

1 Datenstrukturen

- 1.1 Abstrakte Datentypen
- 1.2 Lineare Strukturen
- 1.3 Bäume
- 1.4 **Prioritätsschlangen**
- 1.5 Graphen





Prioritätsschlangen

- Warteschlangen mit „Vordrängeln“
- Einfügen am Ende der Schlange
- Jedes Element hat eine Priorität
- Deq() liefert immer das Element mit der höchsten Priorität



Prioritätsschlangen



- Elemente: $X \in W' = W \times R$
Wertebereich: $L = \{ \} \cup W' \cup W'^2 \cup W'^3 \cup \dots$
- Create : $\rightarrow L$
- Enq* : $W' \times L \rightarrow L$
- Deq : $L \rightarrow L$
- Get : $L \rightarrow W'$
- Empty : $L \rightarrow \text{Bool}$


Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer


W' sind die Prioritäten
(fehlt hier Enq (ohne Stern)?)

Prioritätsschlangen

• Empty(Create())	= true
Empty(Enq(x,z))	= false
• Deq(Enq(x,Create()))	= Create()
Deq(Enq(x,z))	= Enq(x,Deq(z)) if $z \neq \{ \}$
• Get(Enq(x,Create()))	= x
Get(Enq(x,z))	= Get(z) if $z \neq \{ \}$



Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer


Prioritätsschlangen


- Enq() : Standardoperator
- Enq*() :
Füge Element x bzgl. seiner Priorität prio[x] ein:

- $\text{Enq}^*(x, \text{Enq}(y, z)) = \text{Enq}(y, \text{Enq}^*(x, z))$
... if $\text{prio}[x] > \text{prio}[y]$

- $\text{Enq}^*(x, \text{Enq}(y, z)) = \text{Enq}(x, \text{Enq}(y, z))$
... if $\text{prio}[x] \leq \text{prio}[y]$




Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer




Enq* drängelt bis zur richtigen Position vor

Prioritätsschlangen

- Aufwand steigt proportional zur Länge der Liste (= Zahl der Axiom-Anwendungen).
- Liste wird vollständig sortiert, obwohl eigentlich nur das Element mit maximaler Priorität gefunden werden muss.
- Ein- und Ausfüge-Reihenfolge nicht vorhersagbar.



Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer



Effiziente Implementierung

- Warteschlangen: einfache Implementierung (front und back Pointer)
- Binär-Bäume: schneller Elementzugriff
- Array-Implementierung vollständiger Bäume kombiniert beide Vorteile
- Aber: Bei Suchbäumen ist das maximale Element unten im Baum ...



Suche nach dem maximalen Element muss also die gesamte Höhe des Baumes durchlaufen

Heap-Bedingung

- Neue Sortierung der Knoten im Baum. Für alle Knoten $T = \text{Node}(L,x,R)$ muss gelten
 - $\text{max_prio}(L) \leq \text{prio}[x]$
 - $\text{max_prio}(R) \leq \text{prio}[x]$
- Element mit maximaler Priorität ist im Wurzelknoten gespeichert.
- Sortierung der Elemente nur entlang der Pfade im Binärbaum.



Im Heap sind im Gegensatz zum Suchbaum nur die Pfade sortiert (nicht alle Elemente)

Effiziente Implementierung

- Array-Implementierung
 - Front-Pointer bleibt fest
 - Element mit maximaler Priorität in $S[1]$
 - Back-Pointer variabel
 - Einfügen in $S[\text{back}]$

front back

↓ ↓

x_1	x_2	x_n					
-------	-------	-----	-----	-------	--	--	--	--	--

Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

Hybrid aus Warteschlange und Baum

Effiziente Implementierung

- $\text{Enq}(x)$

```

if back = length+1 then
  Q-FULL-ERROR
else
  i ← back
  back ← back+1
  S[i] ← x
  while i ≠ 1 and S[i] > S[i / 2] do
    swap(S[i], S[i / 2])
    i ← i / 2

```

Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

Eingefügtes Element muss sich höchstens an h Elementen vorbei tauschen (bei Heap von Höhe h)

Effiziente Implementierung

- Deq()
 - if back = 1 then
 - Q-EMPTY-ERROR
 - else
 - $i \leftarrow 1$; back \leftarrow back-1; S[1] \leftarrow S[back]
 - while $i \leq (\text{back}-1) / 2$ do
 - $j \leftarrow 2 \cdot i$
 - if $j < \text{back}-1$ and $S[j] < S[j+1]$ then
 - $j \leftarrow j+1$
 - if $S[i] < S[j]$ then
 - swap(S[i], S[j])
 - $i \leftarrow j$
 - else
 - $i \leftarrow \text{back}$

11
Datenstrukturen und Algorithmen
FRIEDRICH-SCHILLER
UNIVERSITÄT

Erstes Element löschen, letztes Element an erste Stelle, dann rekursiv mit größtem Kind vertauschen.

