Graphs

Johan Montelius

KTH

VT26

a linked list

- each node has at most one link

a linked list

- each node has at most one link
- add, remove, lookup

a linked list

- each node has at most one link
- add, remove, lookup
- sorted, unsorted

a linked list

- each node has at most one link
- add, remove, lookup
- sorted, unsorted
- access to last node
- circular?

a tree

- each node has almost two links

a tree

- each node has almost two links
- one root node

a tree

- each node has almost two links
- one root node
- nodes at level n can only have links to level $n + 1$
- ergo: no circular paths
- add, remove, lookup

a tree

- each node has almost two links
- one root node
- nodes at level n can only have links to level $n + 1$
- ergo: no circular paths
- add, remove, lookup
- sorted, unsorted

a tree

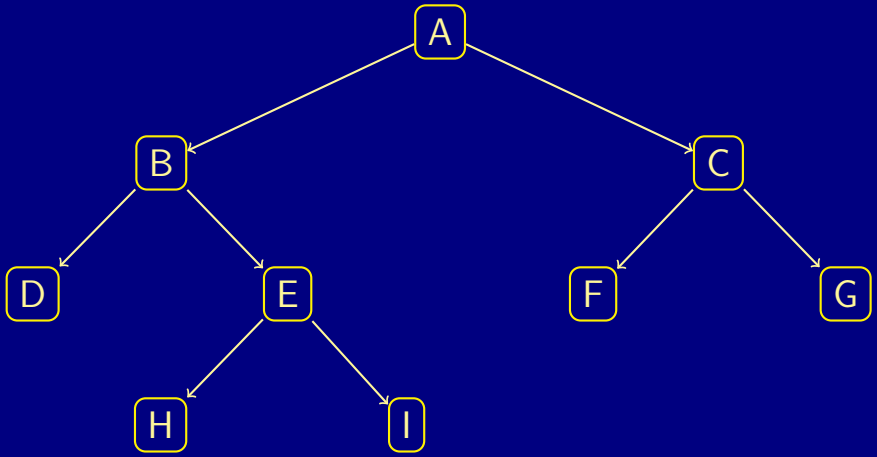
- each node has almost two links
- one root node
- nodes at level n can only have links to level $n + 1$
- ergo: no circular paths
- add, remove, lookup
- sorted, unsorted
- implemented using an array?

a tree

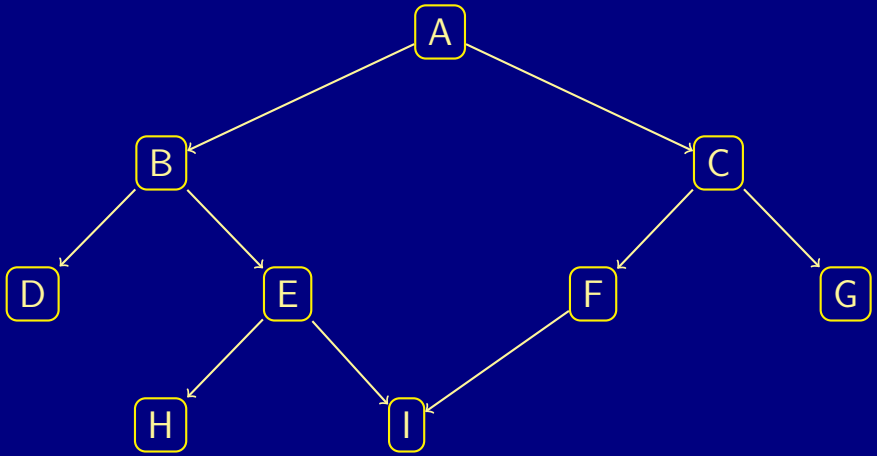
- each node has almost two links
- one root node
- nodes at level n can only have links to level $n + 1$
- ergo: no circular paths
- add, remove, lookup
- sorted, unsorted
- implemented using an array?

