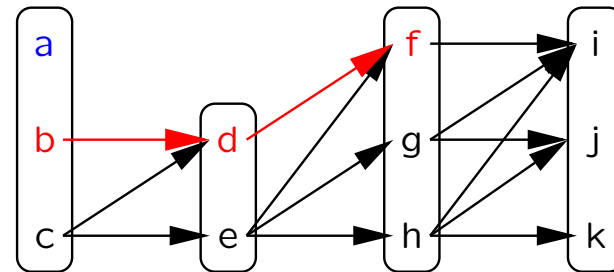
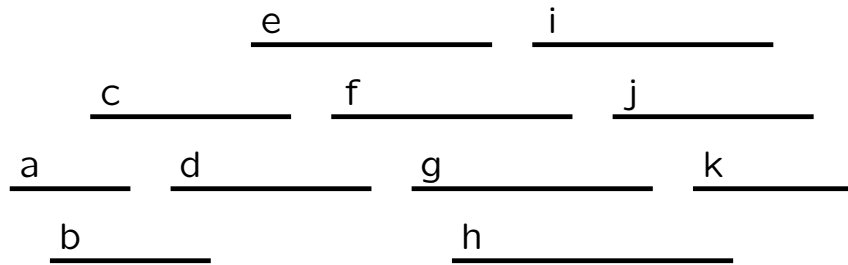
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1										

Stack: ~~a~~ b d

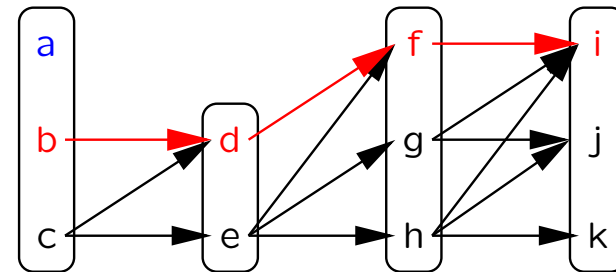
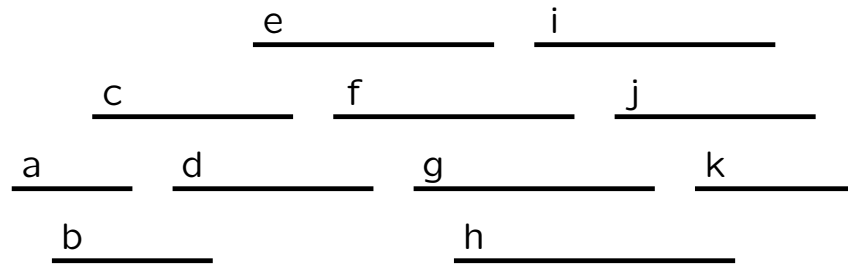
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1										

Stack: ~~a~~ b d f

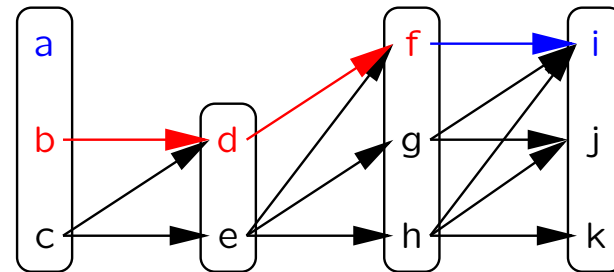
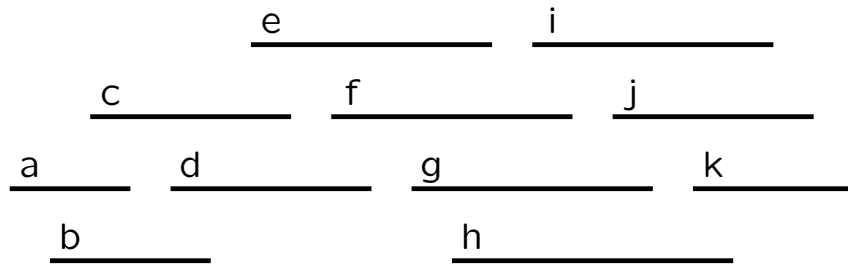
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1										

