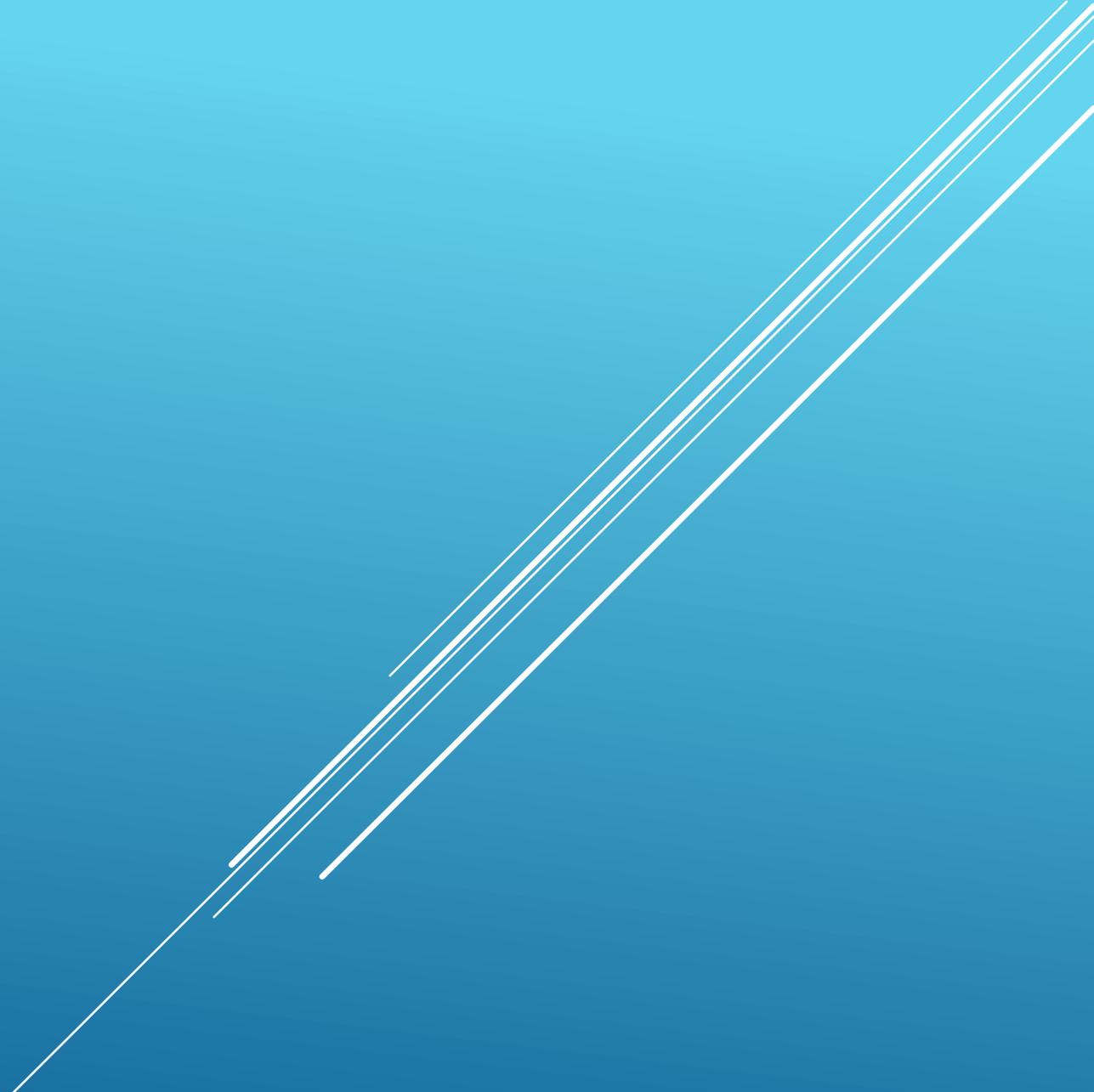


INTERVALLBÄUME

Tanja Lehenauer, Besart Sylejmani

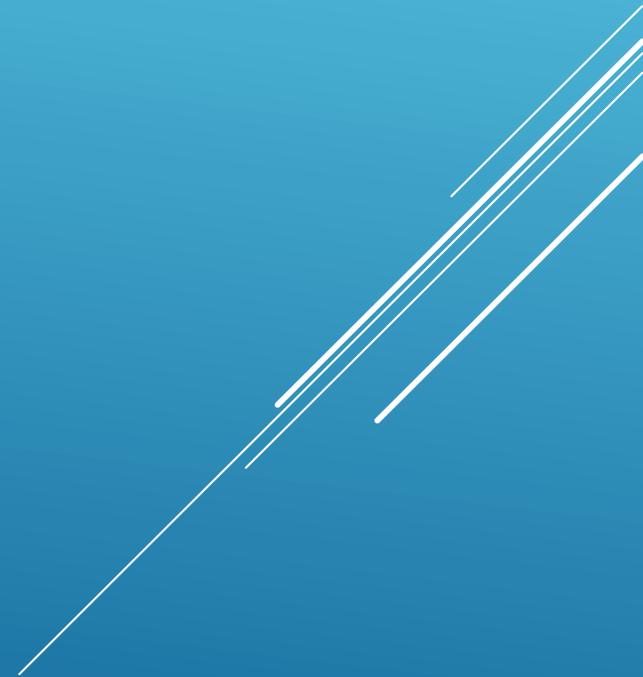


- ▶ Datenstrukturen in der Informatik
 - ▶ Baumstrukturen
- ▶ Warum Intervallbäume?
- ▶ Centered Interval Tree
 - ▶ Konstruktion
 - ▶ Suchen eines Punktes
 - ▶ Suchen eines Intervalls
 - ▶ Adaption

INHALT

- ▶ Eine Datenstruktur ist ein Objekt zur Speicherung und Verwaltung von Daten
