

Prioritätswarteschlange

Eine Prioritätswarteschlange
zur Verwaltung
der günstigsten Alternativen

Prioritätswarteschlange

- Die Prioritätswarteschlange ist eine der Datenstrukturen, die zur Verwaltung von Alternativen bei Suchverfahren verwendet wird:
 - Stapel (stack) bei der Tiefensuche
last in first out
 - Warteschlange (queue) bei der Breitensuche
first in first out

Beide ordnen die Elemente nach ihrem Auftreten ein.

Prioritätswarteschlange

- Die Prioritätswarteschlange ist eine der Datenstrukturen, die zur Verwaltung von Alternativen bei Suchverfahren verwendet wird:
 - Stapel ...
 - Warteschlange ...
 - Prioritätswarteschlange (priority queue)
zB bei Dijkstra

Elemente werden geordnet nach ihrer Bewertung eingefügt.

Prioritätswarteschlange

Allgemein

- Prinzipiell handelt es sich um ein objekt-orientiertes Vorgehen:
Daten mit zugehörigen Zugriffsfunktionen
- Hier wird es funktional realisiert, d.h. das Programm *Prioritätswarteschlange.scm*
 - stellt die zur Verwaltung einer Prioritätswarteschlange notwendigen Funktionen bereit;
 - es benötigt ein Vergleichsprädikat *vor?* , das vom einbindenden Programm kommt.

Prioritätswarteschlange

- fuege-eines-ein
 - fügt ein Element in die Prio-Schlange ein
- Kopf

(define
(fuege-eines-ein neu vor? priows)
'undefiniert)

Prioritätswarteschlange

- fuege-eines-ein
 - fügt ein Element in die Prio-Schlange ein
- Elementarfall Schlange leer

```
(define
  (fuege-eines-ein neu vor? priows)
  (cond
    ((null? priows) (list neu))
    (else
     'undefiniert)
  ))
```

Prioritätswarteschlange

- fuege-eines-ein
 - fügt ein Element in die Prio-Schlange ein
- Elementarfall Element kommt hier hin

```
(define
  (fuege-eines-ein neu vor? priows)
  (cond
    ((null? priows) (list neu))
    ((vor? neu (first priows))
     (cons neu priows))
    (else
     'undefiniert)
  ))
```

Prioritätswarteschlange

- Sonst Position weiter suchen, beim Backtracking Prio-WS wieder aufbauen

```
(define
  (fuege-eines-ein neu vor? priows)
  (cond
    ((null? priows) (list neu))
    ((vor? neu (first priows))
     (cons neu priows))
    (else
     (cons
      (first priows)
      (fuege-eines-ein
       neu vor? (rest priows))))))
  ))
```

Prioritätswarteschlange

- Fuege-alle-ein
 - fügt alle übergebenen Elemente in die Prio-Schlange ein, also Rekursion über die Liste der übergebenen Elemente

```
(define
  (fuege-alle-ein alle vor? priows)
  (cond
    ((null? alle) priows)
    (else
     (fuege-alle-ein
      (rest alle)
      vor?
      (fuege-eines-ein
       (first alle) vor? priows))))))
```

Prioritätswarteschlange

Konkret bei Dijkstra für die Datenstruktur
Kantenliste:

- Ordnung nach den Kantenbewertungen wird allein durch das Übergeben
 - des Vergleichsprädikats *vor?*
 - das Übergeben der einzufügenden Kanten
 - und das Halten der Prioritätswarteschlange in Aufruf und Rückgabe

realisiert.