

Übung 8 – Algorithmen II

Yaroslav Akhremtsev, Demian Hespe – yaroslav.akhremtsev@kit.edu, hespe@kit.edu

Mit Folien von Michael Axtmann

http://algo2.iti.kit.edu/AlgorithmenII_WS17.php

Institut für Theoretische Informatik - Algorithmik II

```
    result = current_weight;
    return true;
}

for( EdgeID eid = graph.edgeBegin( current ); eid != graph.edgeEnd( current ); ++eid ){
    const Edge & edge = graph.getEdge( eid );
    COUNTING( statistic_data.inc( DijkstraStatisticData::TOUCHED_EDGES ); )
    if( edge.forward ){
        COUNTING( statistic_data.inc( DijkstraStatisticData::RELAXED_EDGES ); )
        weight new_weight = edge.weight + current_weight;
        GUARANTEE( new_weight >= current_weight, std::runtime_error, "Weight overflow detected." );
        if( !priority_queue.isReached( edge.target ) ){
            COUNTING( statistic_data.inc( DijkstraStatisticData::SUCCESSFULLY_RELAXED_EDGES ); )
            COUNTING( statistic_data.inc( DijkstraStatisticData::REACHED_NODES ); )
            priority_queue.push( edge.target, new_weight );
        } else {
            if( priority_queue.getCurrentKey( edge.target ) > new_weight ){
                COUNTING( statistic_data.inc( DijkstraStatisticData::SUCCESSFULLY_RELAXED_NODES ); )
                priority_queue.decreaseKey( edge.target, new_weight );
            }
        }
    }
}
```

- Woher kommt die Rechenzeit?
 - **Bestendes Token:** Rechenzeit schon bezahlt
 - **Neues Token:** zählt zur amortisierten Laufzeit der Operation
- Kann ein Token ein anderes Token erstellen?

- ✕ Markierung - Kennzeichnet Knoten, die ein Kind verloren haben
- ✖ Doppelte Markierung - Temporäre Markierung. Muss sofort aufgelöst werden
- Neues, unbenutztes Token
- InsertTree-Token - kann genau 1 InsertTree Operation bezahlen
- Bezahltoken - wird zum Bezahlen genutzt
- Union-Token - zahlt für 1 Union Operation; zusätzlich einen InsertTree-Token erstellen
- Mark Token - Zahlt Cut und Markierung des Parents
-   Aktive Knoten
-  Knoten zum Löschen oder Cutten

Fibonacci Heaps - Insert

minPtr



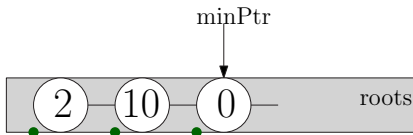
Fibonacci Heaps - Insert



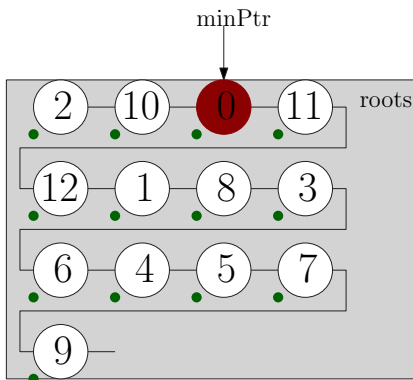
Fibonacci Heaps - Insert



Fibonacci Heaps - Insert



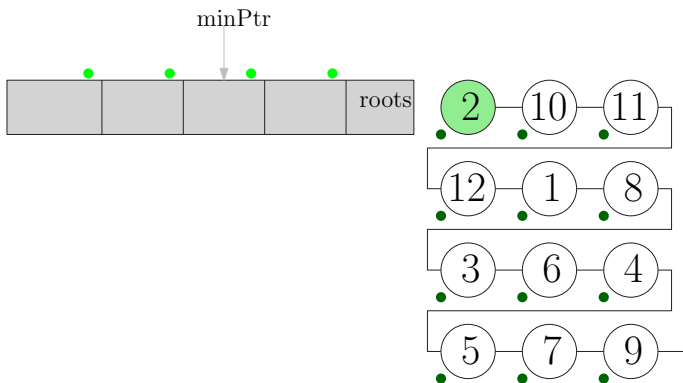
Fibonacci Heaps - Insert



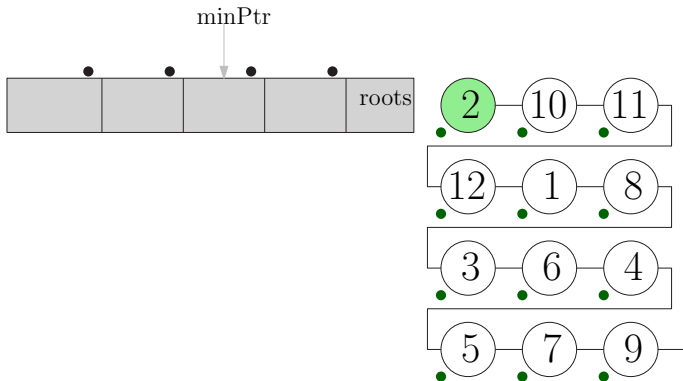
Fibonacci Heaps - Insert

- Insert in konstanter Zeit
- Erzeugt großen Wald einzelner Knoten
- Entspricht ohne **union** nur linearer Liste