- ▶ Daten sind in einer bestimmten Art und Weise angeordnet
-> effizienter Zugriff
- ▶ **BAUMSTRUKTUREN**

DATENSTRUKTUREN IN DER INFORMATIK

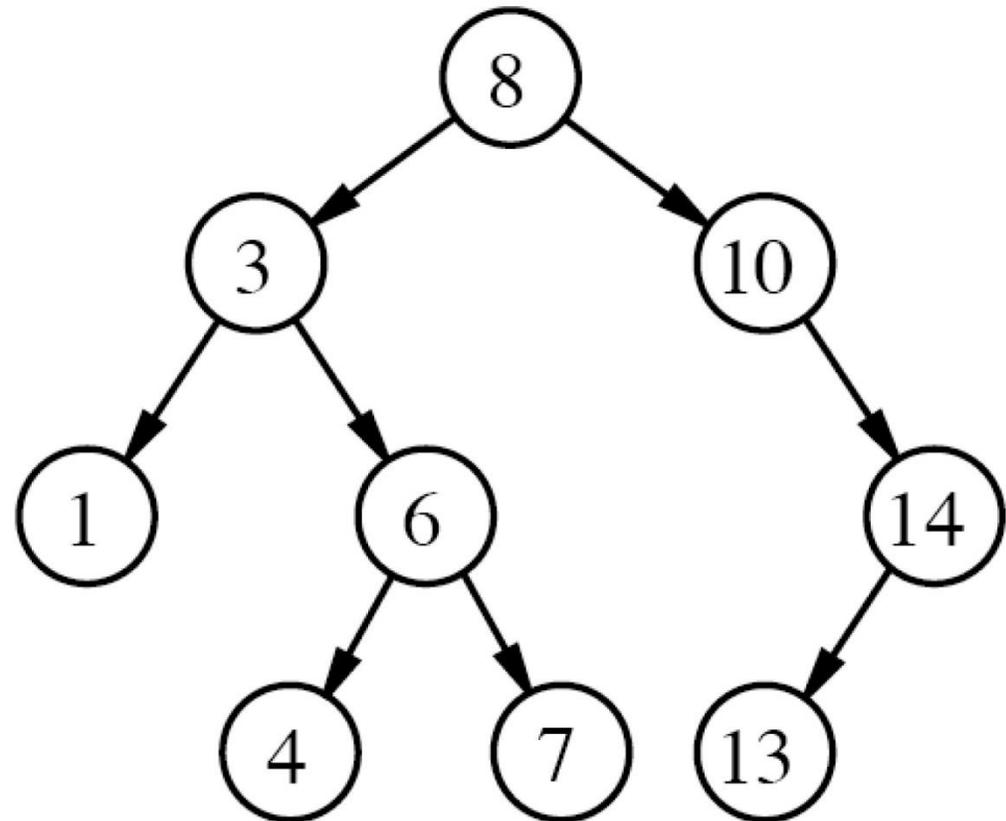


- ▶ Informatik: Out-Trees
- ▶ Wurzel, Knoten, Blatt
- ▶ Elemente stehen in Eltern-Kind-Beziehung zueinander
- ▶ Meistverwendetste Datenstrukturen in der Informatik
-> viele Unterteilungen
- ▶ Geordnete Bäume

