

Heap-append

[code] DS1901-HW02

Problem

원소값(key)들이 모두 다른 정수배열 $A[0..n+m-1]$ 이 존재한다 (n, m 은 자연수). 이 때 $A[0..n-1]$ 은 Heap이지만 $A[n..n+m-1]$ 은 Heap을 만족하지 않는 원소들이다. 이 배열 A 전체를 Heap로 바꾸는 `max_heap_append` 함수를 작성하시오.

[제약조건]

- 함수의 prototype은 다음과 같다.
`void max_heap_append(int A[], int p, int q);`
- p : 위 문제 설명에서 $n-1$
 q : 위 문제 설명에서 $n+m-1$
- 반드시 $A[n], A[n+1], \dots, A[n+m-1]$ 의 순서대로 upward 방향의 Heapify를 통해 Heap을 생성한다. (즉, Priority queue의 HEAP-INCREASE-KEY 함수를 이용한다.)
- 주의) 제출하는 소스파일에는 main 함수는 포함하지 않는다.
(다음 페이지 예처럼 hw02.c만 제출한다.)

[Hints]

- 실습 시간 `heap_sort`에서 사용된 코드 일부를 사용한다.

Submission

Due: 4월 12일 (금) 23시 59분 59초

19950001@ubuntu:~/DS\$ submit DS1901-HW02 mycode.c

Example of main()

```
///// main.c
#include <stdio.h>
#include <stdlib.h>
void max_heap_append(int A[], int p, int q);
void build_max_heap(int A[], int n);

int main(int argc, char *argv[]) {
    int A[9] = {1, 8, 6, 3, 2, 7, 4, 9, 10};
    int n=4;
    build_max_heap(A, n);
    max_heap_append(A, n, 8);
    for ( int i=0; i<9; i++ ) printf("%d ",
A[i]); printf("\n");

    return 0;
}
```

```
///// mycode.c
...
void max_heap_append(int A[], int p, int r)
{
    ...
}
...
void build_max_heap(int A[], int n) {
    ...
}
...
```

[2개의 소스파일로 구현하여 테스트하는 방법]

```
$ gcc mycode.c main.c
```

```
$ a.out
```

Self Test

```
19950001@ubuntu:~/DS$ sftest DS1901-HW02 mycode.c
```

```
MSG> hw02.c was compiled.
```

```
1 th. Testing 3 5 3 1 2 4 6 ---
```

```
Your answer is 6 4 5 2 3 1 : success.
```

```
2 th. Testing 3 5 1 3 6 4 2 ---
```

```
Your answer is 6 5 3 1 4 2 : success.
```

```
3 th. Testing 5 5 4 3 1 2 6 7 8 9 ---
```

```
Your answer is 9 8 6 7 2 3 5 1 4 : success.
```