ساختمان داده ها

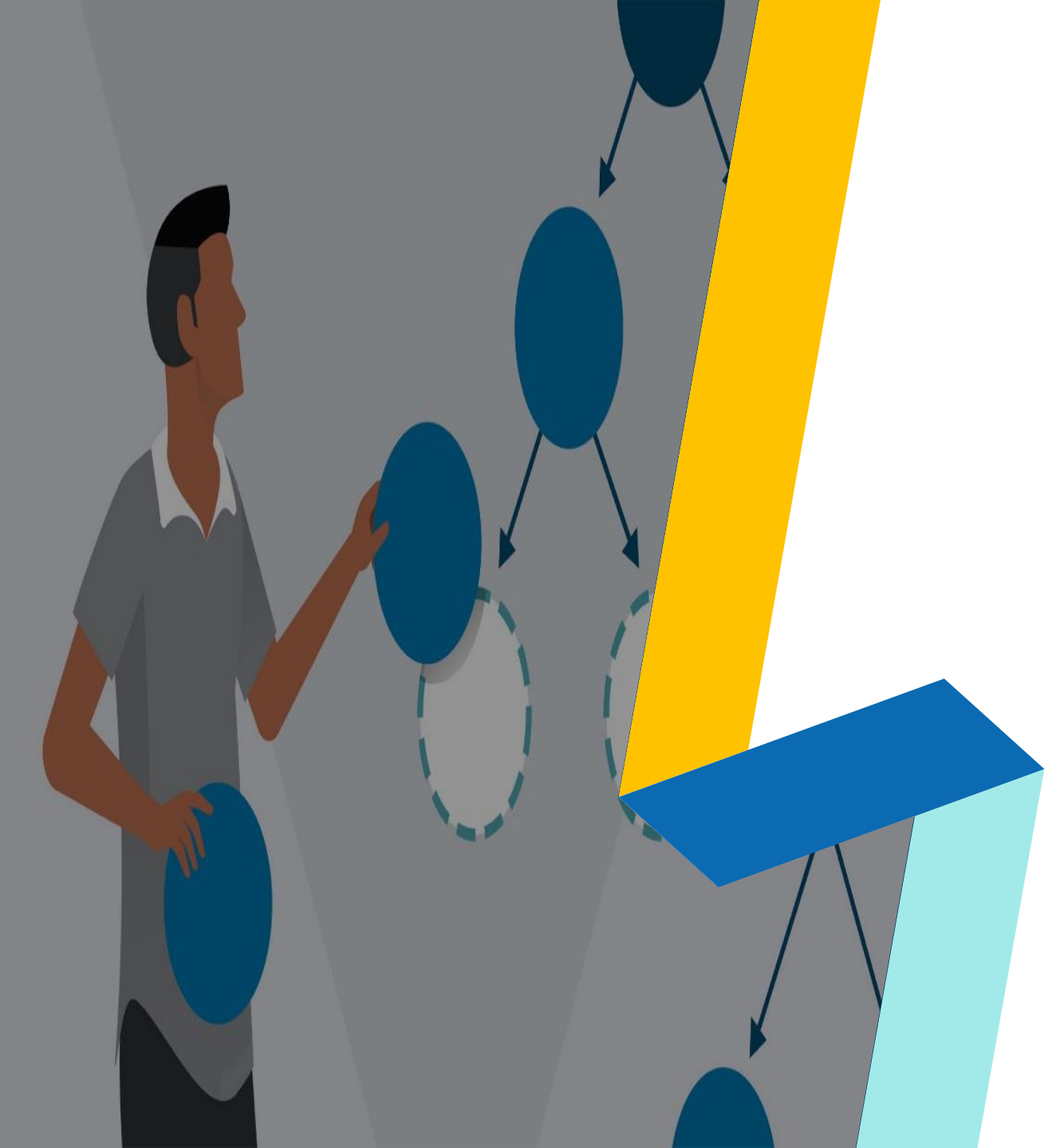
دانشگاه صنعتی نوشیروانی بابل

دکتر حسام عمران پور

طراحان اسلاید:
زهرا ریحانیان و دانیال علیزاده

حل تمرین :
علی باقری

لینک کانال تلگرام اطلاع رسانی و حل تمرین:
t.me/ds_nit_4011



فصل ۹

صف اولویت

صف اولویت

نوع خاصی از صف می باشد که خروج عناصر از آن، به ترتیب ورود عناصر به صف ارتباطی ندارد.