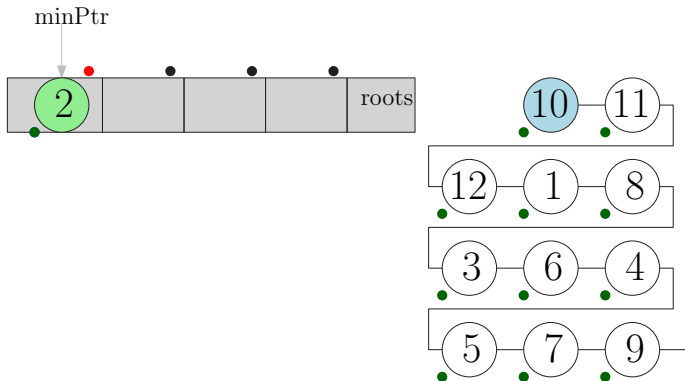
Fibonacci Heaps - Delete Min



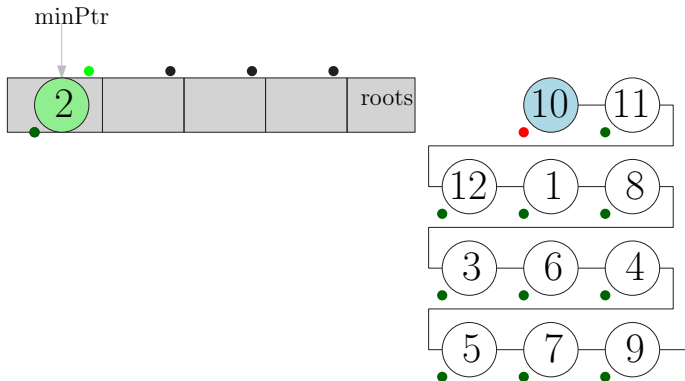
Fibonacci Heaps - Delete Min



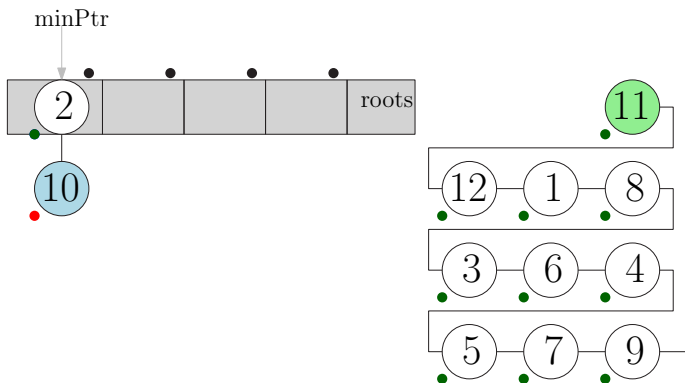
Fibonacci Heaps - Delete Min



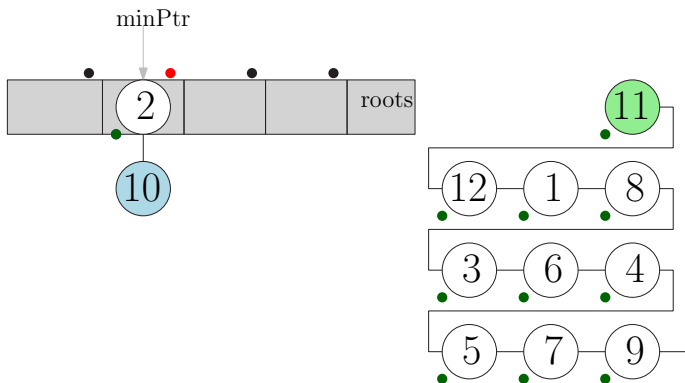
Fibonacci Heaps - Delete Min



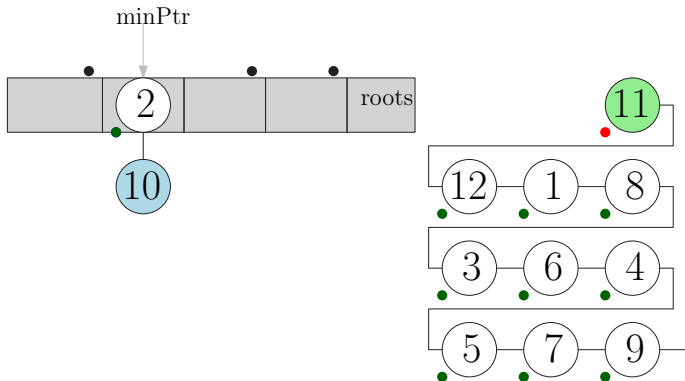
Fibonacci Heaps - Delete Min



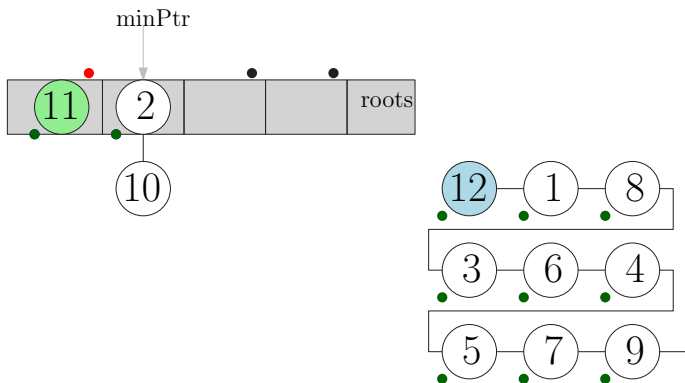
Fibonacci Heaps - Delete Min



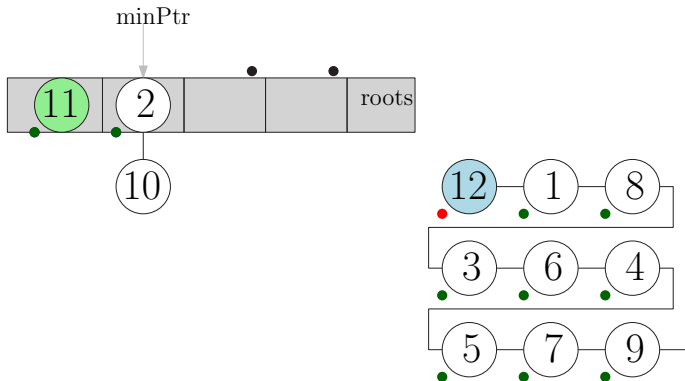
Fibonacci Heaps - Delete Min



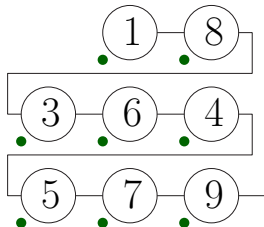
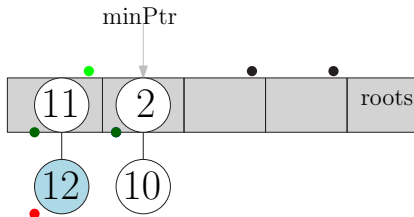
Fibonacci Heaps - Delete Min



