لطفا اول فايل heapify مطالعه شود.

```
jmp for2
148
149
                                                                  int n = arr.length;
                                                                  // Build heap (rearrange array)
         endfor2:
152 :
153 endp
                                                                  for (int i = n / 2 - 1; i >= 0; i--)
        ret
                                                                      heapify(arr, n, i);
154
155 n db 5
                                                                  // One by one extract an element from hear
```

مقدار دهی N در خط 155 وجود دارد چون مقدار n در مسئله ثابت و 5 فرض شده است.

```
112 heapsort proc
                                                                    // Build heap (rearrange array)
                                                                    for (int i = n / 2 - 1; i >= 0; i--)
heapify(arr, n, i);
113
         mov al, n
         mov dl, 02h
114
         div dl
116
         dec al
                                                                    // One by one extract an element from heap
         mov i, al
                                                                    for (int i=n-1; i>=0; i--)
118
         for1: CMP i, 0000h
```

تمام نكته هاش قبلا گفته شده

```
mov i, al
for1: CMP i, 0000h
jl next
                                                               // Build heap (rearrange array)
                                                               for (int i = n / 2 - 1; i >= 0; i --)
                                                                   heapify(arr, n, i);
    mov al, n
```

نكته هاش قبلا گفته شده

شرطو چک میکنه اگه i کوچکتر از 0 بود از حلقه میاد بیرون

```
// Dutto Heah (Leat.Lauge attray)
for (int i = n / 2 - 1; i >= 0; i--)
heapify(arr, n, i);
// One by one extract an element from heap
for (int i=n-1; i>=0; i--)
    // Move current root to end
    int temp = arr[0];
```

ورودی مند m بود

اینجا میخوایم N رو پاس بدیم به متد

یس مقدار n رو میریزیم تو m

```
// Build heap (rearrange array)
for (int i = n / 2 - 1; i >= 0; i--)
    heapify(arr, n, i);
// One by one extract an element from heap
for (int i=n-1; i>=0; i--)
{
```

مثل عكس قبل

```
118
119
```

```
mov 1, al
forl: CMP i, 0000h
117
119
              jl next
              mov al, n
             mov m, al
             mov ah, i
123
             mov j, ah
124
             CALL heapify
125
              dec i
126
              jmp for1
```

```
mov i, al
118
        for1: CMP i, 0000h
119
            jl next
            mov al, n
            mov m, al
            mov ah, i
123
            mov j, ah
124
            CALL heapify
```

```
mov i, al
                                                                   // Build heap (rearrange array)
118
         for1: CMP i, 0000h
                                                                  for (int i = n / 2 - 1; i >= 0; i--)
119
             jl next
                                                                       heapify(arr, n, i);
             mov al, n
            mov m, al
                                                                   // One by one extract an element from heap
            mov ah, i
                                                                  for (int i=n-1; i>=0; i--)
             mov j, ah
124
             CALL heapify
             dec i
                                                                       // Move current root to end
             jmp for1
126
                                                                       int temp = arr[0];
                                                                                           توضيح خاصى وجود نداره
                                                                          توضيح غير خاصش ميشه: مند فراخواني ميشه
            mov m, al
                                                                   // Build heap (rearrange array)
            mov ah, i
            mov j, ah
                                                                   for (int i = n / 2 - 1; i >= 0; i -- )
124
             CALL heapify
                                                                       heapify(arr, n, i);
            dec i
126
             jmp for1
                                                                   // One by one extract an element from heap
                                                                            خط 126 هم برای انجام عملیات تکرار است
                            به این خط که میرسه میره به اول حلقه و اونجا شرط بزرگتر مساوی بودن i از 0 بررسی می شود.
129
         next: mov al, n
                                                                  for (int i = n / 2 - 1; i >= 0; i--)
            dec al
                                                                       heapify(arr, n, i);
            mov i, al
         for2: cmp i, 0000h
                                                                  // One by one extract an element from heap
            jl endfor2
                                                                  for (int i=n-1; i>=0; i--)
134
            mov bl, i
136
            mov bh, 00h
                                                                       // Move current root to end
            mov dl, arr[0000h]
                                                                       int temp = arr[0];
            mov dh, arr[bx]
138
                                                                       arr[0] = arr[i]:
                                                                                                    ا مقدار دهی میشه
                                                                   // One by one extract an element from heap
133
            jl endfor2
                                                                   for (int i=n-1; i>=0; i--)
134
            mov bl, i
                                                                                                            مثل قبل
                                                                  // Une by one extract an element from heap
             jl endfor2
134
                                                                  for (int i=n-1; i>=0; i--)
             mov bl, i
             mov bh, 00h
136
                                                                       // Move current root to end
            mov d1, arr[0000h]
mov dh, arr[bx]
mov arr[0000h], dh
                                                                      int temp = arr[0];
arr[0] = arr[i];
arr[i] = temp;
```

139 140

141 142 mov arr[bx], dl

mov j,00h

تو فایل heapify شبیهشو داشتیم جا به جا میشن دو مقدار

// call may bearify on the neduced bear

```
mov j,00h

141

142

mov al, i

mov m, al

145

call heapify

dec i

148

jmp for2

mov j,00h

// call max heapify on the reduced hea

heapify(arr, i, 0);

}
```

ورودی های متد مقدار میگیرن و متد فراخوانی میشه

خط 147 و 148 هم مثل حلقه قبلى

سوالی بود در خدمتم بقیش که خواندن ورودی و نمایش ورودیه