1(1) heapSort 함수에서 입력 리스트의 트리에 대해

① 첫 번째 for문

② 두번째 for문 수행에서 리스트의 상태를 단계적으로 나타내라

```
void heapSort(element a[], int n)
{/* perform a heap sort on a[1:n] */
  int i,j;
  element temp;

  for (i = n/2; i > 0; i--)
    adjust(a,i,n);
  for (i = n-1; i > 0; i--) {
    SWAP(a[1],a[i+1],temp);
    adjust(a,1,i);
  }
}
```

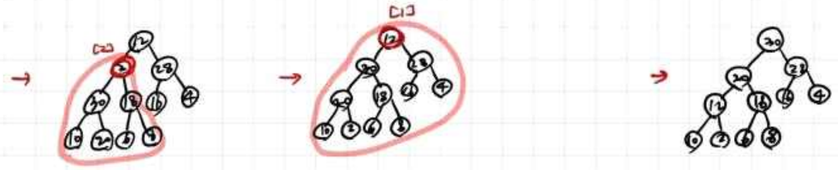
	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
Input:	12	2	16	30	8	28	4	10	20	6	18

① 첫 번째 for문

n=11

→ i = n/2 = 5

12 2 16 30 8 28 4 10 20 6 18 | 12 2 16 30 18 28 4 10 20 6 8 | 12 2 16 30 18 28 4 10 20 6 8

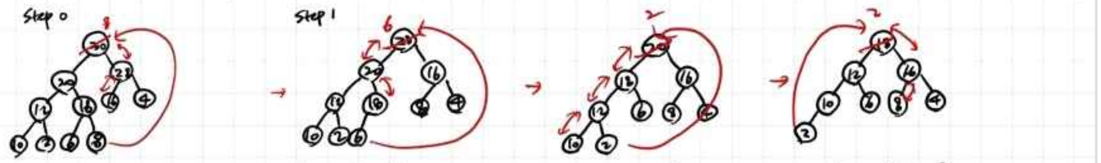


12 2 28 30 18 16 4 10 20 6 8 | 12 30 28 20 18 16 4 10 2 6 8 | 30 20 28 12 18 16 4 10 2 6 8

after Initialization of max heap

② 두 번째 for문 (매번 adjust를 수행한 직후의 상태)

Step 0 Step 1

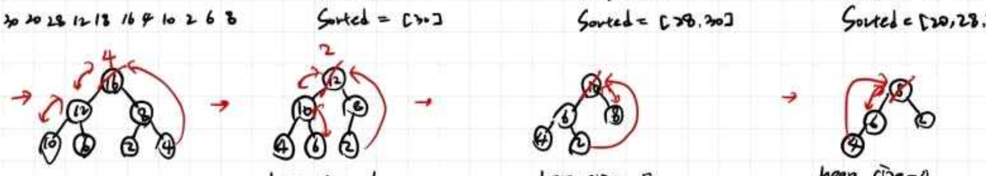


30 20 28 12 18 16 4 10 2 6 8

heap size = 10
Sorted = [30]

heap size = 9
Sorted = [28, 30]

heap size = 8
Sorted = [20, 28, 30]



heap size = 7
Sorted = [18, 20, 28, 30]

heap size = 6
Sorted = [16, 18, 20, 28, 30]

heap size = 5
Sorted = [12, 16, 18, 20, 28, 30]

heap size = 4
Sorted = [10, 12, 16, 18, 20, 28, 30]

heap size = 3
Sorted = [8, 10, 12, 16, 18, 20, 28, 30]

heap size = 2
Sorted = [6, 8, 10, 12, 16, 18, 20, 28, 30]

heap size = 1
Sorted = [4, 6, 8, 10, 12, 16, 18, 20, 28, 30]

→ Sorted = [2, 4, 6, 8, 10, 12, 16, 18, 20, 28, 30]

2(1)

(1) 기수정렬 과정을 그림으로 나타내고 (r=10)

d = 2 (자릿수)   n=11 (개수)

Input : 12 2 16 30 8 28 4 10 20 6 18

InitList[1] InitList[2] InitList[3] InitList[4] InitList[5] InitList[6] InitList[7] InitList[8] InitList[9] InitList[10] InitList[11]

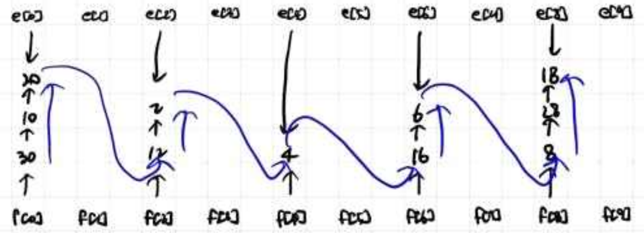
[12] → [2] → [16] → [30] → [8] → [28] → [4] → [10] → [20] → [6] → [18]

Index :	1	2	3	4	5	6	7	8	9	10	11
link :	2	3	4	5	6	7	8	9	10	11	0
a(InitList):	12	2	16	30	8	28	4	10	20	6	18

→ 다음에 가리킬 Index값

first : 1

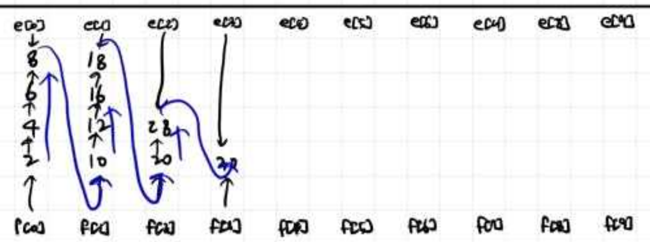
result: 12 2 16 30 8 28 4 10 20 6 18



Index :	1	2	3	4	5	6	7	8	9	10	11
link :	2	7	10	8	6	11	3	9	1	5	0
a(InitList):	12	2	16	30	8	28	4	10	20	6	18

first : 4

result: 30 10 20 12 2 4 16 6 8 28 18



Index :	1	2	3	4	5	6	7	8	9	10	11
link :	3	7	11	0	8	4	10	1	6	5	9
a(InitList):	12	2	16	30	8	28	4	10	20	6	18

first : 2

result: 2 4 6 8 10 12 16 18 20 28 30