Stack: ~~a~~ b d f i

# An Example

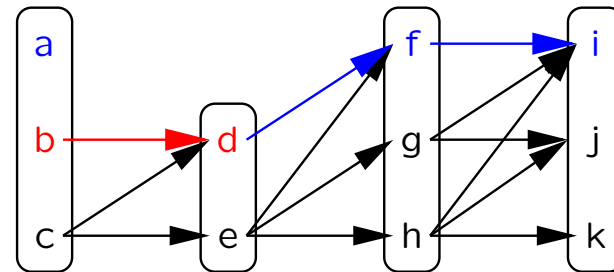
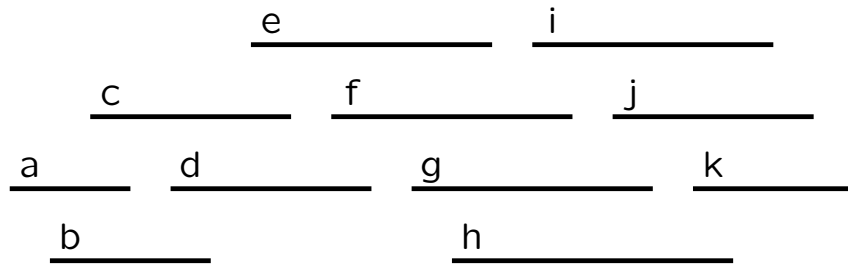


vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1								2		

Stack: ~~a~~ b d f ~~k~~



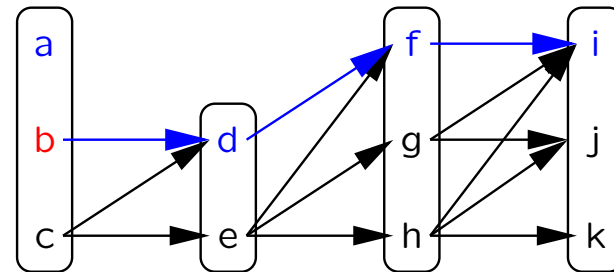
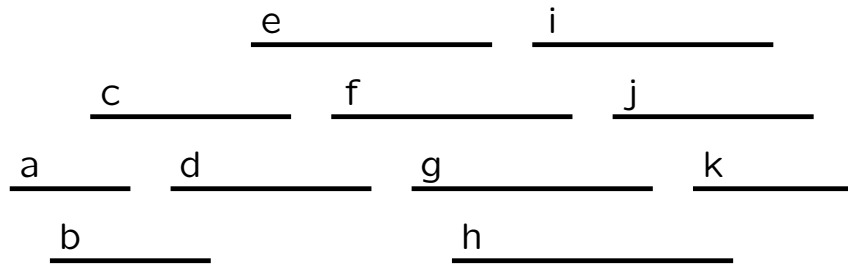
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1					3			2		

Stack: ~~a~~ b d ~~f~~ i

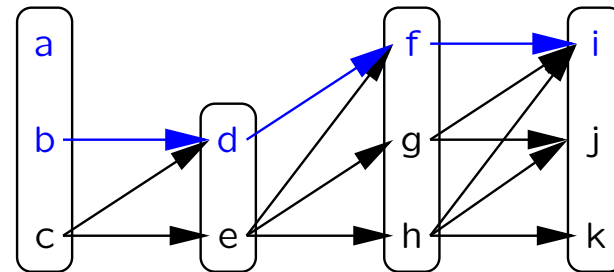
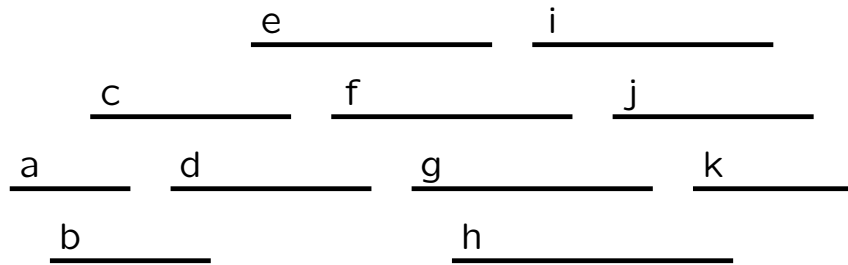
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1			4		3			2		