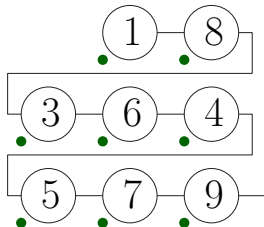
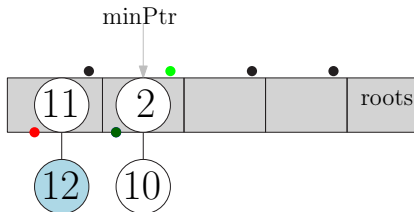
Fibonacci Heaps - Delete Min



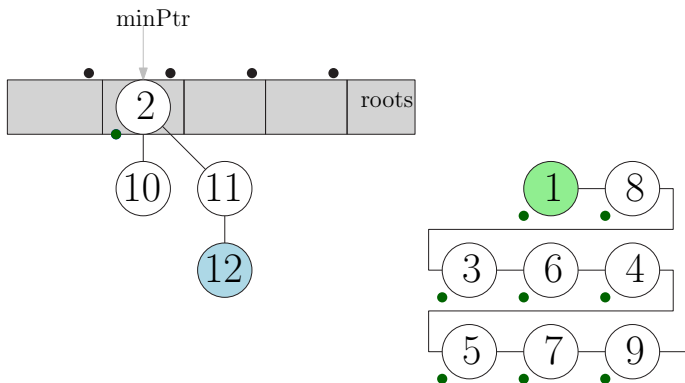
Fibonacci Heaps - Delete Min



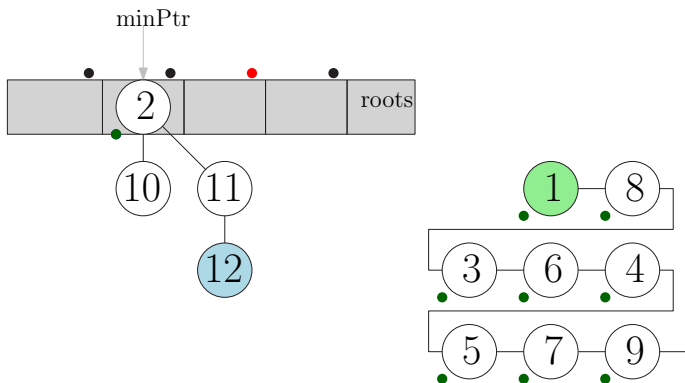
Fibonacci Heaps - Delete Min



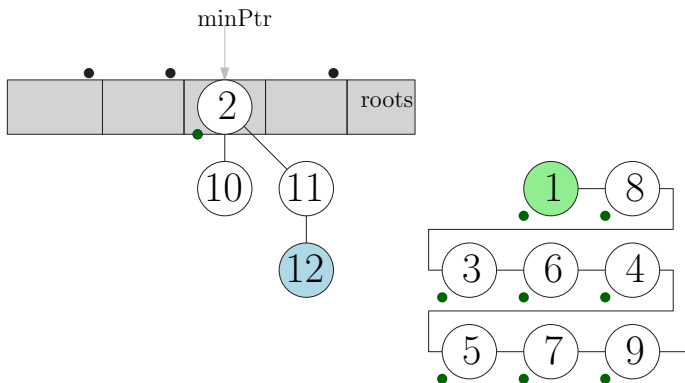
Fibonacci Heaps - Delete Min



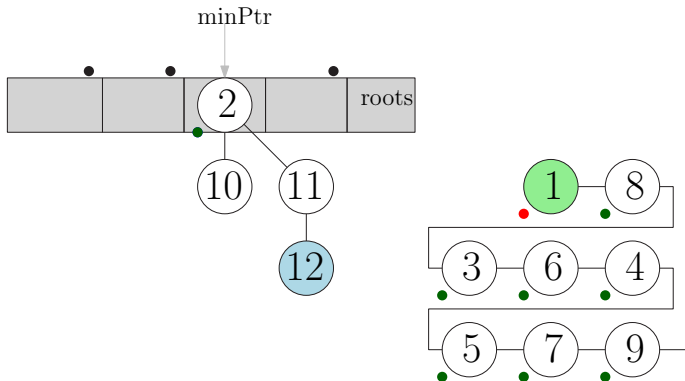
Fibonacci Heaps - Delete Min



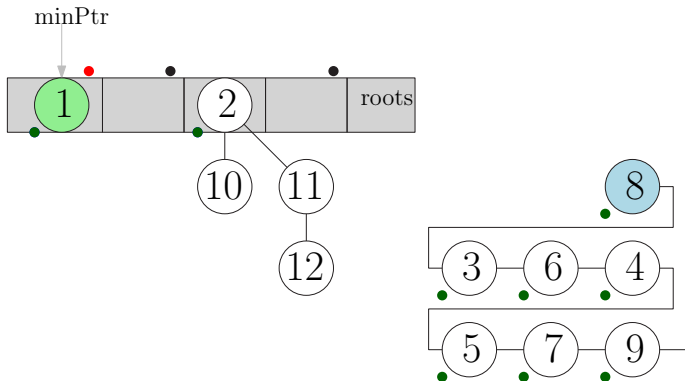
Fibonacci Heaps - Delete Min



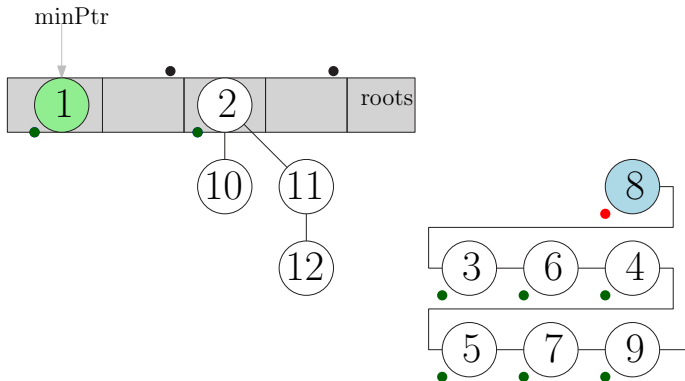
Fibonacci Heaps - Delete Min



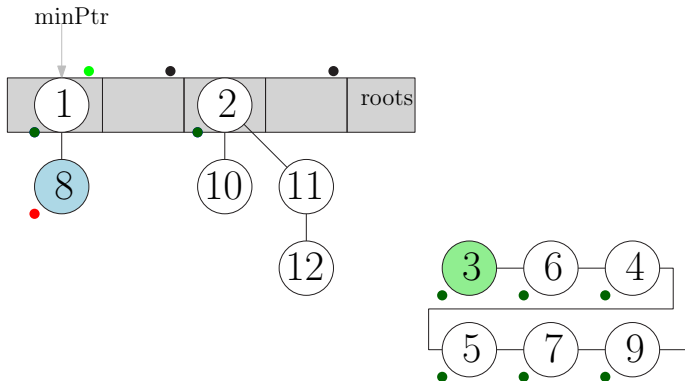
Fibonacci Heaps - Delete Min



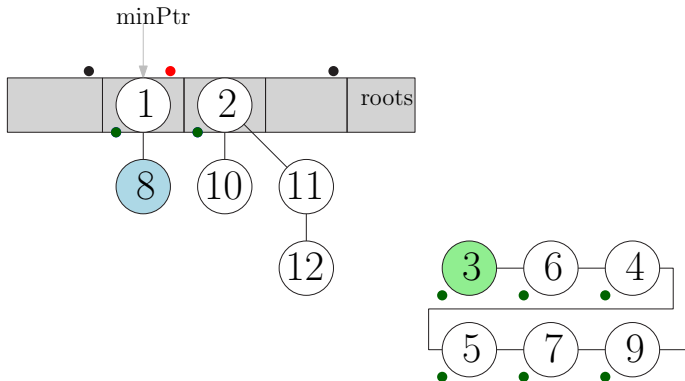
