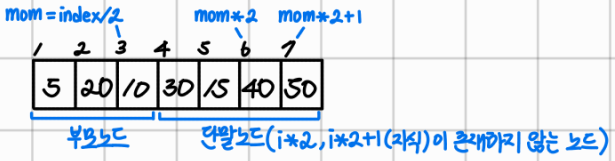
n개의 데이터 중 최대값을 찾기 위해 n개의 승간만 필요했던 변형 토너먼트 방식에 있어 새롭게 삽입되는 데이터를 부모와 비교하며 자리 바꿈을 통해 최대값을 배열의 1번 인덱스에 위치하게 하는 방식 외 미리 모든 데이터를 배열에 삽입하고 진행하는 방식도 가능하다.

아래 예제는 새로운 방식이다.

입력 : 5, 20, 10, 30, 15, 40, 50

전체 7개의 데이터이므로 마지막 인덱스는 7. 따라서, 7/2 = 3. 즉, 인덱스 3이 자식노드를 가지고 있는 마지막 노드임.
= index

인덱스 3부터 역순으로,



* 두 자식 중 더 큰 자식의 인덱스를 son으로 설정

array[mom*2] > array[mom*2+1]

son = mom*2

* mom과 son 비교 후 son이 더 크면 둘의 위치 swap

array[mom] < array[son]

swap(&array[mom], &array[son])

☆ mom = son

→ 이제서 mom의 자식 노드가 더 있는지 추가 탐색

while(mom >= index/2) ← 조건 충족하면

* 10의 자식 노드가 없으므로 끝

인덱스 2 탐색 후, 5 30 50 20 15 40 10

인덱스 1 탐색 후, 50 30 5 20 15 40 10 → 50 30 40 20 15 5 10
구현 중요 포인트

출력 : 5, 20, 50, 30, 15, 40, 10
5, 30, 50, 20, 15, 40, 10
50, 30, 40, 20, 15, 5, 10

조건 : 전체 for loop의 조건은 i = n/2 to 1 이다.

```
main(void){
    int size;
    int tournamentArray[200];
    size = read_Data(tournamentArray);
    makeTournament(size, tournamentArray);
}
```