Stack: ~~a~~ b ~~d~~ f i

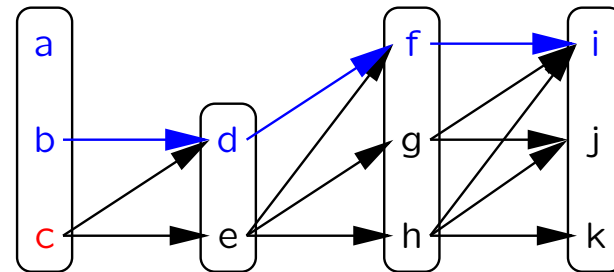
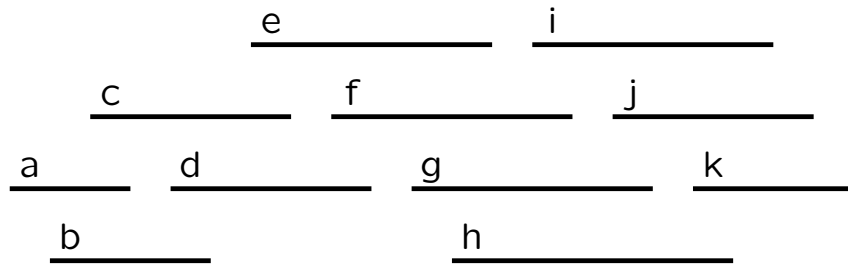
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4		3			2		

Stack: a b d f i

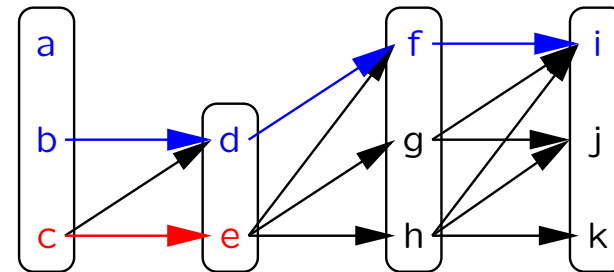
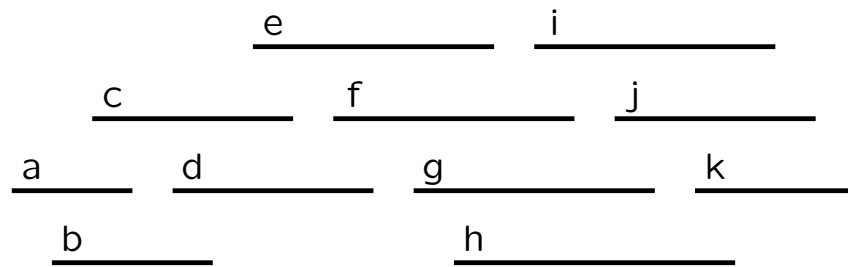
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4		3			2		

Stack: a b d f i c

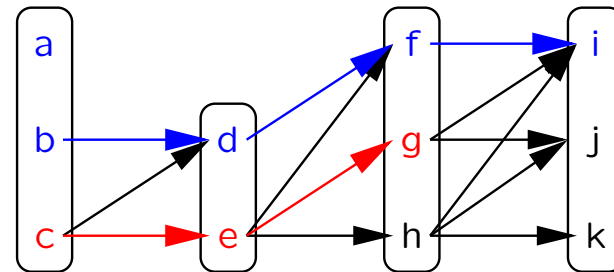
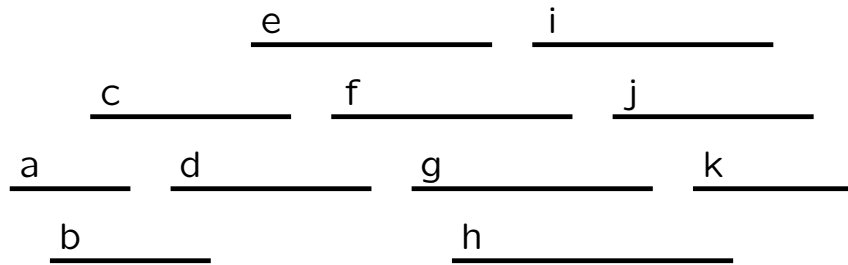
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4		3			2		