راه های پیاده سازی صف اولویت :

۱) در هنگام خارج کردن، با یک جستجو O(n)، عنصر دارای بیشترین مقدار را پیدا کرده و آن را از صف خارج نمود. در این صورت عملیات EnQueue دارای پیچیدگی زمانی O(1) خواهد بود.

۲) نگهداری عناصر به صورت مرتب؛ که سبب کاهش پیچیدگی زمانی عمل DeQueue و افزایش پیچیدگی EnQueue می گردد.

هرم

یکی از رایج ترین روش های پیاده سازی صف اولویت، استفاده از ساختار هرم است.

یک درخت دودویی است که در آن مقدار موجود در هر گره، بزرگتر (کوچکتر) از مقدار موجود در فرزندان آن است.

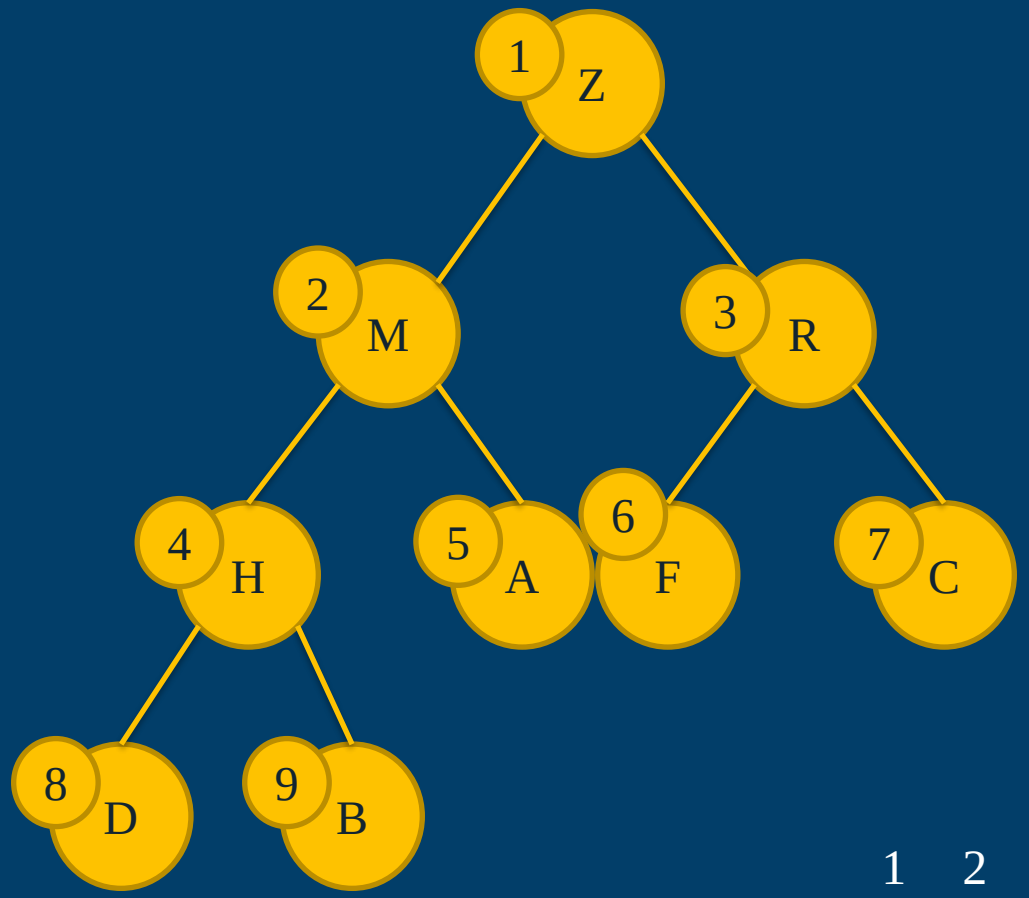
دو گروه هرم وجود دارد :

۱) هرم بیشینه (هرم بزرگی) : مقدار موجود در هر گره، بزرگتر از مقدار گره های فرزند است.