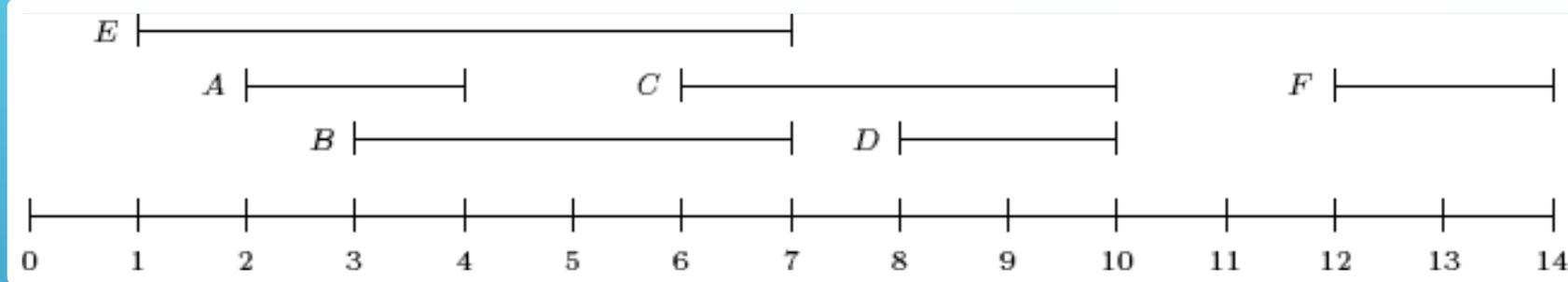
BAUMSTRUKTUREN

Source:

http://users.ics.forth.gr/~kateveni/225/09a/binary_tree.jpg



Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>

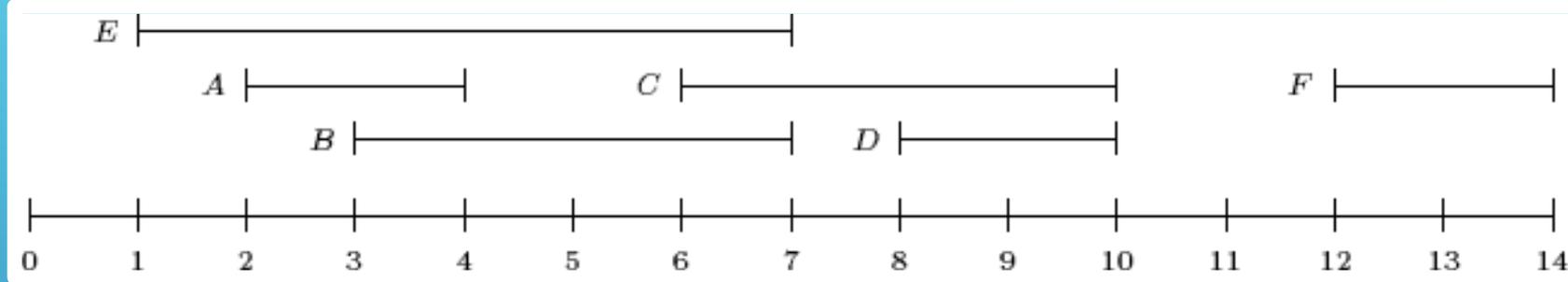


- ▶ Geordnete Datenstruktur zur Speicherung von Intervallen
- ▶ Intervall = Paar geordneter reeller Zahlen
- ▶ Für Ereignisse, die eine zusammenhängende Zeitspanne in Anspruch nehmen
- ▶ In welchen Zeitspannen hat ein bestimmtes Ereignis statt gefunden?

Welche Zeitspannen schneiden sich mit einer bestimmten, gegebenen Zeitspanne?

WARUM INTERVALLBÄUME

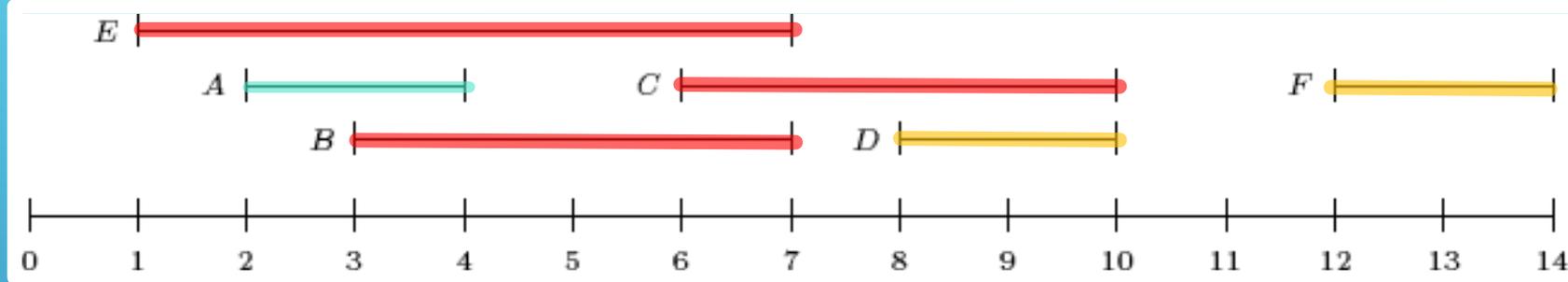
Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>



