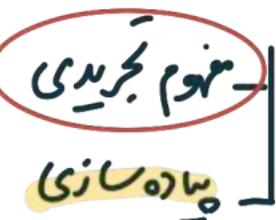


# Priority Queue

key field : Priority  
اولویت

Abstract definition

صف اولویت بندی سرہ



که صفت نه ترتیب خروج با خصیر آن بعض مقدار فیلڈ طبیعه عنصر تعین میشود

اولویت خروج با خصیر است که مقدار فیلڈ طبیعه آن درین سایر خصیر کمترین باشد



صف اولویت‌بندی در رده ساختارهای داده‌ای پوشا

## Dynamic Data Structure

$$\left\{ \begin{array}{l} S.\underline{\text{Search}}(k) \rightsquigarrow x.\text{key} == k \\ S.\underline{\text{Insert}}(x) \rightsquigarrow S = S \cup \{x\} \\ S.\underline{\text{Delete}}(x) \rightsquigarrow S = S - \{x\} \end{array} \right.$$

(صف اولویت‌بندی سرعت‌گذاری مسجحه، ارج و م Raf بخصوصی به صورت زیر تعریف و شونده است)

S.Search(k)      نها حدود      PQ.getMin()

اضافه کردن

عصر مینیم

S.Insert(x)      اضافه کردن عصر      PQ.insert(x)

(خواه به صفت

S.Delete(x)      حذف نها حدود      PQ.extractMin()

حذف عصر مینیم

لیستهای صفت و لیستهای سبدی

(لیستهای سبدی) لیستهای فربنوسن، لیستهای اریتمتیکی

✓ (MinHeap) Heap اسکاریو

Fibonacci Heap . . .

Data structure \ function	getMin	insert	extractMin
sorted Array	$O(1)$	$O(n)$	$O(n)$
sorted List	$O(1)$	$O(n)^*$	$O(1)$
Min Heap	$O(1)$	$O(\lg n)$	$O(\lg n)$

\* Cost for searching the proper position to insert

بُحُور مَابِه می توان صَف او لَوْسَ بَنْدِی تَعْرِيف کرد / عَصْر با بَزْلَرَین

مَعَادِر فَيْلَد طَبِيد او لَوْسَ خَرْوَج دَارَد  $\leftarrow$  MaxHeap

MAX-HEAP-MAXIMUM( $A$ )

1 **if**  $A.\text{heap-size} < 1$   
2      **error** "heap underflow"  
3 **return**  $A[1]$

جَبَّاجِي بَزْلَرَين عَصْر  $\rightarrow$

MAX-HEAP-EXTRACT-MAX( $A$ )

1  $max = \text{MAX-HEAP-MAXIMUM}(A)$   
2  $A[1] = A[A.\text{heap-size}]$   
3  $A.\text{heap-size} = A.\text{heap-size} - 1$   
4 MAX-HEAPIFY( $A, 1$ )  
5 **return**  $max$

اسْخَراج  
(عَصْر و حَذْف)  
بَزْلَرَين عَصْر

MAX-HEAP-INSERT( $A, x, n$ )

1 **if**  $A.\text{heap-size} == n$   
2      **error** "heap overflow"  
3  $A.\text{heap-size} = A.\text{heap-size} + 1$   
4  $k = x.\text{key}$   
5  $x.\text{key} = -\infty$   
6  $A[A.\text{heap-size}] = x$   
7 map  $x$  to index  $\text{heap-size}$  in the array  
8 **MAX-HEAP-INCREASE-KEY( $A, x, k$ )**

رجَع بَلْ عَصْر جَدِيد  
(رَصَف او لَوْسَ بَنْدِي)

افزالن مقدار فیلڈ طبیعی (اولویت) سے عضور (رکھنے اولویت بندی  
و خنکام سازی) ساختہ دہائی (رکھنے کے لیے نیاز)

### MAX-HEAP-INCREASE-KEY( $A, x, k$ )

- 1 **if**  $k < x.key$
- 2       **error** “new key is smaller than current key”
- 3        $x.key = k$
- 4       find the index  $i$  in array  $A$  where object  $x$  occurs
- 5       **while**  $i > 1$  and  $A[\text{PARENT}(i)].key < A[i].key$
- 6           exchange  $A[i]$  with  $A[\text{PARENT}(i)]$ , updating the priority queue objects to array indices
- 7            $i = \text{PARENT}(i)$

نامه ۱. در توابع Max-Heap-Increasekey و Max-Heap-Insert

فرضیه سود حرصنگر آرایه  $A$  نهادارنده یک سُنّ از مُنْتَج نوع داروهای پیچیده است؛ لعنی آن خود میتواند شامل فیلترهای متعارف باشد. فیلتر کلید  $x.key$  نیز از آنهاست.