Stack: a b d f i c e

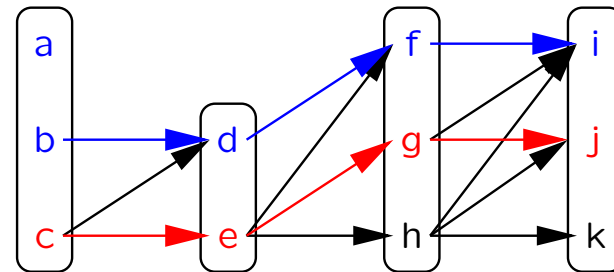
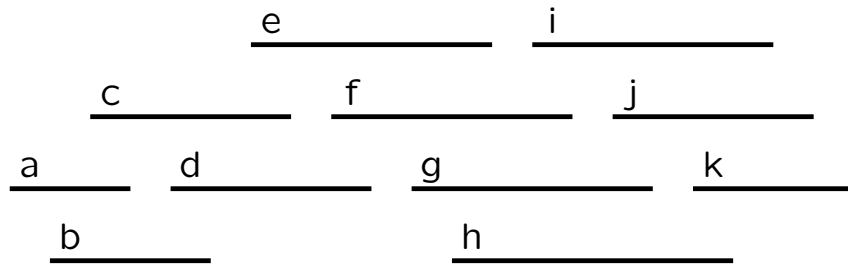
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4		3			2		

Stack: a b d f i c e g

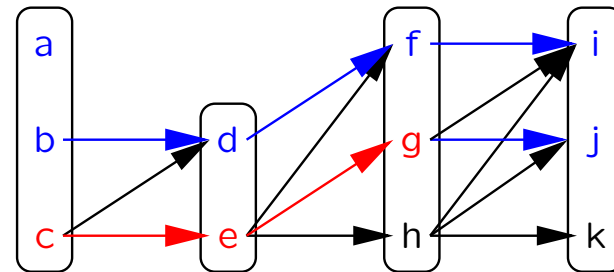
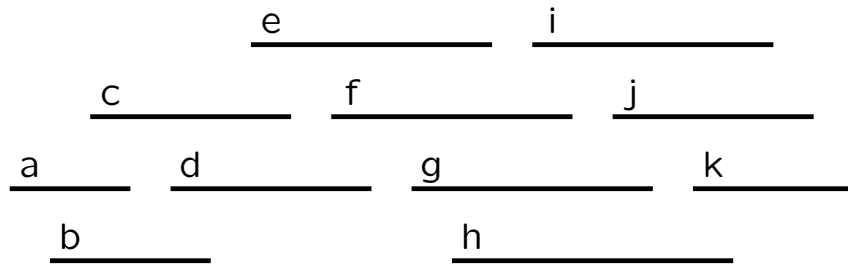
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4		3			2		

Stack: a b d f i c e g j

# An Example

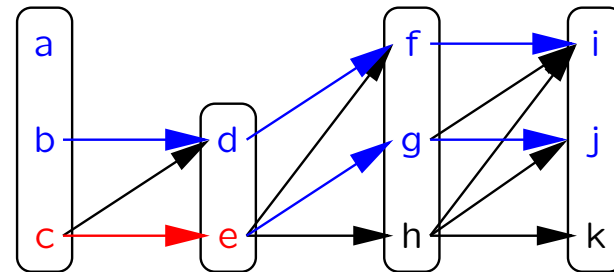
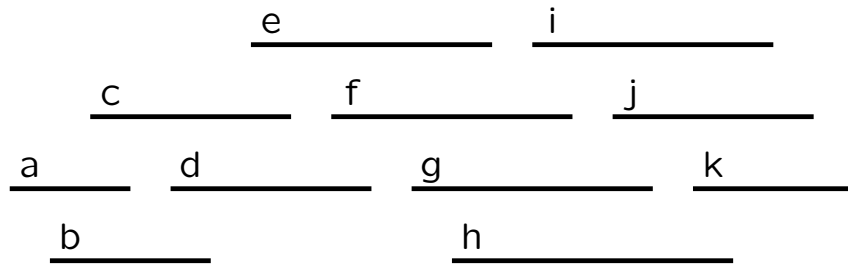


vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4		3			2	6	

Stack: 
~~a~~ ~~b~~ ~~d~~ f i c e g j



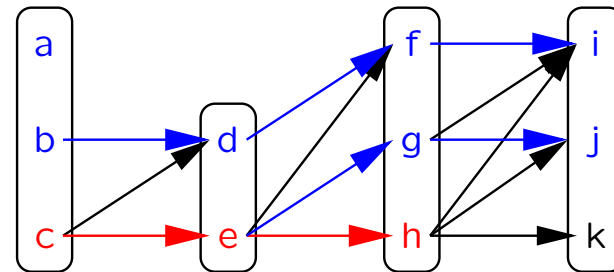
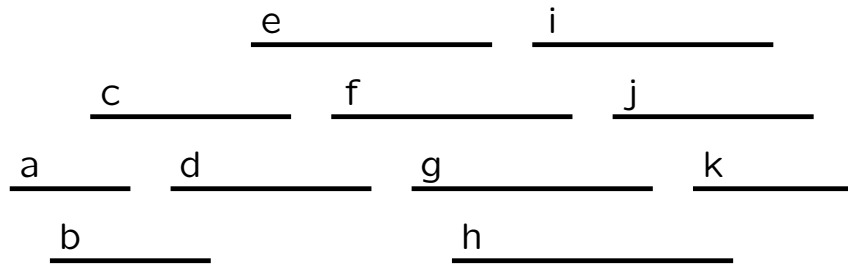
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4		3	7		2	6	

Stack: 
~~a~~ ~~b~~ ~~d~~ f i c e ~~g~~ j

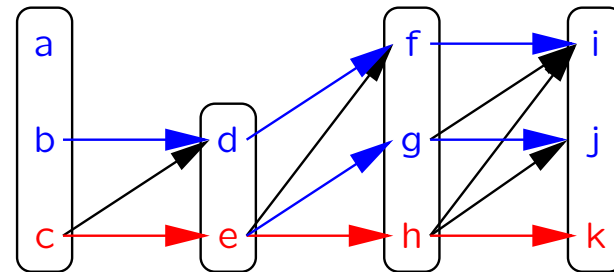
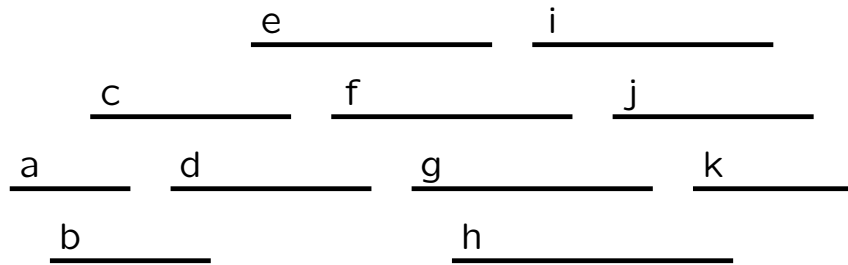
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4		3	7		2	6	

Stack: 
a b d f i c e g j h

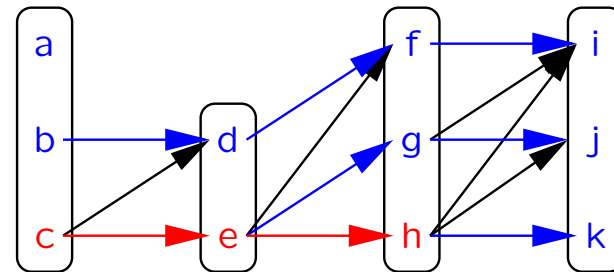
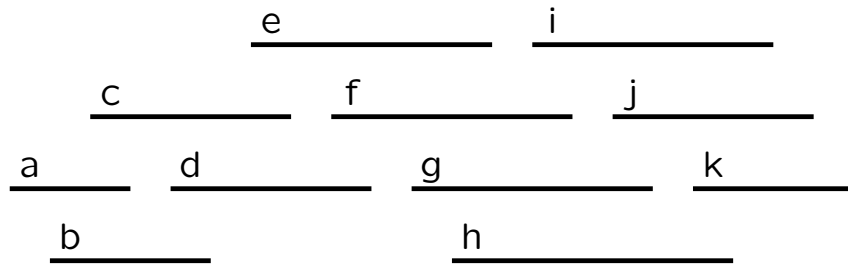
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4		3	7		2	6	