- ▶ Alle Intervalle durchgehen?
-> zu ineffektiv
- ▶ Binärbaum?
-> 2 Bäume und das Ergebnis zusammenführen, kommt im Aufwand Brute-Force gleich
- ▶ Antwort: INTERVALLBÄUME!
-> **Centered Interval Tree**

WARUM INTERVALLBÄUME

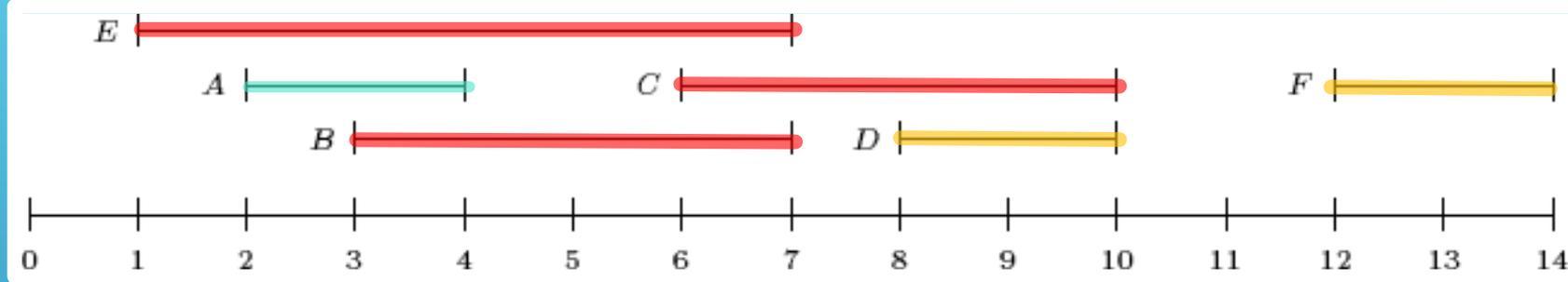
Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>



- ▶ Bereich aller Intervalle in der Mitte teilen
-> Mittelpunkt C
- ▶ 3 Arten von Intervallen
 - ▶ zur Gänze links von C -> *i_left*
 - ▶ zur Gänze rechts von C -> *i_right*
 - ▶ C überlappend -> *i_center*

KONSTRUKTION

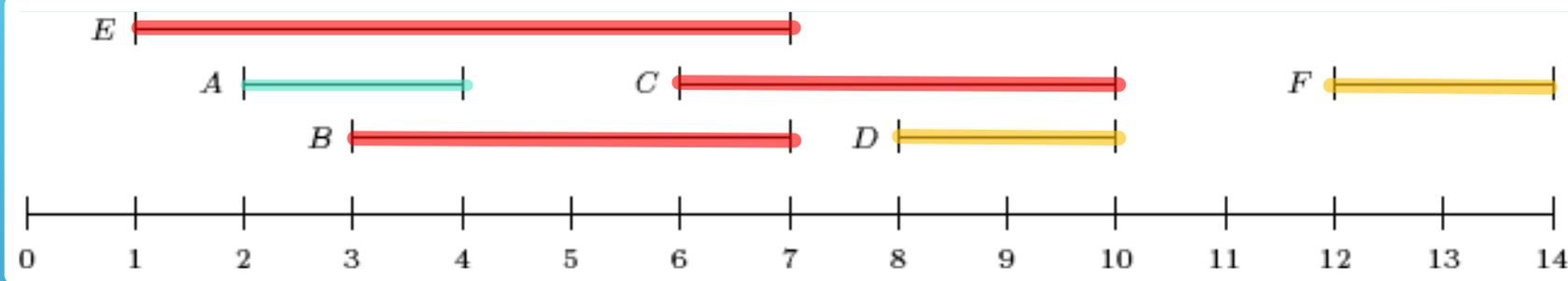
Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>



- ▶ Selbiges mit `i_left` und `i_right`, bis keine Intervalle mehr übrig bleiben
- ▶ Mit `i_center` verfahren wir gesondert:
 - ▶ Werden in 2 Listen gespeichert, die mit dem Knoten verlinkt sind
 - ▶ 1x nach Anfangspunkt
 - ▶ 1x nach Endpunkt

