

3.3.4 Implementierung von Vorrangwarteschlangen

- ❑ Eine Minimum-Vorrangwarteschlange kann wie folgt durch eine Minimum-Binomial-Halbe implementiert werden (und analog kann eine Maximum-Vorrangwarteschlange durch eine Maximum-Binomial-Halbe implementiert werden).

Hilfsoperation

- ❑ Zusammenfassen zweier Bäume B_1 und B_2 mit Grad k zu einem Baum mit Grad $k + 1$
 - 1 Wenn die Priorität des Wurzelknotens von B_1 größer als die Priorität des Wurzelknotens von B_2 ist, mache B_1 zum Nachfolger mit dem größten Grad von B_2 :

```
B2.sibling = nil
B2.degree = B2.degree + 1
B1.parent = B2
if B2.child == nil then
    B2.child = B1.sibling = B1
else
    B1.sibling = B2.child.sibling
    B2.child = B2.child.sibling = B1
end
```

- 2 Andernfalls mache B_2 zum Nachfolger mit dem größten Grad von B_1 .

Operationen

- ❑ Vereinigen zweier Halden H_1 und H_2 zu einer neuen Halde H
 - 1 Erstelle einen leeren Zwischenspeicher für bis zu drei Bäume.
 - 2 Setze $k = 0$.
 - 3 Solange H_1 oder H_2 oder der Zwischenspeicher nicht leer sind:
 - 1 Wenn der erste (verbleibende) Baum von H_1 Grad k besitzt, entnimm ihn aus H_1 und füge ihn zum Zwischenspeicher hinzu.
 - 2 Entsprechend für H_2 .
 - 3 Wenn der Zwischenspeicher jetzt einen oder drei Bäume enthält, entnimm einen von ihnen und füge ihn am Ende von H an.
 - 4 Wenn der Zwischenspeicher jetzt noch zwei Bäume enthält, fasse sie zu einem Baum mit Grad $k + 1$ zusammen, der als „Übertrag“ für den nächsten Schritt im Zwischenspeicher verbleibt.
 - 5 Erhöhe k um 1.

Beispiel: H_1 mit $2^2 + 2^1 + 2^0 = 7$ Elementen und H_2 mit $2^3 + 2^2 + 2^1 = 14$ Elementen ergibt H mit $2^4 + 2^2 + 2^0 = 21$ Elementen

❑ Einfügen eines Objekts mit einer bestimmten Priorität

Erzeuge eine temporäre Halde mit einem einzigen Baum mit Grad 0, die das Objekt enthält, und vereinige sie mit der aktuellen Halde.

❑ Auslesen eines Objekts mit minimaler Priorität

Durchsuche die Liste der Wurzelknoten nach einem Objekt mit minimaler Priorität und liefere es zurück.

❑ Entnehmen eines Objekts mit minimaler Priorität

- 1 Durchsuche die Liste der Wurzelknoten nach einem Objekt mit minimaler Priorität und entferne diesen Knoten aus der Liste.
- 2 Wenn dieser Knoten Nachfolger besitzt:
Vereinige die Liste seiner Nachfolger (beginnend mit dem Nachfolger mit dem kleinsten Grad, der über `child` → `sibling` direkt zugreifbar ist) mit der verbleibenden Halde.

❑ Nachträgliches Ändern der Priorität eines Objekts

- 1 Setze die Priorität des Objekts auf die neue Priorität.
- 2 Wenn die neue Priorität kleiner oder gleich der alten ist:
Solange die Priorität des Objekts kleiner als die seines Vorgängers ist:
Vertausche das Objekt mit seinem Vorgänger
durch „Über-Kreuz-Verbinden“ der beiden Objekte mit den zugehörigen Knoten.
- 3 Andernfalls, sofern sich das Objekt nicht in einem Blattknoten befindet:
Entferne das Objekt und füge es mit der neuen Priorität wieder ein.

❑ Entfernen eines Objekts

- 1 Entnimm ein Objekt mit minimaler Priorität.
- 2 Wenn das entnommene Objekt nicht das zu entfernende Objekt ist:
 - 1 Ändere die Priorität des zu entfernenden Objekts auf die Priorität des entnommenen Objekts.
 - 2 Verbinde das entnommene Objekt mit dem Knoten des zu entfernenden Objekts.
 - 3 Setze die Priorität des zu entfernenden Objekts wieder auf ihren ursprünglichen Wert.

Laufzeit der Operationen

- ❑ Wenn die Halde N Objekte enthält, besitzen alle `sibling`-Listen höchstens Länge $O(\log_2 N)$ und alle Bäume höchstens Tiefe $O(\log_2 N)$ (vgl. § 3.3.2).
- ❑ Daher besitzen alle Operationen höchstens Laufzeit $O(\log_2 N)$.

3.3.5 Entwurfsüberlegungen für die Datenstruktur

- ❑ Beim Vereinigen zweier Halden müssen ihre Bäume nach aufsteigendem Grad durchlaufen werden.
Deshalb sind die Wurzelknoten in dieser Reihenfolge verkettet.
- ❑ Beim Entnehmen eines Wurzelknotens muss die Liste seiner Nachfolger mit der verbleibenden Wurzelliste vereinigt werden.
Deshalb ist die Liste der Nachfolger eines Knotens ebenfalls nach aufsteigendem Grad sortiert verkettet.
Dies weicht von der Darstellung in Cormen et al. (vgl. § 1.3) ab, wo die Liste nach absteigendem Grad verkettet ist und deshalb vor einer Vereinigung erst einmal invertiert werden muss.
- ❑ Beim Zusammenfassen zweier Bäume mit dem gleichen Grad wird einer der Bäume zum Nachfolger mit dem größten Grad des anderen Baums, d. h. er muss am Ende der Nachfolgerliste angehängt werden.
Damit dies möglichst einfach und effizient geht, verweist `child` auf den Nachfolger mit dem größten Grad.
- ❑ Damit aber auch der Nachfolger mit dem kleinsten Grad direkt zugreifbar ist, ist die Liste der Nachfolger zirkulär verkettet.
- ❑ Für die Wurzelliste genügt jedoch eine einfache Verkettung.