Stack: 
~~a~~ ~~b~~ ~~d~~ f i c e ~~g~~ j h k

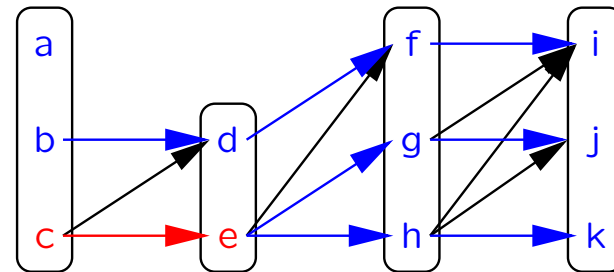
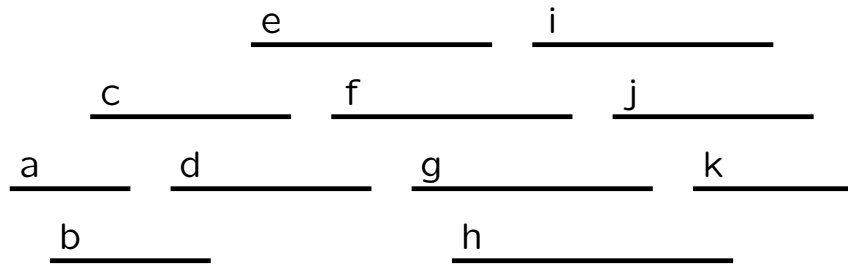
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4		3	7		2	6	8

Stack: a b d f i c e g j h k

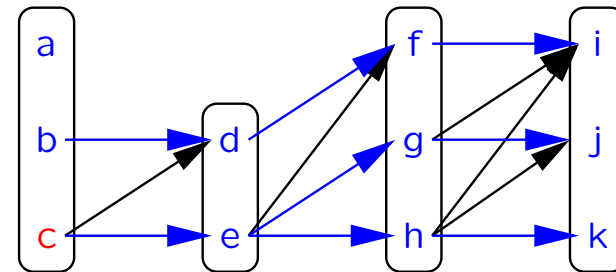
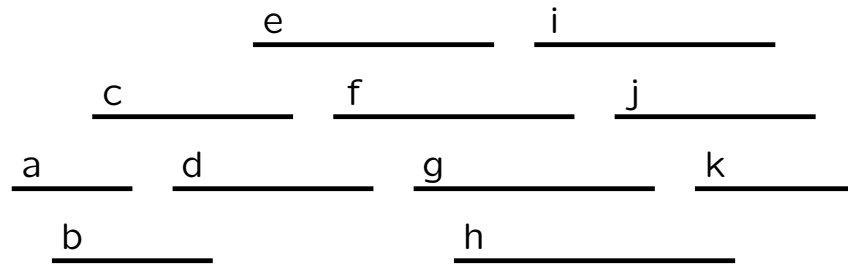
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4	8	3	7	9	2	6	8

Stack: a b d f i c e g j h k

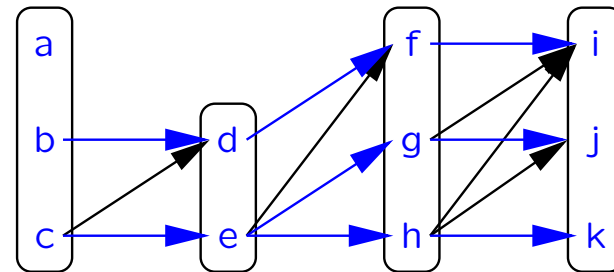
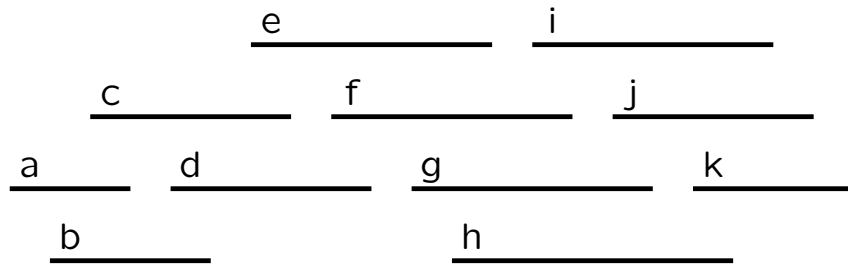
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5		4	10	3	7	9	2	6	8