KONSTRUKTION

Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>

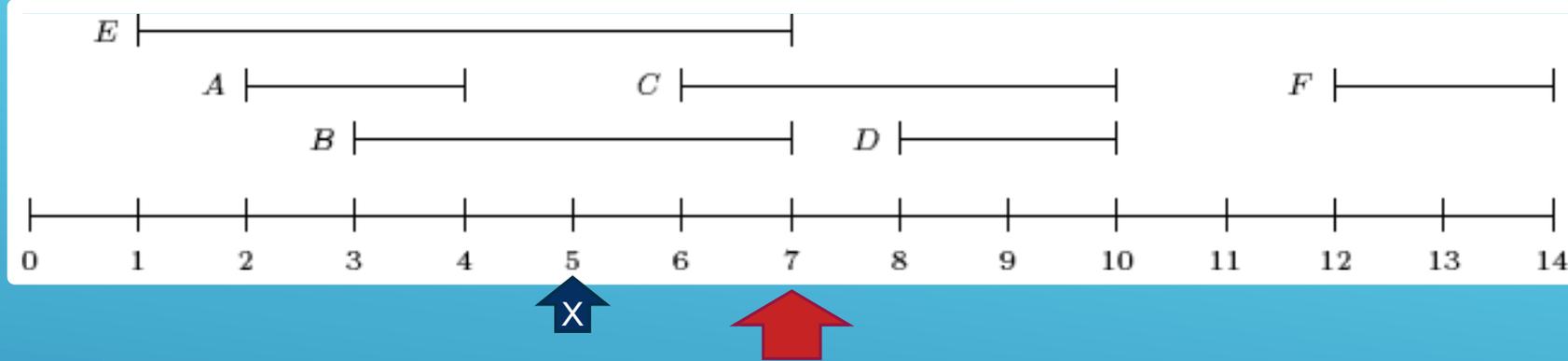


▶ Jeder Knoten speichert also:

- ▶ einen Mittelpunkt *C*
- ▶ einen Pointer zu einem Knoten, der alle Intervalle enthält die zur Gänze links von *C* sind
- ▶ einen Pointer zu einem anderen Knoten, der alle Intervalle enthält die zur Gänze rechts von *C* sind
- ▶ eine Liste, die alle *C* überlappenden Intervalle bezüglich ihres Anfangspunktes speichert
- ▶ eine Liste, die alle *C* überlappenden Intervalle bezüglich ihres Endpunktes speichert

KONSTRUKTION

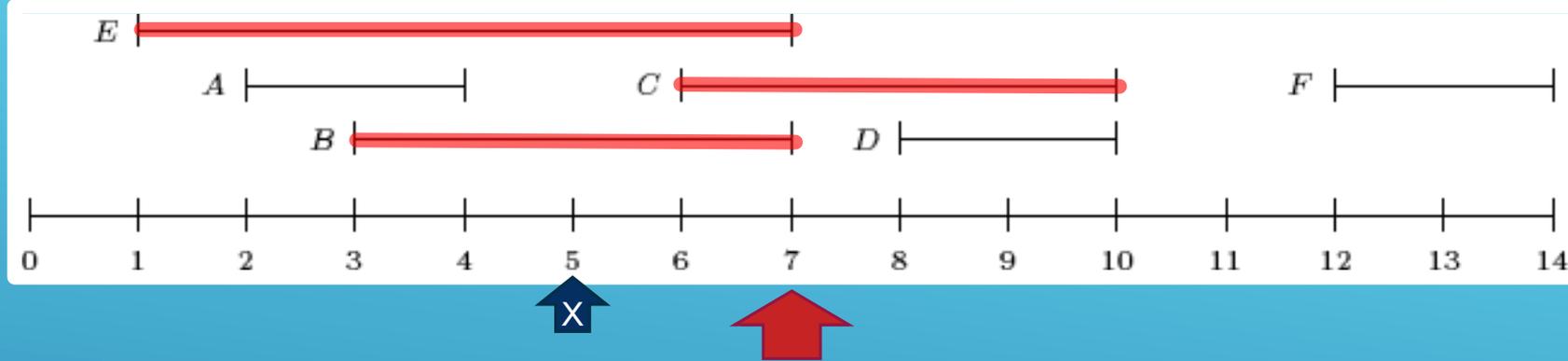
Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>



- ▶ Punkt X?
- ▶ Baum wird ähnlich eines Binärbaums durchgegangen aber mit speziellem Augenmerk auf die die Intervalle in `i_center`, die unseren Mittelpunkt C überlappen
- ▶ Vergleich X mit C
 - ▶ wenn $X < C \rightarrow i_left$
 - ▶ Wenn $X > C \rightarrow i_right$

