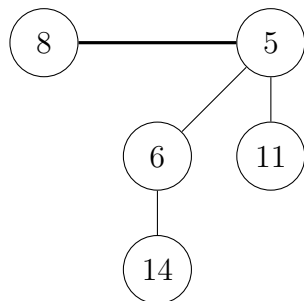

Effiziente Algorithmen und Datenstrukturen

Abgabetermin: 30. November 2010 vor der Vorlesung

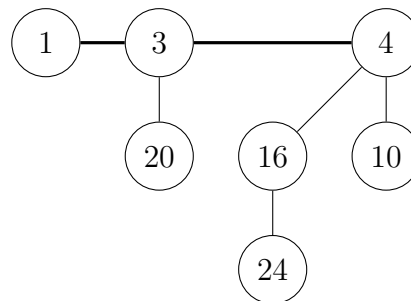
Aufgabe 1 (10 Punkte)

- a) Bei Binomial-Queues ist die Worst-Case-Laufzeit der Operationen **Insert** und **Delete** jeweils $O(\log n)$. Geben Sie für eine anfangs leere Binomial-Queue eine Folge von insgesamt n **Insert**- und **Delete**-Operationen an, so dass die Gesamt-Laufzeit $\Theta(n \log n)$ ist.
- b) Betrachten Sie die folgenden Binomial Heaps (die min-Pointer sind entsprechend gesetzt):

Heap A:



Heap B:



Führen Sie die folgenden Operationen in der angegebenen Reihenfolge aus und geben Sie den entstehenden Binomial-Heap an (jede Operation soll dabei auf das Ergebnis der vorhergehenden Operation angewendet werden):

- (a) `merge(A,B)`
- (b) `extractMin()`

Gehen Sie bei `merge` von kleinen zu großen Binomialbäumen in der Reihenfolge ihrer Priorität (von klein nach groß).

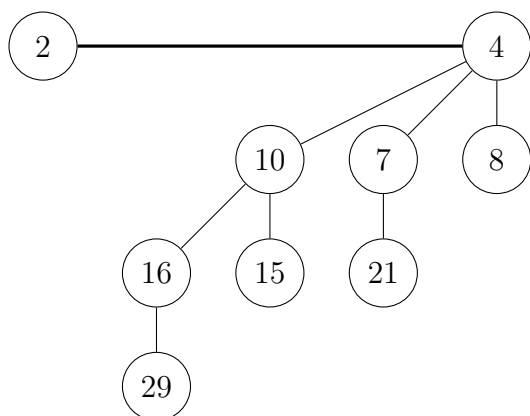
Aufgabe 2 (10 Punkte)

Die Pfadlänge P_n eines Binomialbaums B_n sei definiert als die Summe der Level (Tiefen) aller seiner Knoten. Leiten Sie eine geschlossene Formel für P_n in Abhängigkeit von n her.

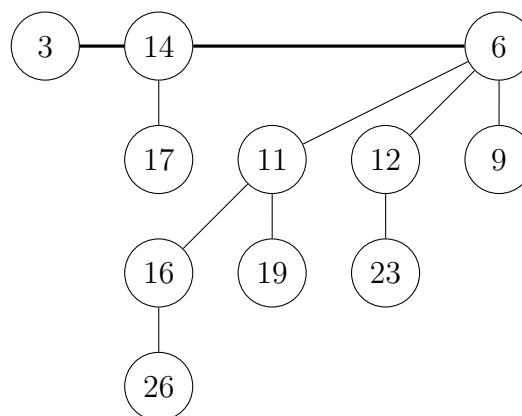
Aufgabe 3 (10 Punkte)

Betrachten Sie die folgenden Fibonacci-Heaps (Die min-Pointer sind entsprechend gesetzt):

Heap A:



Heap B:



Führen Sie die folgenden Operationen in der angegebenen Reihenfolge aus und geben Sie den jeweils entstehenden Fibonacci-Heap an (jede Operation soll dabei auf das Ergebnis der vorhergehenden Operation angewendet werden):

1. `merge(A,B)`
2. `insert(22)`
3. `extractMin()`
4. `delete(16)`
5. `decreaseKey(19,3)`

Aufgabe 4 (10 Punkte)

In den Bäumen eines Fibonacci-Heaps ist es nach Konstruktionsvorschrift erlaubt, dass von jedem inneren Knoten ein Kind entfernt werden darf, ohne dass der Baum restrukturiert werden muss. Hierdurch können stark entartete Bäume entstehen, das heißt Bäume, deren Tiefe in $\Omega(n)$ liegt, wobei n die Anzahl der Knoten im Baum sei. Beschreiben Sie eine Abfolge von Operationen, mit der, angefangen bei einem leeren Heap, ein derartiger Baum entsteht.