Stack: a b c f i c e g j h k

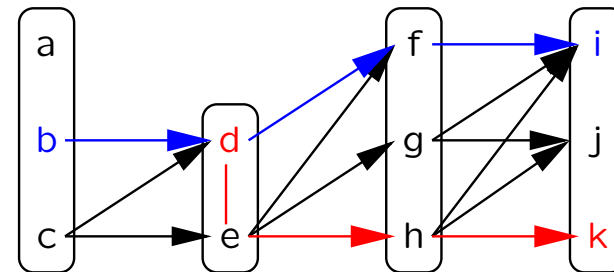
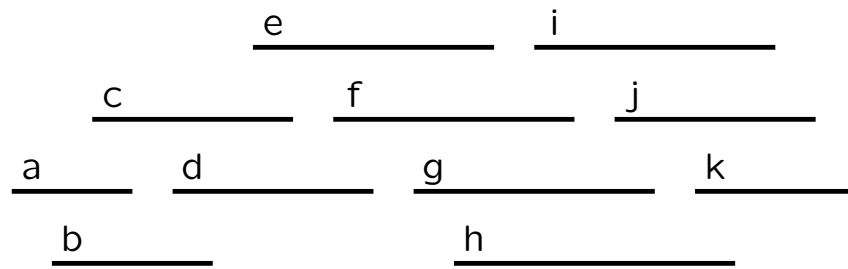
# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5	11	4	10	3	7	9	2	6	8

Stack: a b d f i c e g j h k

# An Example



vertex	a	b	c	d	e	f	g	h	i	j	k
$\lambda$	1	1	1	2	2	3	3	3	4	4	4
$\pi$	1	5	11	4	10	3	7	9	2	6	8

$$\text{dist}(b, i) = \lambda(i) - \lambda(b) + 1 - \text{adj}_{H^t}(b, i) = 4 - 1 + 1 - 1 = 3$$

$$\text{dist}(d, k) = \lambda(k) - \lambda(d) + 1 - \text{adj}_{H^t}(d, k) = 4 - 2 + 1 - 0 = 3$$



## Result for proper interval graphs

**Theorem 4** *The family of  $n$ -vertex proper interval graphs enjoys a distance labeling scheme using labels of length*

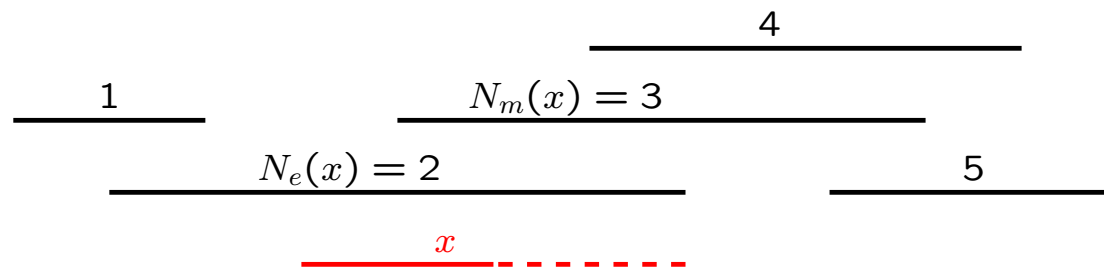
$$2 \lceil \log n \rceil \text{ bits}$$

*and the distance decoder has a  $\mathcal{O}(1)$  time complexity.*

*Moreover, given the sorted list of intervals, all the labels can be computed in  $\mathcal{O}(n)$  time.*

## Generalization to Interval graphs

Extending some intervals of a layout  $\mathcal{I}$  of an interval graph  $G$ , we compute a layout  $\mathcal{I}'$  of a proper interval graph  $G'$ .