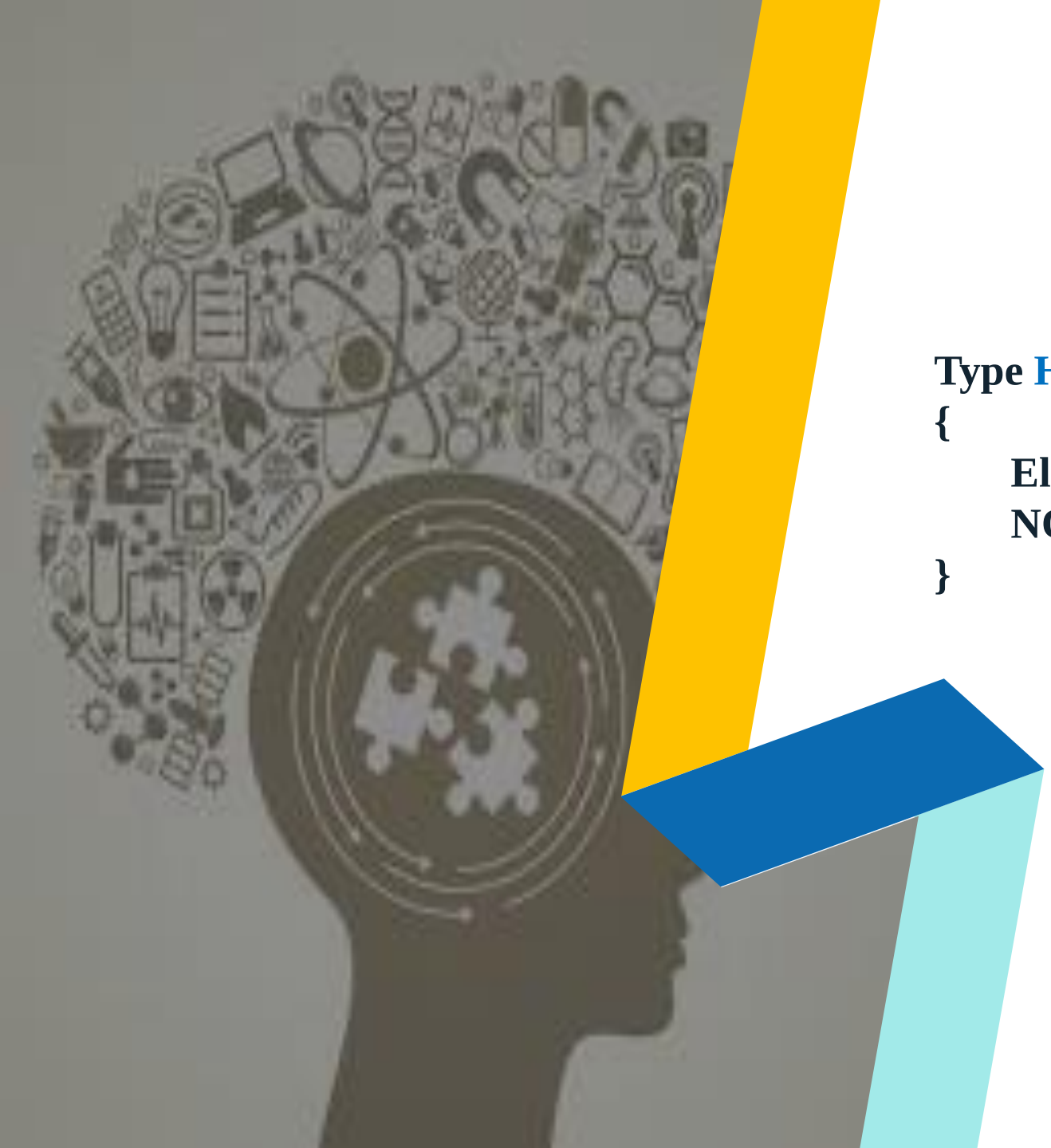
۲) هرم کمینه (هرم کوچکی) : مقدار موجود در هر گره، کوچکتر از مقدار گره های فرزند است.