Fibonacci Heaps - Delete Min



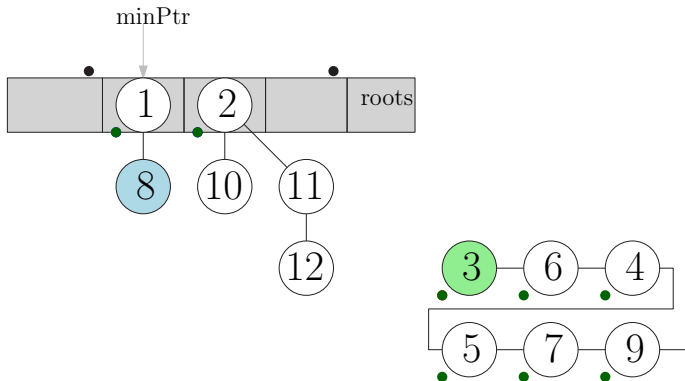
Fibonacci Heaps - Delete Min



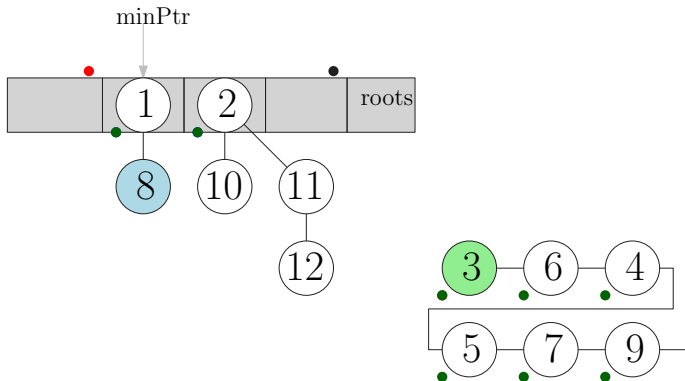
Fibonacci Heaps - Delete Min



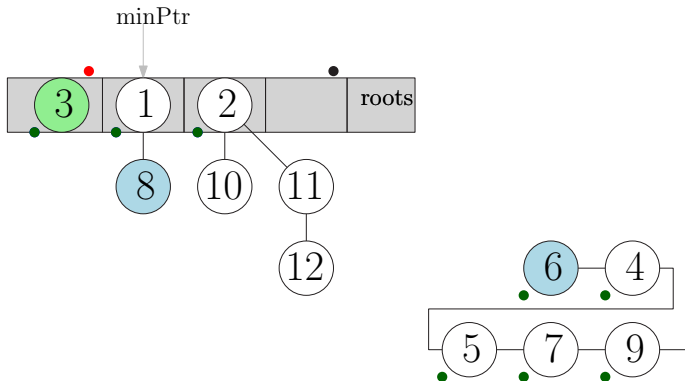
Fibonacci Heaps - Delete Min



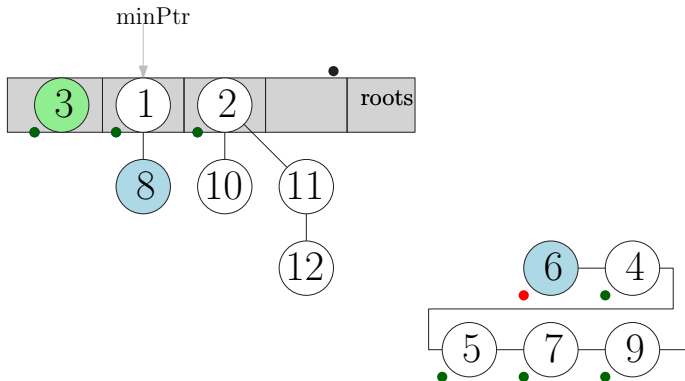
Fibonacci Heaps - Delete Min



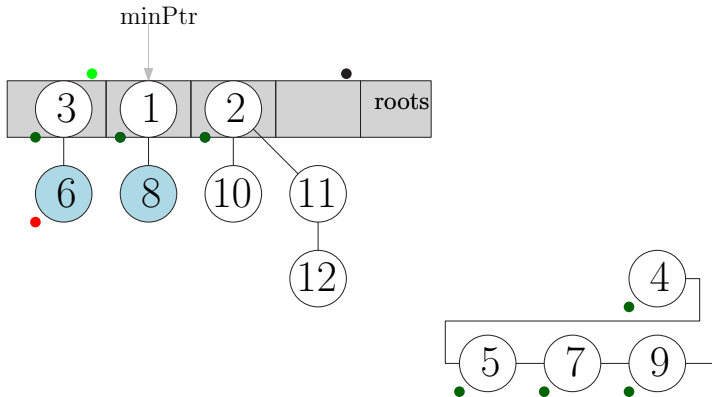
Fibonacci Heaps - Delete Min



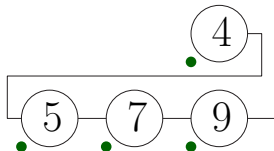
Fibonacci Heaps - Delete Min



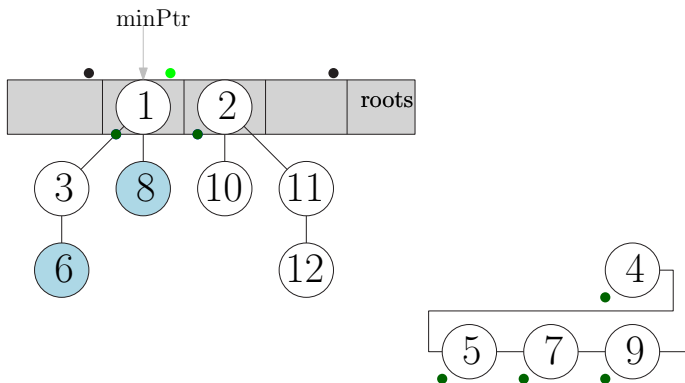
Fibonacci Heaps - Delete Min



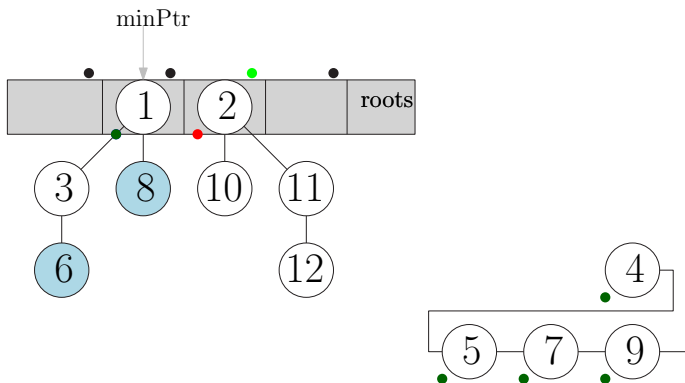
KIT
Karlsruher Institut für Technologie