SUCHANFRAGE: PUNKT

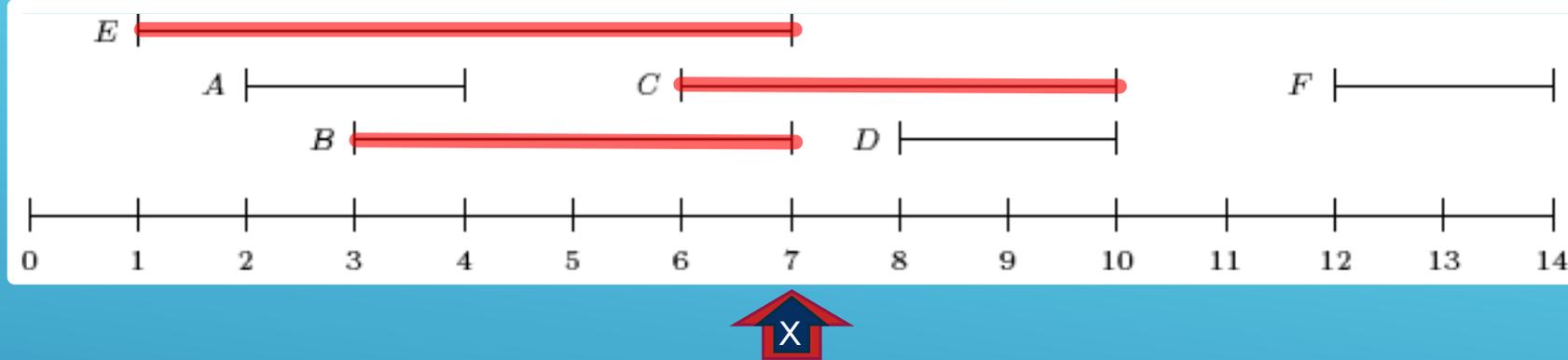
Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>



- ▶ Jeder Knoten, der von der Suchanfrage durchlaufen wird: i_center
- ▶ Annahme: $X < C$
 - ▶ alle Intervalle müssen nach X enden
 - ▶ wir befragen die Liste mit den Anfangspunkten der Intervalle
 - ▶ alle Intervalle die in i_center sind und deren Anfangspunkt kleiner als X ist, überlappen X (weil sie nach C enden müssen)

SUCHANFRAGE: PUNKT

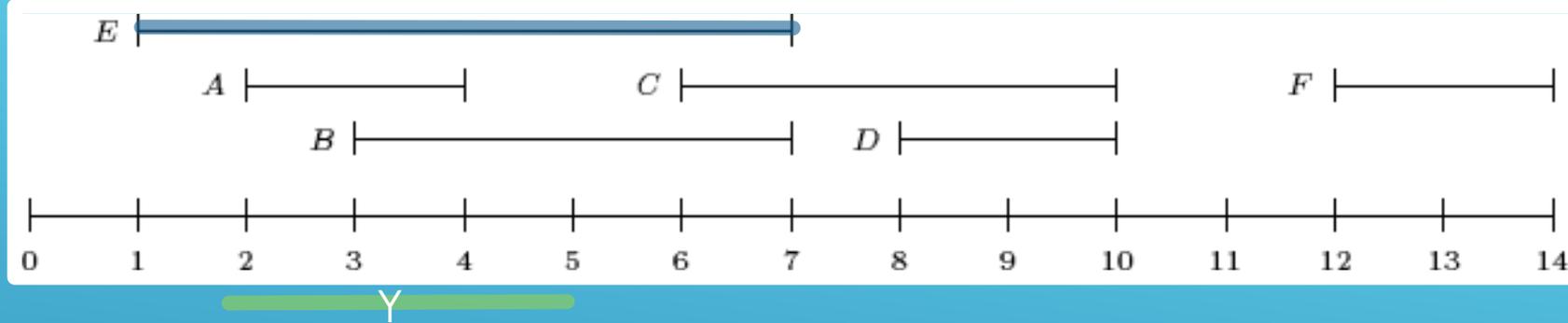
Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>



- ▶ Annahme: $X > C$
 - ▶ im Grunde analog
 - ▶ Liste mit Endpunkten wird befragt
- ▶ Annahme $X = C$
 - ▶ alle Intervalle in i_center werden von X geschnitten
 - ▶ unsere Suche ist beendet