نامه ۲. هر سُنّ از نوع داروهای پیچیده که در  $A$  ذخیره میسود باشد از موقعیت ذخیره سازی خودش در  $A$  مطلع باشد. می راه برای تحقق این حذف، تعریف و بروزرسانی یک نمایش دو طرفه است به فرمی که  $\rightarrow^i$  و  $map(x) \rightarrow^i x$ ،  $map(i) \rightarrow^i x$  را بخواهد.

برای این حذف نیز طافی است یک فیلتر به صورت  $x.ptr$  در نظر گرفته باشیم (ذخیره در  $A$  را نگهداشی کند).

## MAX-HEAP-INSERT( $A, x, n$ )

- 1 **if**  $A.\text{heap-size} == n$   
**error** "heap overflow"
- 2  $A.\text{heap-size} = A.\text{heap-size} + 1$
- 3  $k = x.\text{key}$
- 4  $x.\text{key} = -\infty$
- 5  $A[A.\text{heap-size}] = x$   $x.\text{ptr} = A.\text{heap-size}$
- 6 ~~map  $x$  to index  $\text{heap-size}$  in the array ↗~~
- 7 ~~map  $x$  to index  $\text{heap-size}$  in the array ↗~~
- 8 MAX-HEAP-INCREASE-KEY( $A, x, k$ )

## MAX-HEAP-INCREASE-KEY( $A, x, k$ )

- 1 **if**  $k < x.\text{key}$   
**error** "new key is smaller than current key"
- 2  $x.\text{key} = k$   $i = x.\text{ptr}$  ↗
- 3 ~~find the index  $i$  in array  $A$  where object  $x$  occurs~~
- 4 ~~find the index  $i$  in array  $A$  where object  $x$  occurs~~
- 5 **while**  $i > 1$  and  $A[\text{PARENT}(i)].\text{key} < A[i].\text{key}$   
exchange  $A[i]$  with  $A[\text{PARENT}(i)]$ , ~~updating the priority queue objects to array indices ↗~~
- 6 ~~update the priority queue objects to array indices ↗~~
- 7  $i = \text{PARENT}(i)$   $A[i].\text{ptr} = i$   
 $A[\text{Parent}(i)].\text{ptr} = \text{Parent}(i)$

۳۰. اگر نوع داده‌ای عنصر صفت اولویت، سی نوع داده‌ای ساده‌تر نام  
دھان نفس فیلڈ کیمپ را بازی فیلن، توابع Max-Heap-Insert و Max-Heap-Increasekey  
بھورتے زیر، سادھی موند:

MAX-HEAP-INSERT( $A, x, n$ )

- 1 **if**  $A.\text{heap-size} == n$
- 2     **error** "heap overflow"
- 3      $A.\text{heap-size} = A.\text{heap-size} + 1$
- 4      ~~$k = x.\text{key}$~~
- 5      ~~$x.\text{key} = -\infty$~~
- 6      $A[A.\text{heap-size}] = x$
- 7     ~~map  $x$  to index  $\text{heap-size}$  in the array~~
- 8     MAX-HEAP-INCREASE-KEY( $A, x, k$ )  
 $, i = A.\text{heap-size}$

MAX-HEAP-INCREASE-KEY( $A, x, k$ )

- 1 ~~**if**  $k < x.\text{key}$~~
- 2     ~~**error** "new key is smaller than current key"~~
- 3      ~~$x.\text{key} = k$~~
- 4     ~~find the index  $i$  in array  $A$  where object  $x$  occurs~~
- 5     **while**  $i > 1$  and  $A[\text{PARENT}(i)].\text{key} < A[i].\text{key}$
- 6         exchange  $A[i]$  with  $A[\text{PARENT}(i)]$ , ~~updating the priority queue objects to array indices~~
- 7          $i = \text{PARENT}(i)$