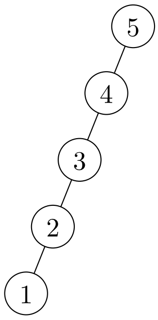
RWTH Aachen
Lehrgebiet Theoretische Informatik
Rossmanith–Reidl–Sánchez

SS 2015
Lösungsvorschlag der Klausur
?. August 2015

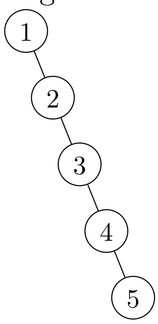
Übung zur Vorlesung Algorithmen und Datenstrukturen

Aufgabe 1 (4+3+3 Punkte)

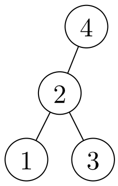
a) Es werden vier Rotationen ausgeführt. Der Baum sieht dann so aus:



b) Wieder werden vier Rotationen ausgeführt und wir erhalten diesen Baum:

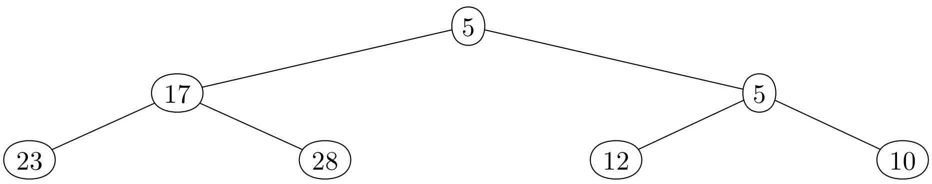


c) Nachdem das Blatt 5 entfernt wurde, wird eine Splay-Operation auf 4 ausgeführt. Mit einem Zag-zag gefolgt von einem Zag wird sie dabei in die Wurzel hochrotiert. Der resultierende Baum ist dieser:

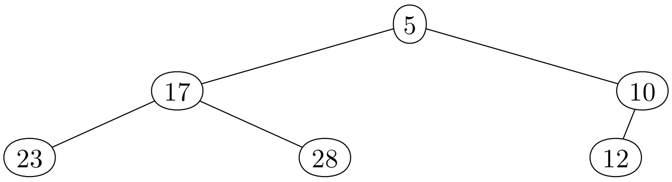


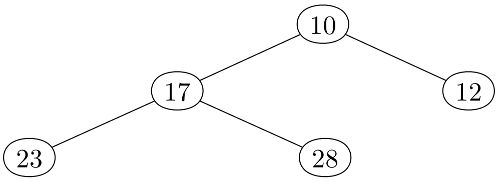
Aufgabe 2 (4+2+2+2 Punkte)

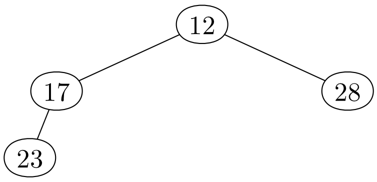
Der entstehende Heap sieht so aus:



Die weiteren Heaps nach jeweiligem Entfernen des kleinsten Elements sind:







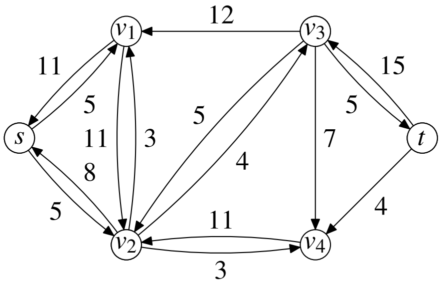
Aufgabe 3 (2+3+3+3 Punkte)

(a) Der eingezeichnete Fluß f hat offenbar den Wert 19, weil aus dem Startknoten ein Fluß von $11 + 8$ austritt.

(b) Das Residualnetzwerk ist unten abgebildet.

(c) 23, wie der minimale Schnitt über den mit 12/12, 7/7 und 4/4 beschrifteten Kanten zeigt. Wir werden das aber anders beweisen:

(d) Im Residualnetzwerk kann man einen augmentierenden Pfad der Kapazität 4 finden (s, v_2, v_3, t) – und danach keinen weiteren mehr, wie das (hier nicht abgebildete) zugehörige Residualnetzwerk zeigt.



Aufgabe 4 (2+2+2+2 Punkte)

Zunächst zu Heapsort: Die Laufzeit ist hier stets $\Theta(n \log n)$ unabhängig von b, da ja Worte der Länge b in einem Schritt verarbeitet werden.

Jetzt Radixsort: Die Laufzeit ist $\Theta(bn)$, also in den vier Fällen $\Theta(n)$, $\Theta(n \log n)$, $\Theta(n^{3/2})$ und $\Theta(n^2)$.

Man sieht, daß im ersten Fall Radixsort schneller ist und in den letzten beiden Fällen Heapsort. Im zweiten Fall sind die Laufzeiten bis auf einen konstanten Faktor gleich.

Aufgabe 5 (4+4 Punkte)

Wir können einfach eine Tiefensuche durchführen und dann sehen, ob eine Rückwärtskante existiert. Der Graph ist genau dann kreisfrei, wenn es keine Rückwärtskante gibt.

Wenn es tatsächlich eine Rückwärtskante gibt, dann gibt es auch einen Kreis, da die beiden Endknoten der Rückwärtskante durch einen aus Baumkanten bestehenden Pfad verbunden sind. Umgekehrt kann es ohne Rückwärtskanten keinen Kreis geben, denn bei einem ungerichteten Graphen gibt es nur Baum- und Rückwärtskanten. Ohne Rückwärtskanten gibt es also nur Baumkanten und der ganze Graph ist ein Wald und damit kreisfrei.