بزرگترین (کوچکترین) عنصر یک برگ است و فرزندی ندارد.

نمایش هرم بصورت آرایه



1	2	3	4	5	6	7	8	9
Z	M	R	H	A	F	C	D	B



ساختمان داده هرم

```
Type HEAP = Record
{
      Elements : ElementType [Max];
      NOE : Integer;
}
```

درج در هرم

عملیات درج در هرم شامل دو قسمت است :
۱) قرار دادن عنصر جدید در انتهای لیست.
۲) برقراری شرط هرم.

قسمت اول سبب می شود که یک برگ به انتهای آرایه هرم اضافه شود. در نتیجه سطح آخر هرم، از چپ به راست پر می شود.
در قسمت دوم بررسی می شود اگر عنصر اضافه شده از گره ی پدر خود بزرگتر بود، عنصر اضافه شده با پدر خود جا به جا می شود. (به این عمل صعود حبابی یا به صورت ساده تر صعود گویند.)

پیاده سازی عملیات درج در هرم

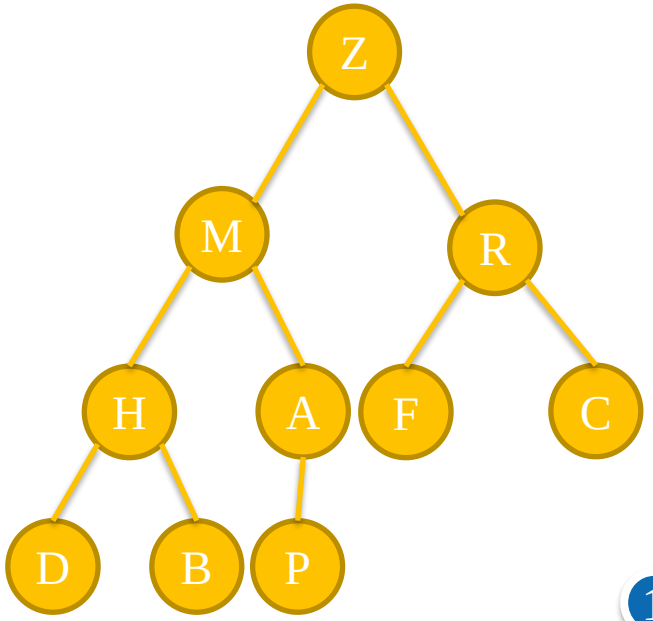
```
ENQUEUE_HEAP ( ref H : HEAP ; x : ElementType )
{
        H.NOE++;
        H.Elements [H.NOE] = x;
        BUBBLE_UP (H, H.NOE);
}
```


```
BUBBLE_UP ( ref H : HEAP ; i : Integer )
{
        if I <= 1 then
                return;
        if H.Elements[i] > H.Elements[i/2] then
                Swap (H.Elements[i] , H.Elements[i/2]);
                BUBBLE_UP (H, i/2);
}
```