let's relax the rules

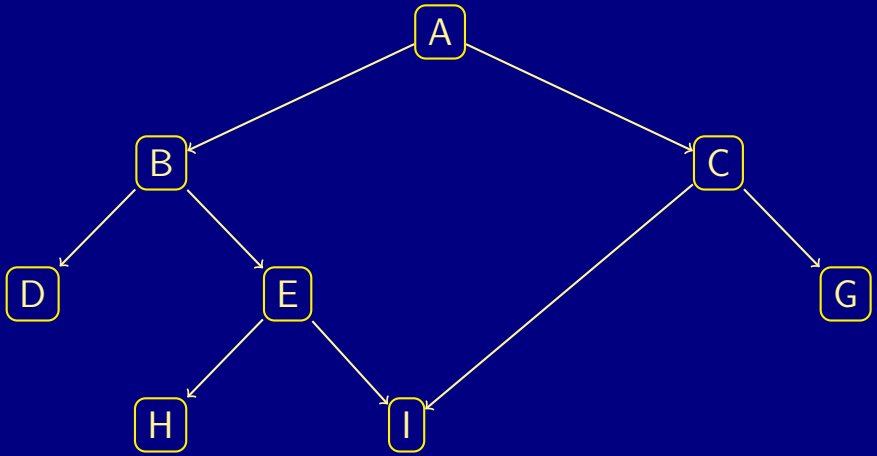
relaxing the rules



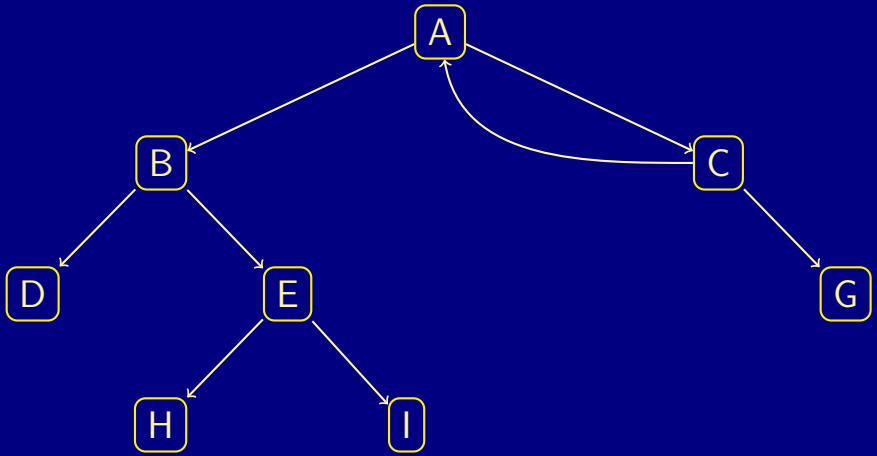
relaxing the rules



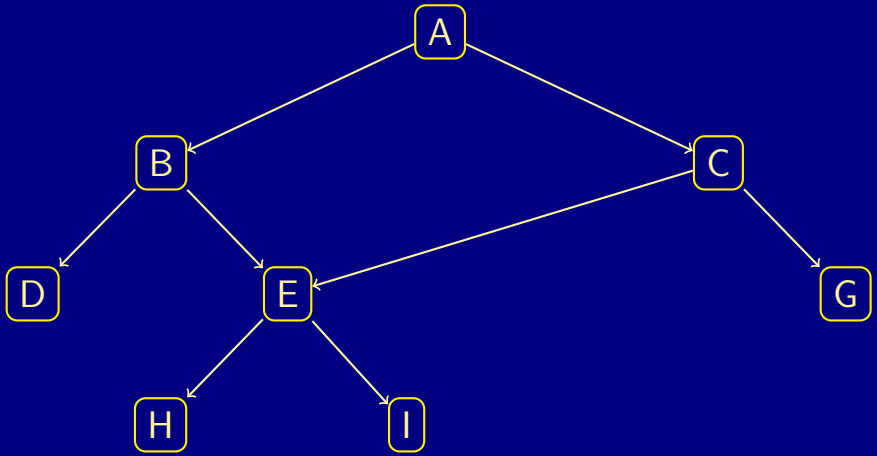
relaxing the rules



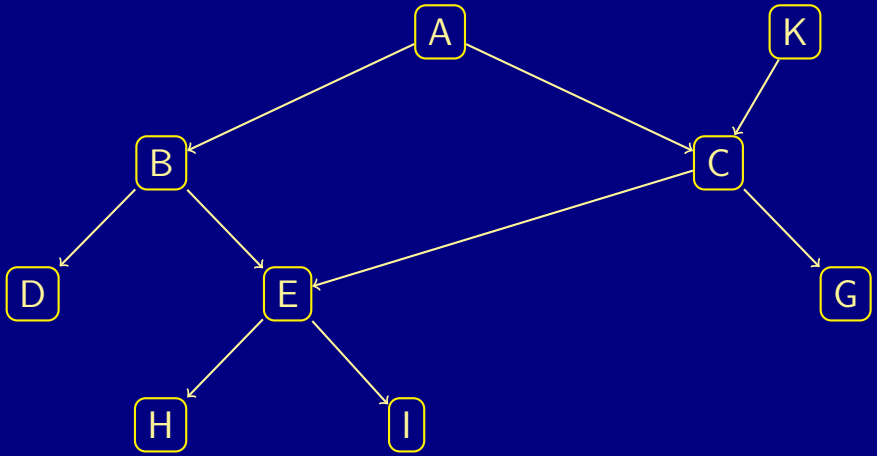