SUCHANFRAGE: PUNKT

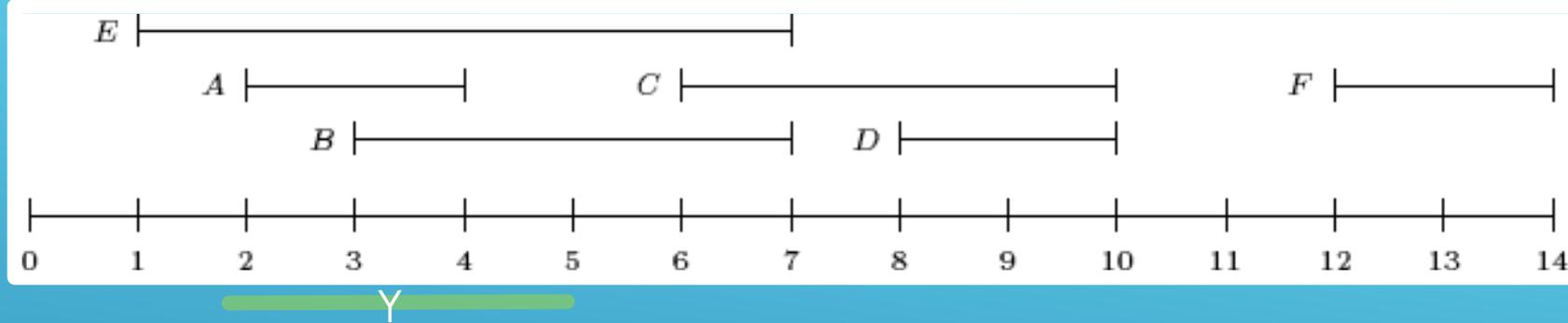
Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>



- ▶ Ein Intervall I ist ein Resultat für Y wenn:
 - ▶ der Anfangspunkt des Intervalls in Y liegt
 - ▶ der Endpunkt des Intervalls in Y liegt
 - ▶ das Intervall Y umschließt

SUCHANFRAGE: INTERVALL

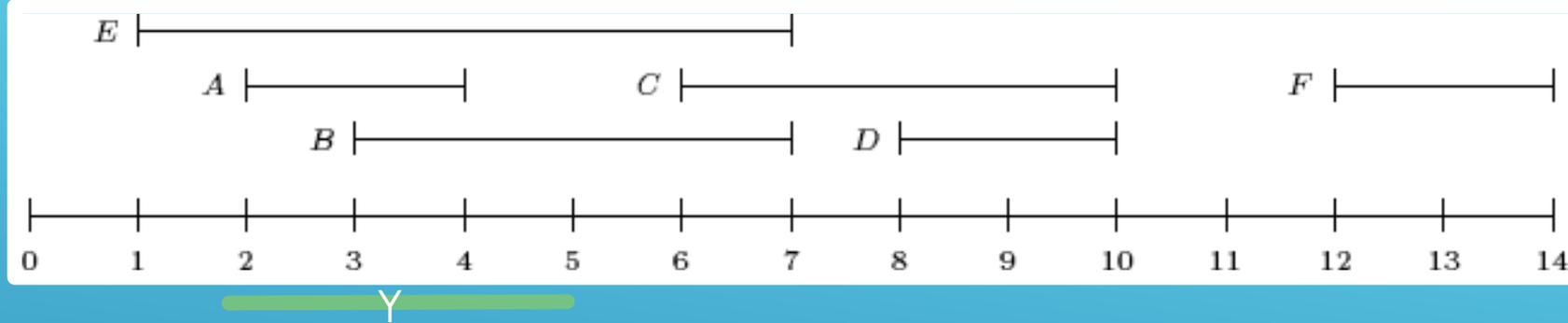
Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>



- ▶ Welche Intervalle beginnen oder enden in Y?
- ▶ Binärbaum mit Anfangs- und Endpunkten
 - ▶ Anfangspunkt Y als Minimum
 - ▶ Endpunkt Y als Maximum
 - ▶ alle Ergebnisse können der Resultatliste zugefügt werden
ABER: Achtung auf Dubletten

SUCHANFRAGE: INTERVALL

Source: <http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/Images/Ps/abb-5-14.gif>



- ▶ Welche Intervalle umschließen Y?
- ▶ Beliebiger Punkt in Y durch den Intervallbaum
 - ▶ alle Intervalle, die durch den Punkt geschnitten werden, sind Resultate
 - ▶ **AUCH HIER: Achtung auf Dubletten**

SUCHANFRAGE: INTERVALL

- ▶ Intervallbaum kann adaptiert werden
 - ▶ Elemente können zugefügt werden
 - ▶ Elemente können gelöscht werden
- ▶ bei Hinzufügen wird der Baum mit der selben Methode wie beim Suchen durchlaufen und in den richtigen Knoten gespeichert
- ▶ Bei Löschen wird der Intervall lokalisiert und entfernt
- ▶ **Problematik: Was passiert, wenn durch den Löschvorgang ein Knoten leer bleibt?**