Beispiel

back

↓

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

12
Datenstrukturen und Algorithmen
FRIEDRICH-SCHILLER
UNIVERSITÄT

Beispiel



Enq(5)

back

↓

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

1 2 3 4 5 6 7 8 9 10

13  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer  **FWTH AACHEN UNIVERSITY**



Beispiel

back

↓

5									
---	--	--	--	--	--	--	--	--	--

1 2 3 4 5 6 7 8 9 10

14  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer  **FWTH AACHEN UNIVERSITY**

Beispiel

Enq(3)

back

5									
1	2	3	4	5	6	7	8	9	10

15 **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

Beispiel

back

5	3								
1	2	3	4	5	6	7	8	9	10

16 **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

Beispiel

Enq(2)

back

5	3								
1	2	3	4	5	6	7	8	9	10

17

Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Beispiel

back

5	3	2							
1	2	3	4	5	6	7	8	9	10

18

Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG



Beispiel

Enq(8)

back

5	3	2							
---	---	---	--	--	--	--	--	--	--

1 2 3 4 5 6 7 8 9 10



19  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer  **FWTHAACHEN UNIVERSITY**

Beispiel

back

5	3	2	8						
---	---	---	---	--	--	--	--	--	--



1 2 3 4 5 6 7 8 9 10

20  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer  **FWTHAACHEN UNIVERSITY**

Beispiel

The diagram shows an array of 10 cells. The first four cells contain the values 5, 3, 2, and 8. The remaining six cells are empty. An arrow labeled "back" points to the fifth cell. A bracket below the array indicates the range from index 1 to 5.



1	2	3	4	5	6	7	8	9	10
5	3	2	8						

21  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer 

Beispiel

The diagram shows an array of 10 cells. The first four cells contain the values 5, 8, 2, and 3. The remaining six cells are empty. An arrow labeled "back" points to the fifth cell. A bracket below the array indicates the range from index 1 to 5.

1	2	3	4	5	6	7	8	9	10
5	8	2	3						

22  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer 

Beispiel

back

5	8	2	3						
1	2	3	4	5	6	7	8	9	10

↑

23 **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

Beispiel

back

8	5	2	3						
1	2	3	4	5	6	7	8	9	10

↑

24 **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer



Beispiel

Enq(7)
back

↓

1	2	3	4	5	6	7	8	9	10
8	5	2	3						

1 2 3 4 5 6 7 8 9 10

25  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer  **FRIEDRICH-ALEXANDER
UNIVERSITÄT**



Beispiel

back

↓

1	2	3	4	5	6	7	8	9	10
8	5	2	3	7					

1 2 3 4 5 6 7 8 9 10

26  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer  **FRIEDRICH-ALEXANDER
UNIVERSITÄT**

Beispiel

back

8	5	2	3	7					
1	2	3	4	5	6	7	8	9	10

27

Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Beispiel

back

8	7	2	3	5					
1	2	3	4	5	6	7	8	9	10

28

Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Beispiel

back

8	7	2	3	5					
1	2	3	4	5	6	7	8	9	10

↑



29  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer  FRIEDRICH-ALEXANDER
UNIVERSITY

Beispiel

Enq(6)

back

8	7	2	3	5					
1	2	3	4	5	6	7	8	9	10

30  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer  FRIEDRICH-ALEXANDER
UNIVERSITY

Beispiel

Enq(6)

back

8	7	2	3	5	6				
1	2	3	4	5	6	7	8	9	10

31

Datenstrukturen und Algorithmen
 Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

Beispiel

Enq(6)

back

8	7	6	3	5	2				
1	2	3	4	5	6	7	8	9	10

32

Datenstrukturen und Algorithmen
 Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer



Beispiel

Enq(6)

back

8	7	6	3	5	2				
1	2	3	4	5	6	7	8	9	10

↑



33  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer 

Beispiel

Deq()

back

8	7	6	3	5	2				
1	2	3	4	5	6	7	8	9	10

34  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer 



Beispiel

back

↓

1	2	3	4	5	6	7	8	9	10
	7	6	3	5	2				

1 2 3 4 5 6 7 8 9 10

35  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer 



Beispiel

back

↓

1	2	3	4	5	6	7	8	9	10
	7	6	3	5	2				

1 2 3 4 5 6 7 8 9 10

36  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer 

Beispiel

back

2	7	6		5					
1	2	3		5	6	7	8	9	10

37

Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

FWTHAACHEN
UNIVERSITY

Beispiel

back

2	7	6	3	5					
1	2	3	4	5	6	7	8	9	10

38

Datenstrukturen und Algorithmen
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

FWTHAACHEN
UNIVERSITY

Beispiel

back

2	7	6	3	5					
1	2	3	4	5	6	7	8	9	10

↑ ↑

39 **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

Beispiel

back

7	2	6	3	5					
1	2	3	4	5	6	7	8	9	10



↑ ↑

40 **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer

Beispiel

back



The diagram shows a horizontal array of 10 cells, indexed 1 to 10. The first five cells contain the values 7, 2, 6, 3, and 5. The remaining five cells are empty. An arrow labeled "back" points to the cell at index 6. Below the array, a bracket spans from index 1 to index 5, with two upward-pointing arrows at indices 4 and 5.

41  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer  FRIEDRICH-ALEXANDER
UNIVERSITÄT

Beispiel

back

The diagram shows a horizontal array of 10 cells, indexed 1 to 10. The first five cells contain the values 7, 5, 6, 3, and 2. The remaining five cells are empty. An arrow labeled "back" points to the cell at index 6. Below the array, a bracket spans from index 1 to index 5, with two upward-pointing arrows at indices 4 and 5.

42  **Datenstrukturen und Algorithmen**
Prof. Dr. Leif Kobbelt, Thomas Stroder, Fabian Emmes, Sven Middelberg, Michael Kremer  FRIEDRICH-ALEXANDER
UNIVERSITÄT

Beispiel