relaxing the rules



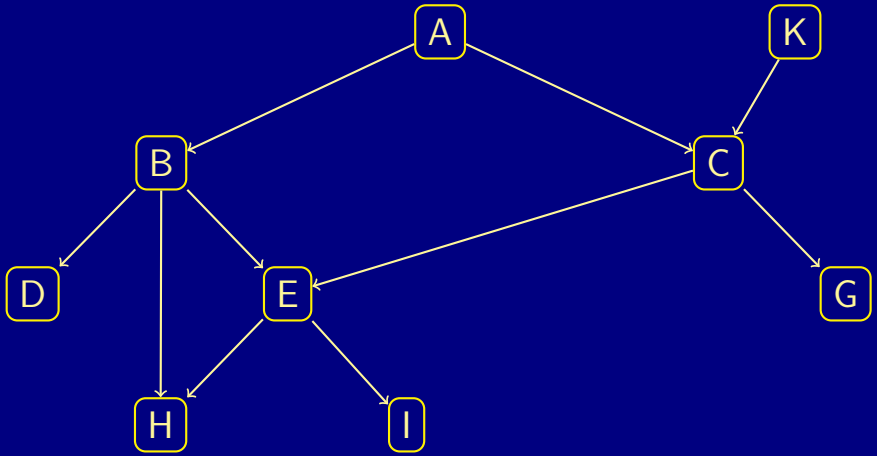
DAG



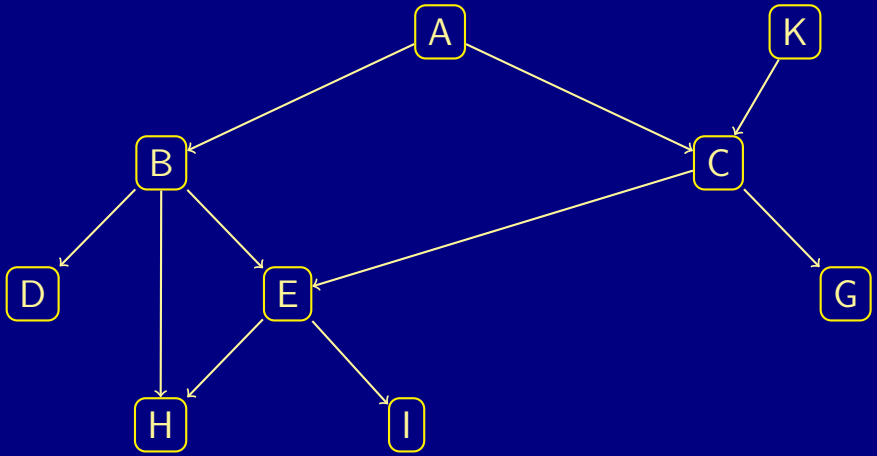
DAG



DAG



DAG



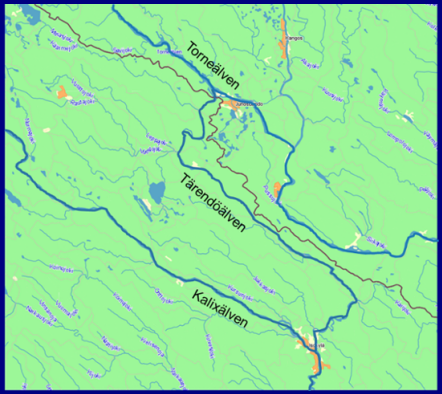
Directed Acyclic Graph

rivers

Is a river a DAG?