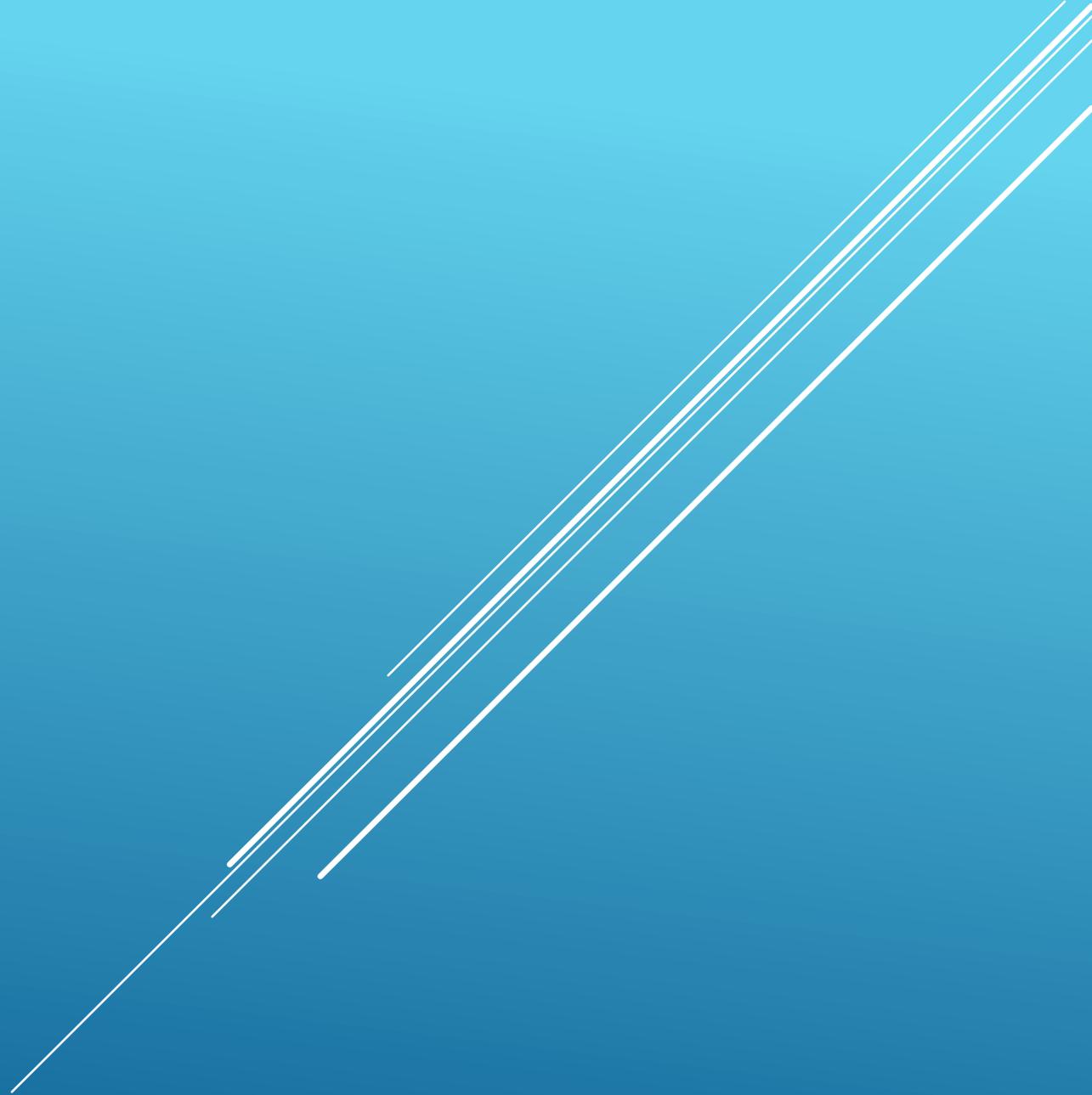
ADAPTION

▶ **Löschung eines Knotens**

- ▶ Ein Intervall kann die Mittelpunkte C mehrerer Knoten überlappen, ist aber gespeichert in dem Knoten, der am nächsten zur Wurzel ist
- ▶ Knoten wird gelöscht -> weiter von der Wurzel entfernterer Knoten wird zur Stelle des gelöschten Knotens „befördert“
- ▶ Resultat: Intervalle zwischen dem gelöschten Knoten und dem Ersatzknoten, könnten direkte Nachfolger des Ersatzelements sein
 - ▶ alle darunterliegenden Intervalle müssen nach Intervallen durchsucht werden, die den neuen Mittelpunkt C des neuen Knoten überlappen
 - ▶ kann zu weiteren leeren Knoten führen -> selber Algorithmus

ADAPTION

FRAGEN?



VIELEN DANK FÜR
DIE AUFMERKSAMKEIT

A decorative graphic consisting of several parallel white lines of varying thicknesses, slanted diagonally from the bottom-left towards the top-right, set against a blue gradient background.