

## لطفا اول فایل heapify مطالعه شود.

```

148     jmp for2
149
150
151     endfor2:
152     ret
153 endp
154
155 n db 0
156 i db 0

```

```

int n = arr.length;

// 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

```

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

```

111
112 heapsort proc
113     mov al, n
114     mov dl, 02h
115     div dl
116     dec al
117     mov i, al
118     for1: cmp i, 0000h
119     jle next

```

```

// 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--)
    heapify(arr, n, i);

```

تمام نکته هاش قبلا گفته شده

```

116     dec al
117     mov i, al
118     for1: cmp i, 0000h
119     jle next
120     mov al, n

```

```

// Build heap (rearrange array)
for (int i = n / 2 - 1; i >= 0; i--)
    heapify(arr, n, i);

```

نکته هاش قبلا گفته شده

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

```

117     mov i, al
118     for1: cmp i, 0000h
119     jle next
120     mov al, n
121     mov m, al
122     mov ah, i
123     mov j, ah
124     call heapify
125     dec i
126     jmp for1
127

```

```

// 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--)
{
    // Move current root to end
    int temp = arr[0];

```

ورودی متد m بود

اینجا میخوایم N رو پاس بدیم به متد

پس مقدار n رو میریزیم تو m

```

116     dec al
117     mov i, al
118     for1: cmp i, 0000h
119     jle next
120     mov al, n
121     mov m, al
122     mov ah, i
123     mov j, ah
124     call heapify
125     dec i

```

```

// 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--)
{
    // Move current root to end
    int temp = arr[0];

```

مثل عکس قبل

```

117     mov i, al
118 for1: cmp i, 0000h
119     jl next
120     mov al, n
121     mov m, al
122     mov ah, i
123     mov j, ah
124     CALL heapify
125     dec i
126     jmp for1
127

```

```

// 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--)
{
    // Move current root to end
    int temp = arr[0];

```

توضیح خاصی وجود ندارد

توضیح غیر خاصش همیشه: متد فراخوانی همیشه

```

121     mov m, al
122     mov ah, i
123     mov j, ah
124     CALL heapify
125     dec i
126     jmp for1
127

```

```

// 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

```

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
130     dec al
131     mov i, al
132 for2: cmp i, 0000h
133     jl endfor2
134
135     mov bl, i
136     mov bh, 00h
137     mov dl, arr[0000h]
138     mov dh, arr[bx]
139     mov arr[0000h], dh

```

```

// 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--)
{
    // Move current root to end
    int temp = arr[0];
    arr[0] = arr[i];

```

مقداردهی همیشه

```

132 for2: cmp i, 0000h
133     jl endfor2
134
135     mov bl, i

```

```

// One by one extract an element from heap
for (int i=n-1; i>=0; i--)
{

```

مثل قبل

```

133     jl endfor2
134
135     mov bl, i
136     mov bh, 00h
137     mov dl, arr[0000h]
138     mov dh, arr[bx]
139     mov arr[0000h], dh
140     mov arr[bx], dl
141
142     mov j, 00h

```

```

// 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];
    arr[0] = arr[i];
    arr[i] = temp;

    // call max heapify on the reduced heap

```

تو فایل heapify شبیهشو داشتیم جا به جا میشن دو مقدار

<pre> 141 142 143 144 145 146 147 148 149 </pre>	<pre> mov j, 00h mov al, i mov m, al call heapify  dec i jmp for2 </pre>	<pre> arr[i] = comp;  // call max heapify on the reduced heap heapify(arr, i, 0); } } </pre>
--	--	--

ورودی های متد مقدار میگیرن و متد فراخوانی میشه

خط 147 و 148 هم مثل حلقه قبلی

سوالی بود در خدمتم

بقیش که خواندن ورودی و نمایش ورودیه