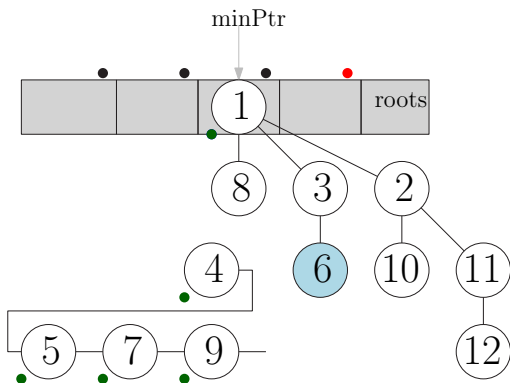
Fibonacci Heaps - Delete Min



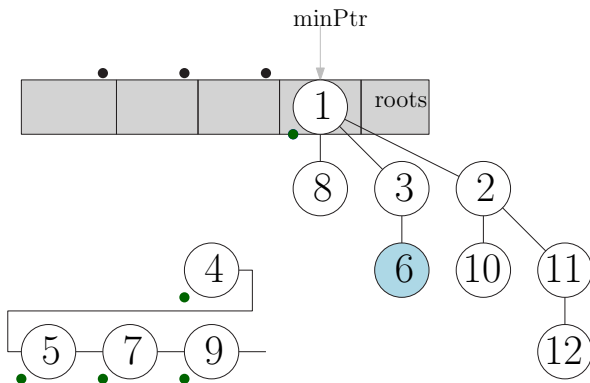
Fibonacci Heaps - Delete Min



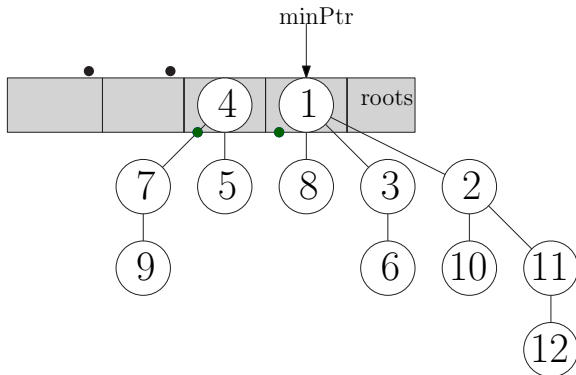
Fibonacci Heaps - Delete Min



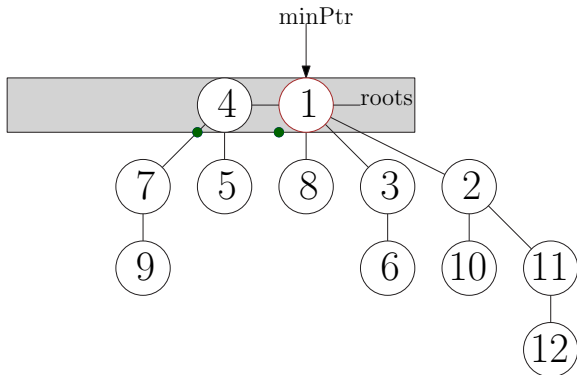
Fibonacci Heaps - Delete Min



Fibonacci Heaps - Delete Min

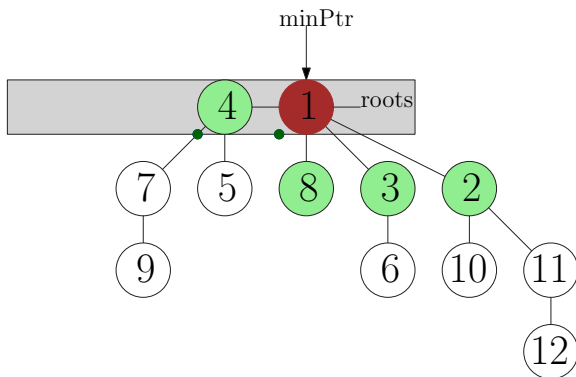


Fibonacci Heaps - Delete Min

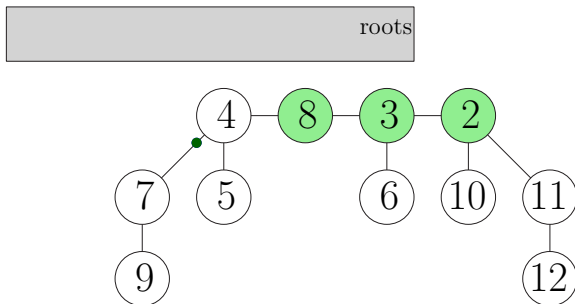


- Union nach **Rang** $\rightarrow$ logarithmischer maxRank
- Vergleich: Pairing Heaps **streng paarweise**
- Rang bezieht sich nur auf **direkte** Nachfolger
- Frage: Wie lange kann eine einzelne DeleteMin Operation laufen?