MAX_HEAP

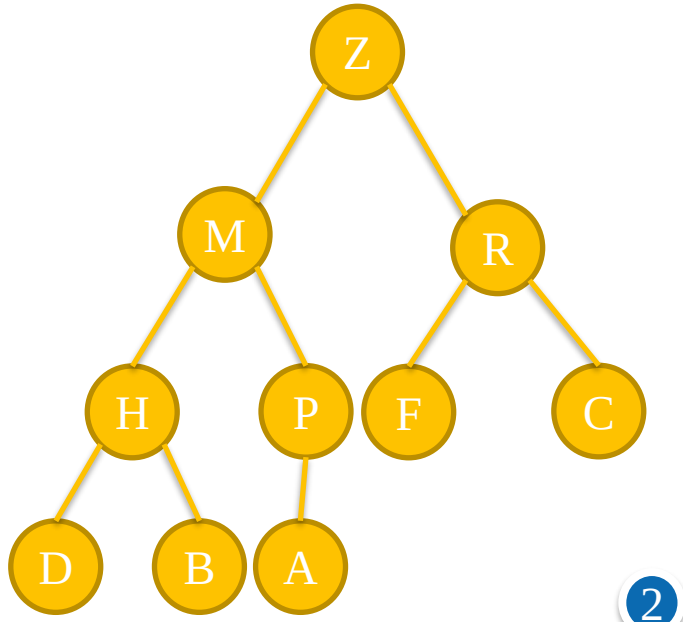
```
BUBBLE_UP ( ref H : HEAP ; i : Integer )
{
        if I <= 1 then
                return;
        if H.Elements[i] < H.Elements[i/2] then
                Swap (H.Elements[i] , H.Elements[i/2]);
                BUBBLE_UP (H, i/2);
}
```

MIN_HEAP



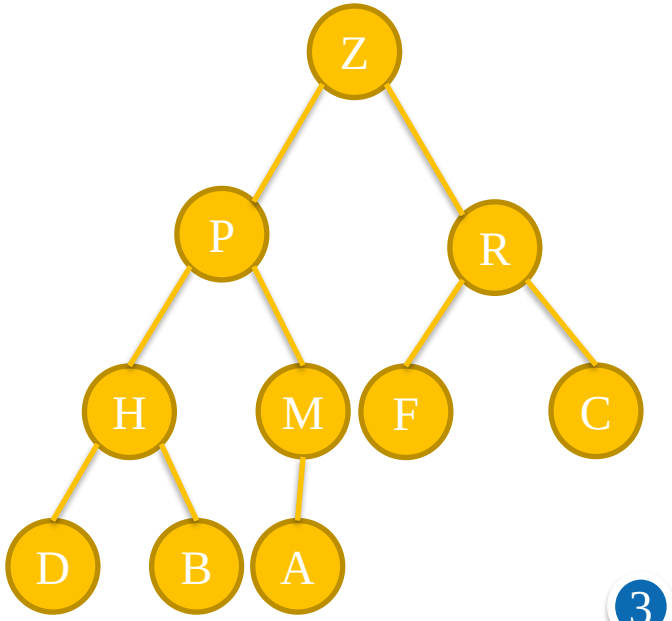
1

Z	M	R	H	A	F	C	D	B	P



2

Z	M	R	H	P	F	C	D	B	A



3

Z	P	R	H	M	F	C	D	B	A

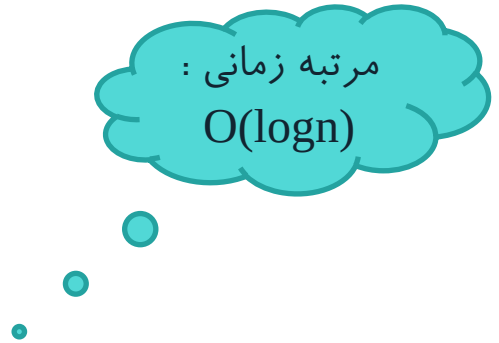
خروج از هرم

ویژگی خاص هرم بیشینه در قرار دادن بزرگترین عنصر در ریشه، به عملیات خروج از هرم صورت ویژه ای می دهد. به عبارتی خروج از هرم تنها محدود به عنصر موجود در ریشه است و تنها ریشه ی درخت که حاوی بزرگترین عنصر هرم بیشینه است می تواند از آن حذف شود.