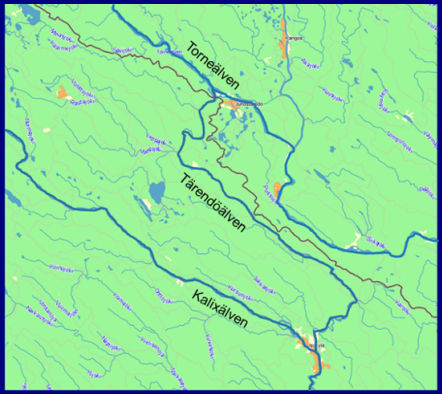
rivers

Is a river a DAG?



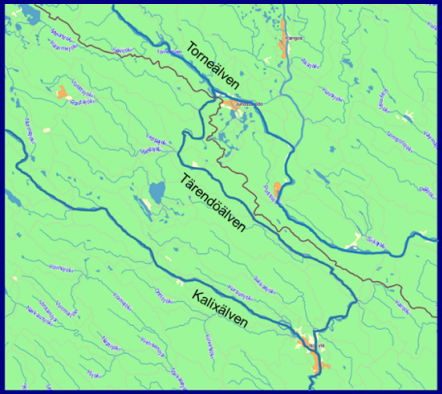
rivers

Is a river a DAG?
Is a tree a DAG?



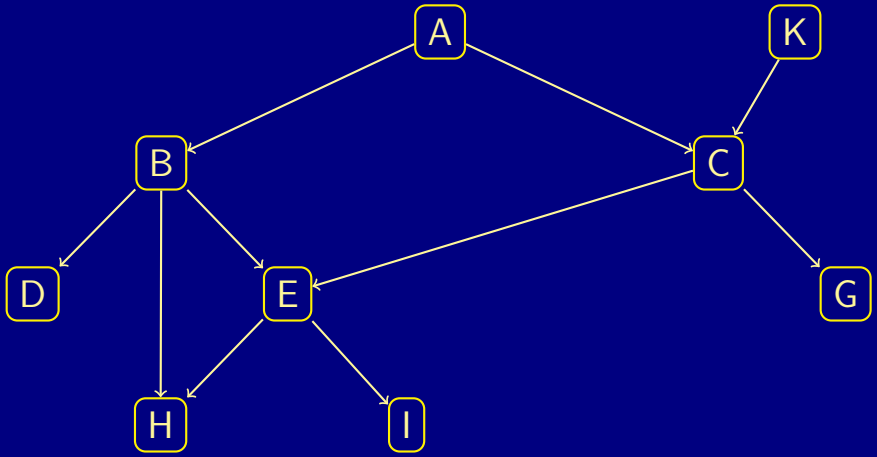
rivers

Is a river a DAG?
Is a tree a DAG?