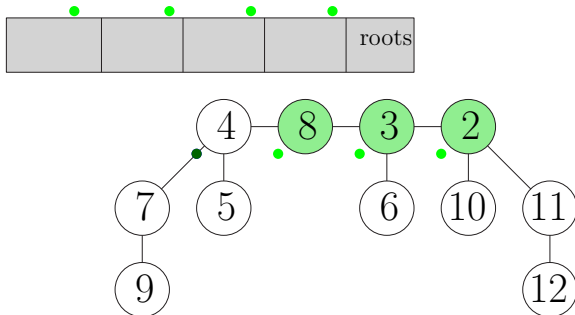
Fibonacci Heaps - Delete Min



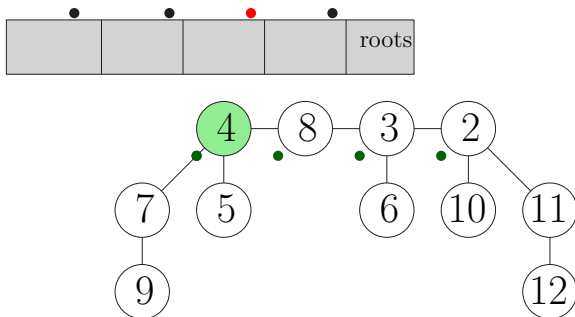
Fibonacci Heaps - Delete Min



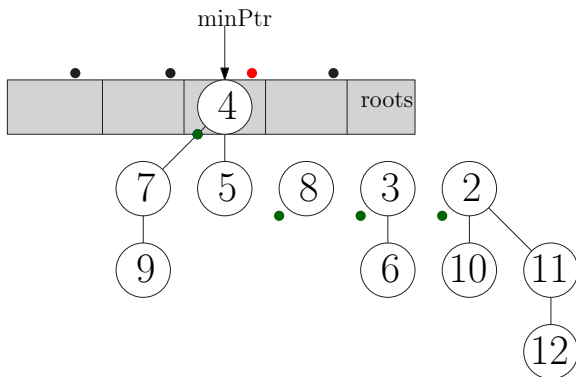
Fibonacci Heaps - Delete Min



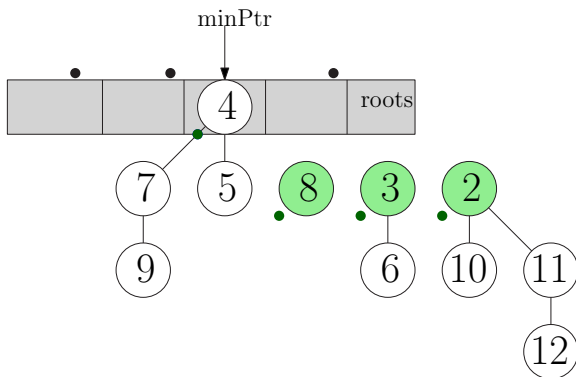
Fibonacci Heaps - Delete Min



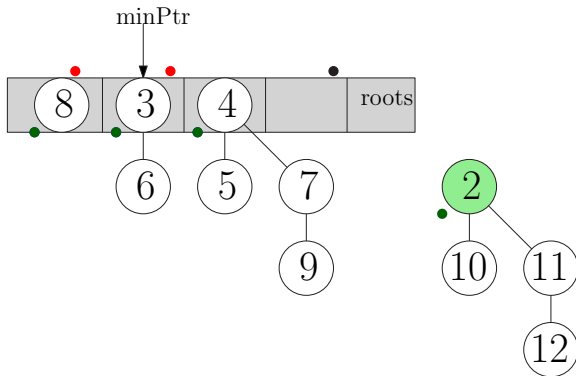
Fibonacci Heaps - Delete Min



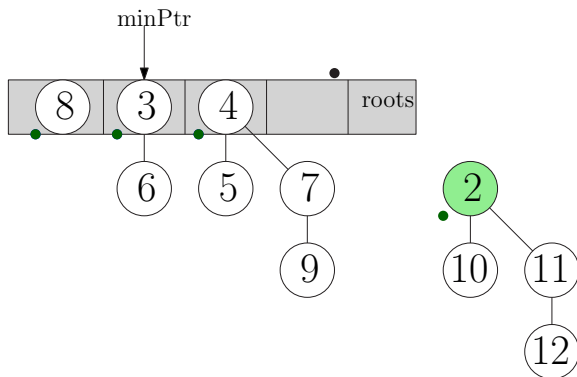
Fibonacci Heaps - Delete Min



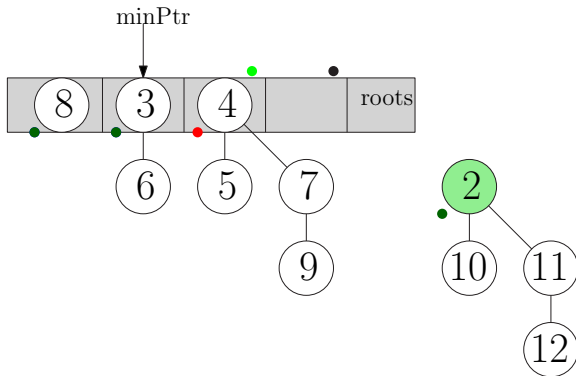
Fibonacci Heaps - Delete Min



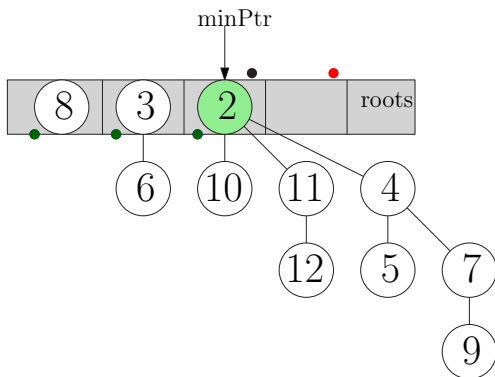
Fibonacci Heaps - Delete Min



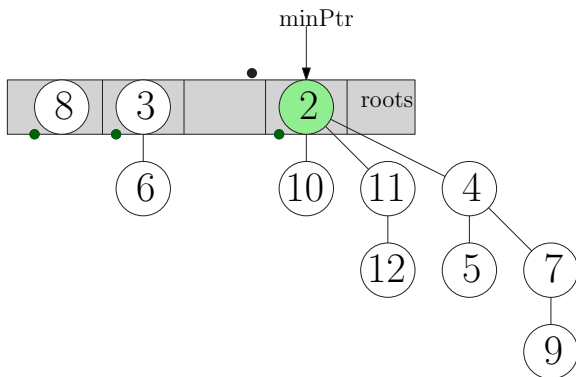
Fibonacci Heaps - Delete Min



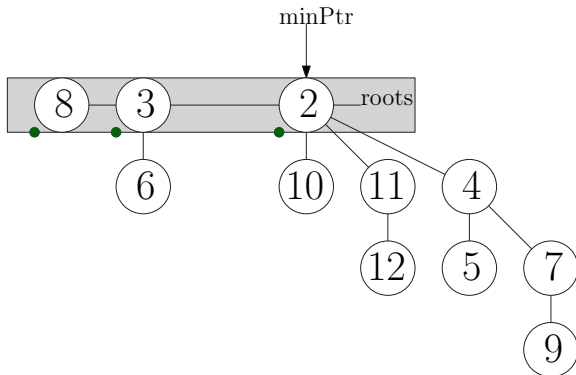
Fibonacci Heaps - Delete Min