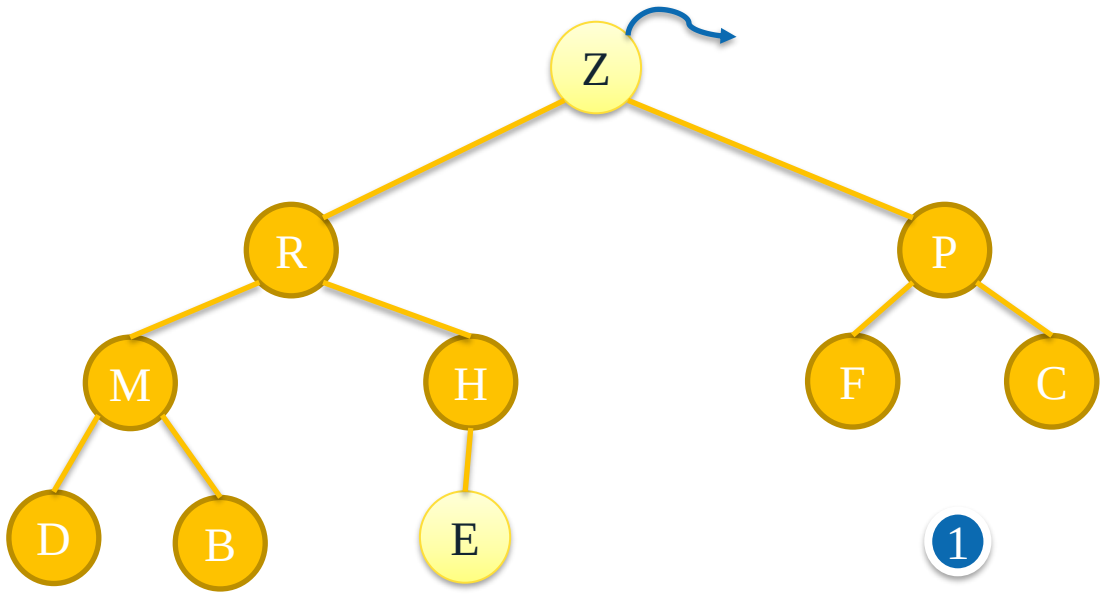
این عملیات شامل دو قسمت است :
۱) حذف ریشه و قرار گرفتن آخرین عنصر لیست در مکان ریشه
۲) برقراری شرط هرم

پیاده سازی عملیات خروج از هرم

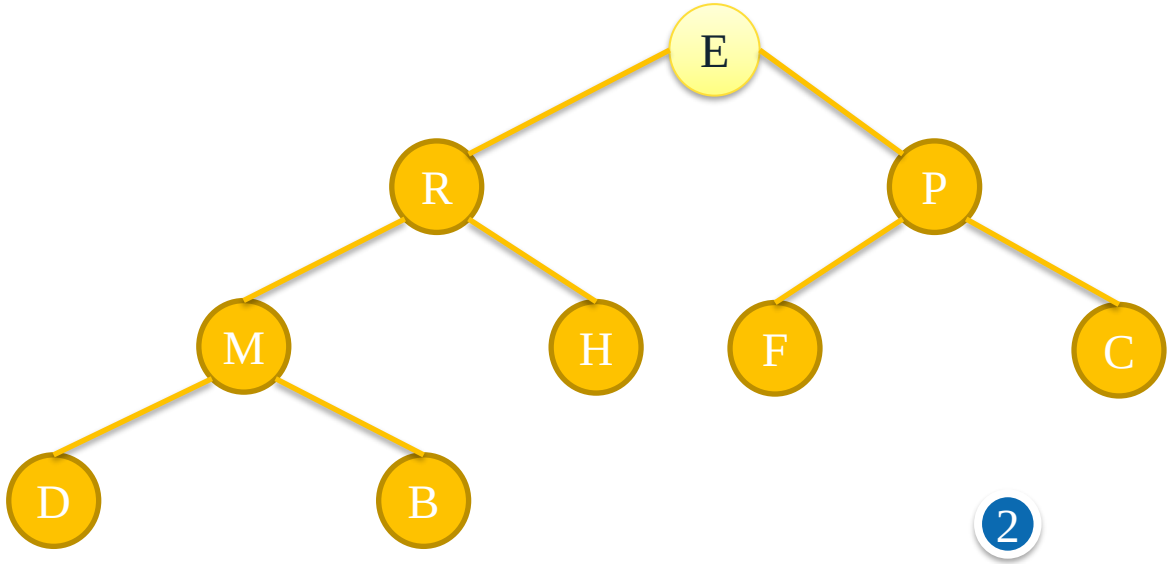
```
DEQUEUE_HEAP ( ref H : HEAP ) : ElementType
{
        maximum = H.Elements[1];
        H.Elements[1] = H.Elements[H.NOE];
        H.NOE--;
        SIFT_DOWN (H, 1);
        return maximum;
}
```