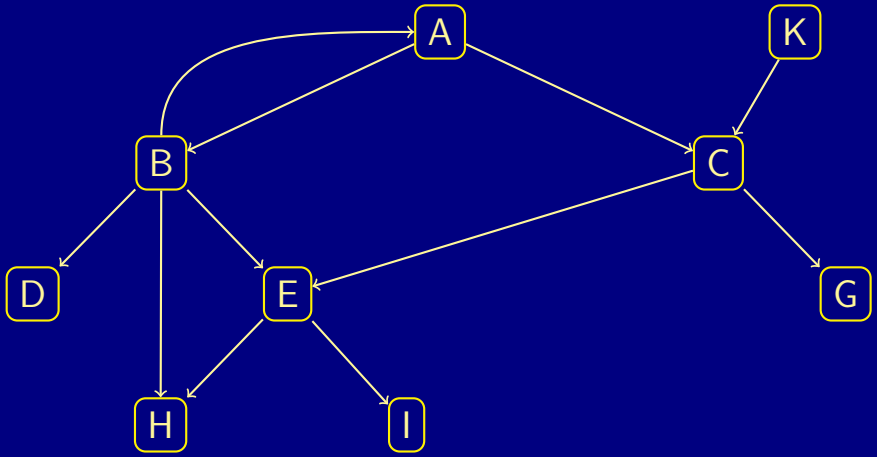




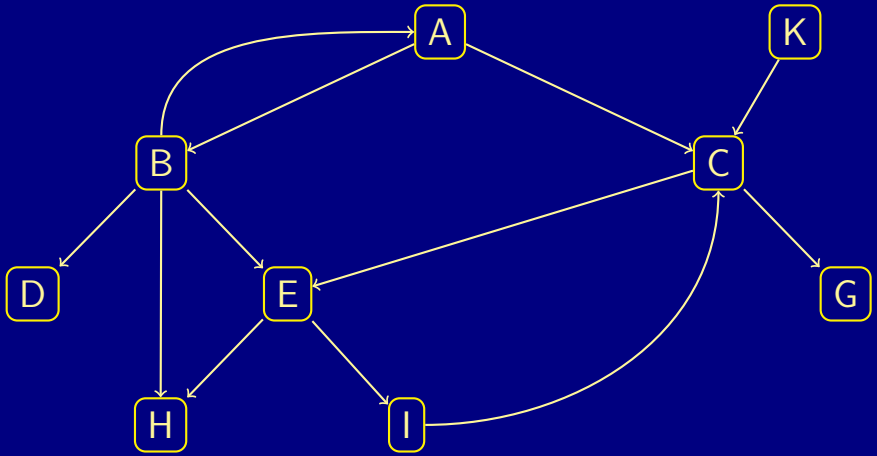
Directed graph



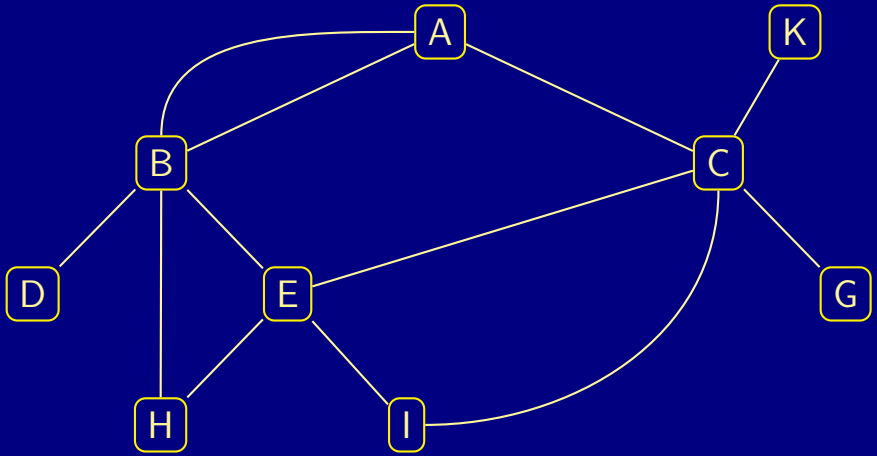
Directed graph



Directed graph



undirectional graph



trains in Sweden

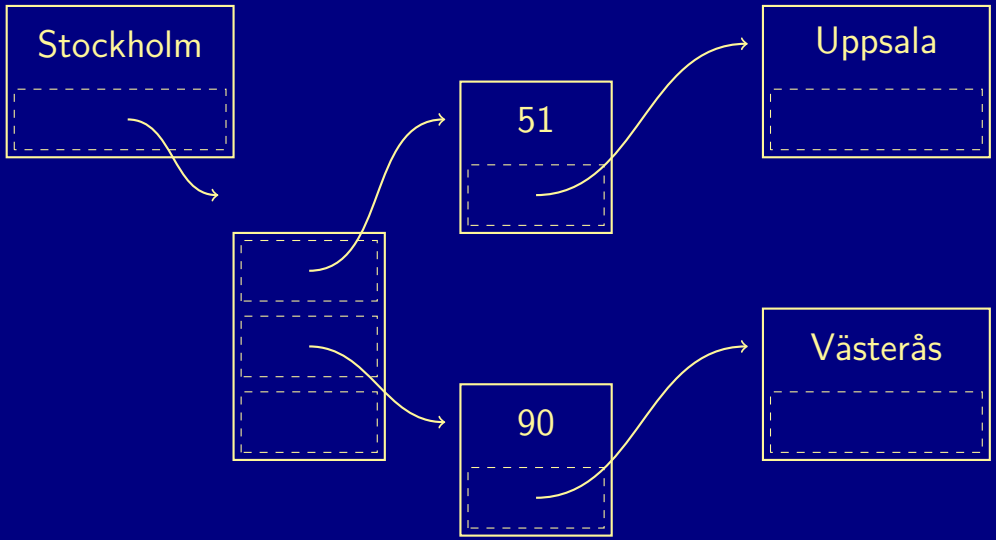


represent the graph

```
typedef struct conn {
  struct city *dst;
  uint dist;
  struct conn *next;
} conn;
```

```
typedef struct city {
   char *name;
   int id;
   conn *connections;
    :
} city;
```

the graph



the naive solution

What is the shortest path from Malmö to Stockholm?

the naive solution

What is the shortest path from Malmö to Stockholm?
Set a maximum distance of the path.

the naive solution

What is the shortest path from Malmö to Stockholm?
Set a maximum distance of the path.

- Do we have more time left?

the naive solution

What is the shortest path from Malmö to Stockholm?
Set a maximum distance of the path.

- Do we have more time left?
- Are we at the destination?

the naive solution

What is the shortest path from Malmö to Stockholm?
Set a maximum distance of the path.

- Do we have more time left?
- Are we at the destination?
- For each of the direct connected cites:

the naive solution

What is the shortest path from Malmö to Stockholm?
Set a maximum distance of the path.

- Do we have more time left?
- Are we at the destination?
- For each of the direct connected cites:
  - set the maximum distance allowed and

the naive solution

What is the shortest path from Malmö to Stockholm?
Set a maximum distance of the path.

- Do we have more time left?
- Are we at the destination?
- For each of the direct connected cites:
  - set the maximum distance allowed and
  - find the shortest path from the city to the destination.

the naive solution

What is the shortest path from Malmö to Stockholm?
Set a maximum distance of the path.

- Do we have more time left?
- Are we at the destination?
- For each of the direct connected cites:
  - set the maximum distance allowed and
  - find the shortest path from the city to the destination.
- Return the shortest distance found (or null).

the naive solution

What is the shortest path from Malmö to Stockholm?
Set a maximum distance of the path.

- Do we have more time left?
- Are we at the destination?
- For each of the direct connected cites:
  - set the maximum distance allowed and
  - find the shortest path from the city to the destination.
- Return the shortest distance found (or null).

the path solution

the path solution

As before but keep a trail of cities that you have past.

the path solution

As before but keep a trail of cities that you have past.

- Do we have more time left?

the path solution

As before but keep a trail of cities that you have past.

- Do we have more time left?
- Are we at the destination?

the path solution

As before but keep a trail of cities that you have past.

- Do we have more time left?
- Are we at the destination?
- Is the city in the path?

the path solution

As before but keep a trail of cities that you have past.

- Do we have more time left?
- Are we at the destination?
- Is the city in the path?

  :

improvement

improvement

If you have found a path with a distance d,

improvement

If you have found a path with a distance d,
then any other path should be shorter than the found.

assignment

assignment

- build the graph from specification

assignment

- build the graph from specification
- naive search given max distance

assignment

- build the graph from specification
- naive search given max distance
- preventing loops using path

assignment

- build the graph from specification
- naive search given max distance
- preventing loops using path
- improve search by remembering found distances

assignment

- build the graph from specification
- naive search given max distance
- preventing loops using path
- improve search by remembering found distances
- realizing that something needs to be done