If  $\mathcal{I}(x) \subset \mathcal{I}(y)$  then the enclosed neighbor  $N_e(x)$  is the neighbor of  $x$  such that  $\mathcal{I}(x) \subset \mathcal{I}(N_e(x))$  and  $R(N_e(x))$  is maximum

We set  $\mathcal{I}'(x) = [L(x), R(N_e(x))]$

For any vertex  $x$ , the **maximum neighbor**  $N_m(x)$  is the neighbor of  $x$  with  $R(N_m(x))$  maximum.

**Theorem 5** *Let  $x$  and  $y$  be two vertices such that  $R(x) \leq R(y)$ . Then,*

$$dist_G(x, y) = dist_{G'}(N_m(x), y) + 1 - adj_G(x, y)$$

The **label** of the vertex  $x$  is

$$L(x, G) = \langle L(x), R(x), \lambda(x), \pi(x), b(x), \pi(N_m(x)) \rangle$$

where  $b(x)$  is a bit telling whether

$$\lambda(N_m(x)) = \lambda(x) \text{ or } \lambda(N_m(x)) = \lambda(x) + 1$$

## Result for interval graphs

**Theorem 6** *The family of  $n$ -vertex interval graphs enjoys a distance labeling scheme using **labels** of length*

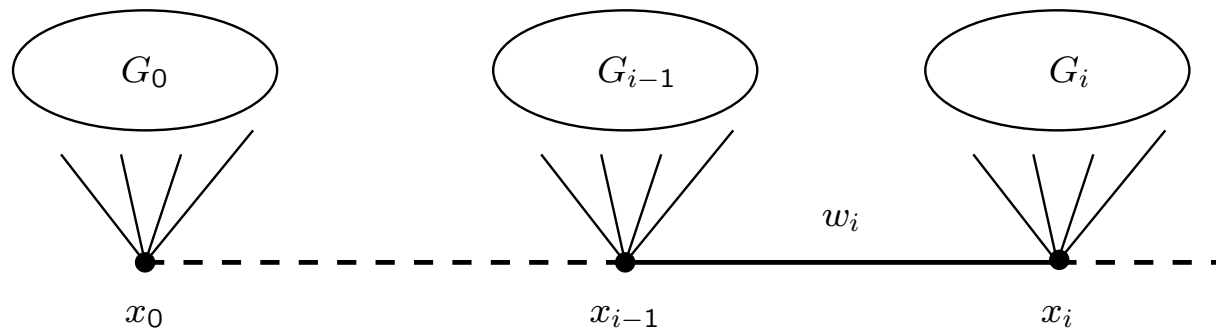
$$5 \lceil \log n \rceil + 3 \text{ bits}$$

*and the **distance decoder** has a  $\mathcal{O}(1)$  time complexity.*

*Moreover, given the sorted list of intervals, all the labels can be computed in  $\mathcal{O}(n)$  time.*

(which is better in both **time** and **space** than the distance matrix)

## Lower bounds (hints)



**Lemma 1** Any distance labeling scheme on  $n$ -vertex interval graphs requires a label of length

$$\frac{1}{N} \log I(N) + \log N - 9 \text{ where } N = \lfloor n / \log n \rfloor$$

**Theorem 7** *The number  $I(n)$  of labeled  $n$ -vertex connected interval graphs satisfies*

$$\frac{1}{n} \log I(n) \geq 2 \log n - \log \log n - \mathcal{O}(1)$$

$\Rightarrow$  *there are  $2^{\Omega(n \log n)}$  unlabeled  $n$ -vertex interval graphs.*

**Theorem 8** *Any distance labeling scheme on the the family of  $n$ -vertex interval graphs requires labels of length at least*

$$3 \log n - 4 \log \log n.$$

**Theorem 9** *Any distance labeling scheme on the the family of  $n$ -vertex **proper interval graphs** requires labels of length at least  $2 \log n - 2 \log \log n - \mathcal{O}(1)$ .*