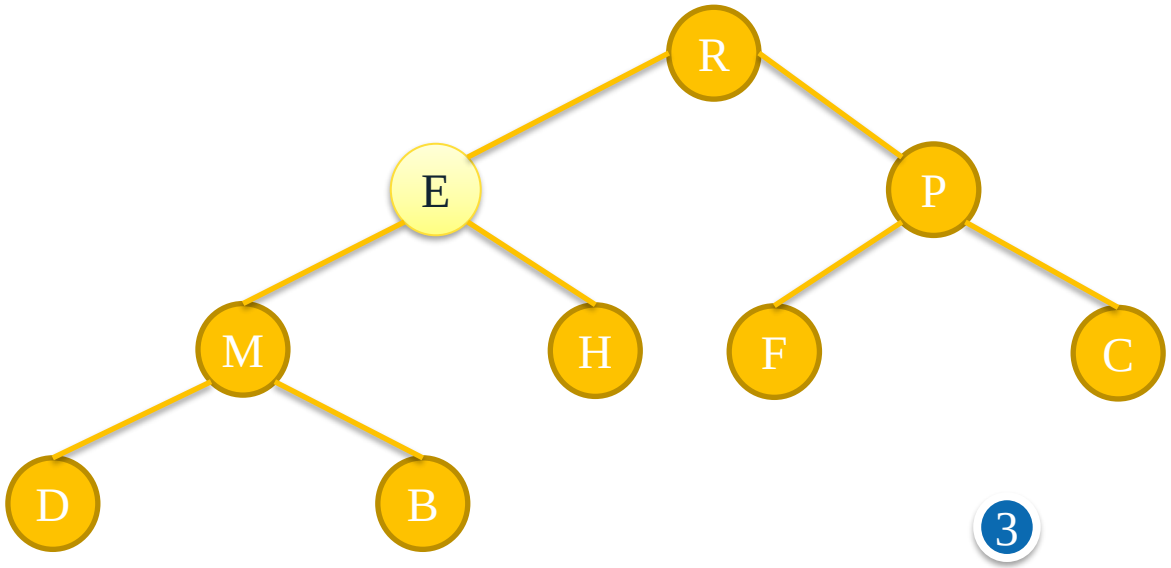

```
SIFT_DOWN ( ref H : HEAP ; i : Integer )
{
        largest = i;
        if 2*i <= H.NOE and H.Elements[largest] < H.Elements[2*i] then
                largest = 2*i;
        if 2*i+1 <= H.NOE and H.Elements[largest] < H.Elements[2*i+1] then
                largest = 2*i+1;
        if i != largest then
                Swap (H.Elements[i], H.Elements[largest]);
                SIFT_DOWN (H, largest);
}
```