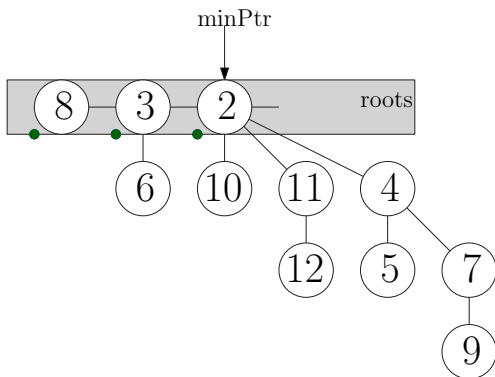
Fibonacci Heaps - Delete Min



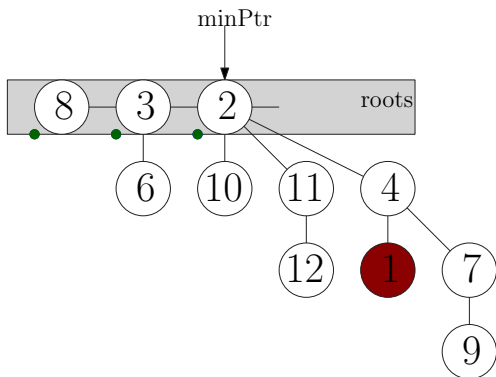
Fibonacci Heaps - Delete Min



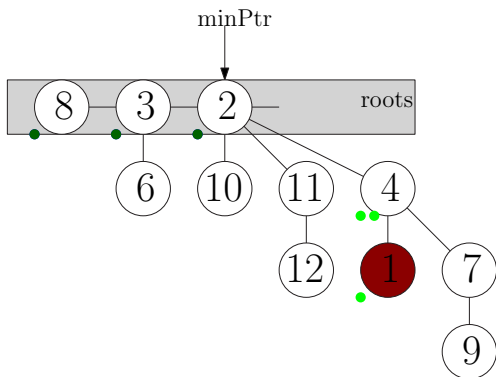
Fibonacci Heaps - Delete Min



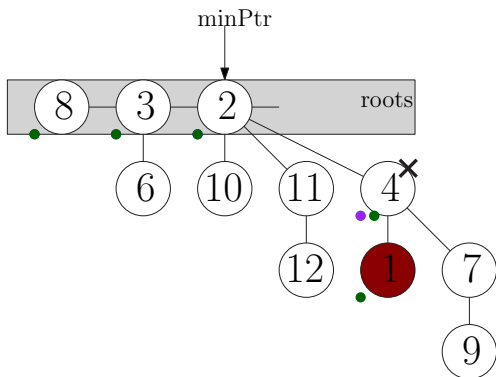
Fibonacci Heaps - Decrease Key



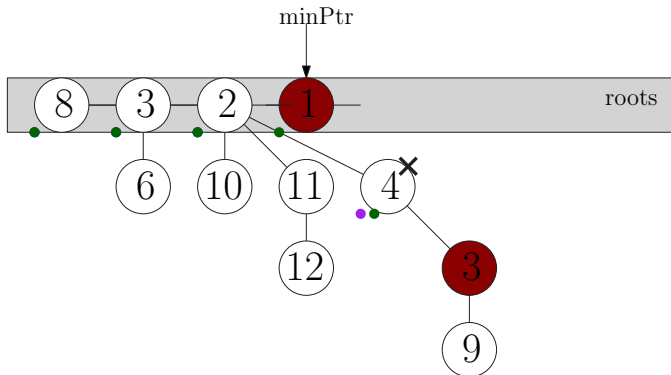
Fibonacci Heaps - Decrease Key



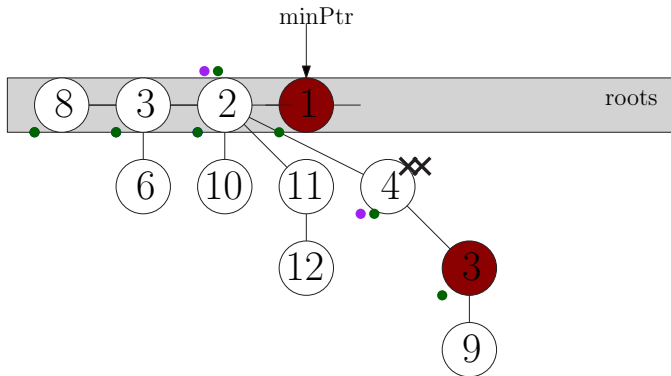
Fibonacci Heaps - Decrease Key



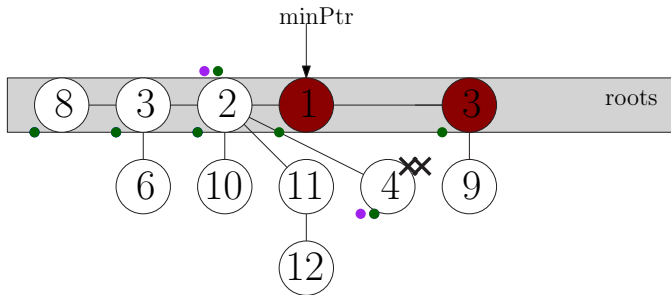
Fibonacci Heaps - Decrease Key



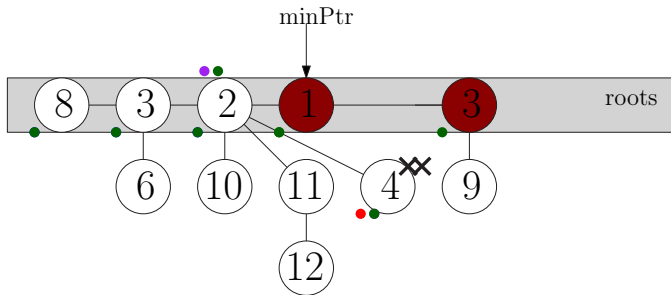
Fibonacci Heaps - Decrease Key



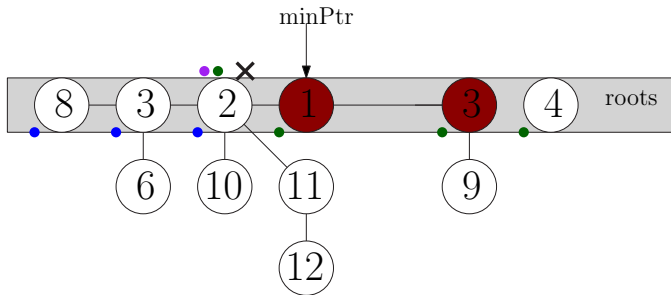
Fibonacci Heaps - Decrease Key



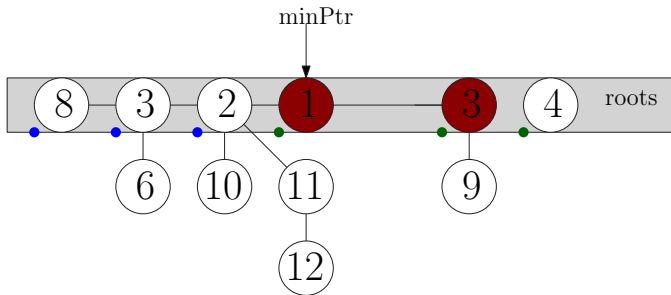
Fibonacci Heaps - Decrease Key



Fibonacci Heaps - Decrease Key



Fibonacci Heaps - Decrease Key



- Amortisierte Laufzeiten sind **optimal** (vergleichsbasiert)
- **Einzelne Operationen** Kosten aber deutlich **länger**

Ende!



Feierabend!

- ✕ Markierung - Kennzeichnet Knoten, die ein Kind verloren haben
- ✖ Doppelte Markierung - Temporäre Markierung. Muss sofort aufgelöst werden
- Neues, unbenutztes Token
- InsertTree-Token - kann genau 1 InsertTree Operation bezahlen
- Bezahltoken - wird zum Bezahlen genutzt
- Union-Token - zahlt für 1 Union Operation; zusätzlich einen InsertTree-Token erstellen
- Mark Token - Zahlt Cut und Markierung des Parents
-   Aktive Knoten
-  Knoten zum Löschen oder Cutten