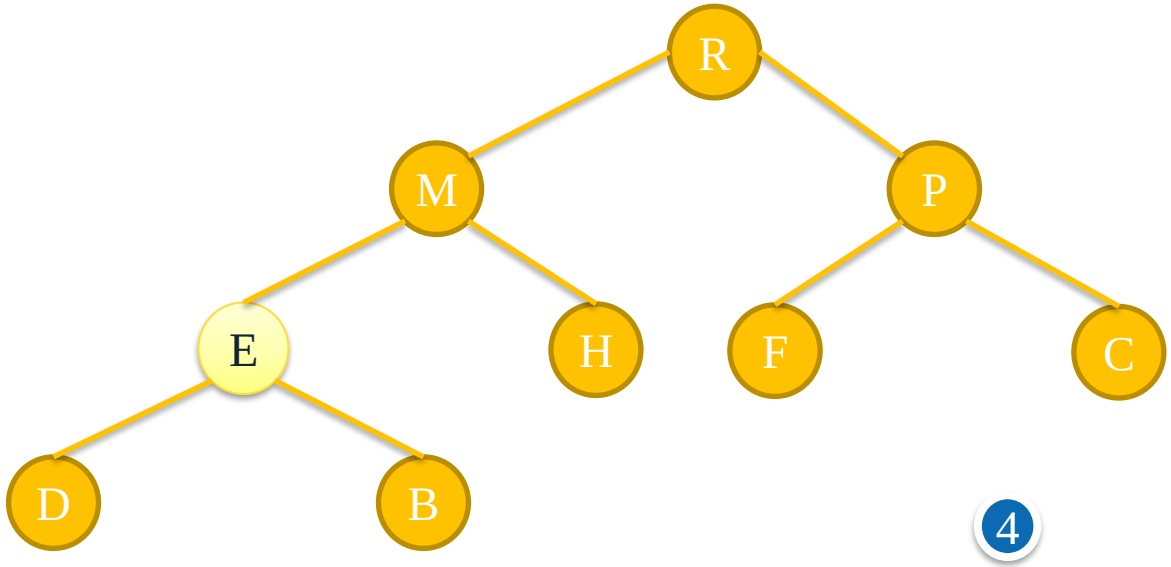
Z	R	P	M	H	F	C	D	B	E



E	R	P	M	H	F	C	D	B



R	E	P	M	H	F	C	D	B



R	M	P	E	H	F	C	D	B

ساخت هرم

یکی از راه های ممکن برای ساخت یک هرم از روی یک آرایه از عناصری با اندازه n با مرتبه ی خطی، اجرای الگوریتم SHIFT_DOWN به صورت بازگشتی بر روی عناصر آرایه است.
در این روش ابتدا زیردرخت های چپ و راست هر گره تبدیل به هرم شده و در نهایت تابع SHIFT_DOWN برای ریشه صدا زده می شود.

```
HEAPIFY1 ( ref H : HEAP ; i : Integer )
{
    if I > H.NOE/2 then
        return;
    HEAPIFY1 (H, i*2);
    HEAPIFY1 (H, i*2+1);
    SIFT_DOWN (H, i);
}
```

فرم بازگشتی

```
HEAPIFY2 ( ref H : HEAP )
{
    for i = H.NOE/2 downto 1 do
        SIFT_DOWN (H, i);
}
```

فرم غیر بازگشتی

مرتب سازی با استفاده از هرم

این عملیات شامل دو قسمت است :

۱) ساخت هرم

۲) n مرتبه خارج نمودن بزرگترین عنصر با صدا زدن تابع DEQUEUE_HEAP

بدین منظور، تابع DEQUEUE_HEAP بصورت دیگری پیاده سازی می شود :

```
DEQUEUE_HEAP2 ( ref H : HEAP )
{
        Swap (H.Elements[1], H.Elements[H.NOE]);
        H.NOE--;
        SIFT_DOWN (H, 1);
}
```

الگوریتم مرتب سازی هرمی

می توان الگوریتم مرتب سازی هرمی را با ترکیب توابع مذکور به صورت شبه کد زیر نوشت :

```
HEAP_SORT ( A : ARRAY ; n : Integer )
{
      for i = n/2 downto 1 do
            SIFT_DOWN (A, i);

      for i = n downto 1 do
            Swap (A[1] , A[i]);
            NOE--;
            SIFT_DOWN (A, i);
}
```

HEARIFY (A, n);

پیچیدگی زمانی :
O(nlogn)

پایان