

자료구조응용

16. Winner Tree (15점)

2020.5.11.(월)

1. [승자트리를 이용한 정렬] k 개의 런에 대해 승자트리를 사용한 합병정렬을 수행하라. 단, 각 런은 10개의 정렬된 정수로 구성되어 있으며, k 는 2의 거듭제곱 (power of 2)임을 가정하라.

(1) 실행순서

- ① 난수생성을 위한 seed와 k 를 입력받는다.
- ② $k \times 10$ 의 2차원 배열을 생성한다.
- ③ 1~100 사이에서 발생시킨 k 개의 난수를 각 런의 첫 번째 key로 지정한다. 각 런의 2~10 번째 키는 첫 번째 키에서 1씩 순서대로 증가하여 저장한다 * 각 key는 중복 가능하다.

생성된 k 개의 난수($k=8$)

10	9	20	6	8	9	90	17
11	10	21	7	9	10	91	18
12	11	22	8	10	11	92	19
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
19	18	29	15	17	18	99	26

- ④ ③에서 생성한 키 데이터에 대해 초기 승자트리를 구성한다. * `initWinner()`
* winner tree 는 완전이진트리이며 노드 레벨에 따라 배열에 순차적으로 저장된다.
* winner tree 의 각 노드는 키 값(혹은 레코드)에 대한 포인터(혹은 인덱스)만을 가진다.
* `initWinner()`는 recursive postorder traversal 형태로 작성한다.
- ⑤ 승자트리를 사용한 정렬을 수행한다.
* { 최소키를 sorted 배열에 저장 -> 최소키가 최종 승자로 선택된 레코드에는 다음 키 값으로 `nums` 배열의 값을 치환(레코드의 키 값을 다 소진한 경우에는 무한대를 의미하는 임의의 값으로 최소키를 치환) -> 승자트리를 재구성(`adjustWinner()`) } 이 과정을 $k * 10$ 번 반복함
* `adjust` 시, 치환된 키의 index --> parent index --> sibling index를 구할 수 있음. 치환된 키와 sibling 키의 비교를 루트 방향으로 수행함
- ⑥ 정렬된 결과 배열의 인덱스 순서대로 키 값을 출력한다.

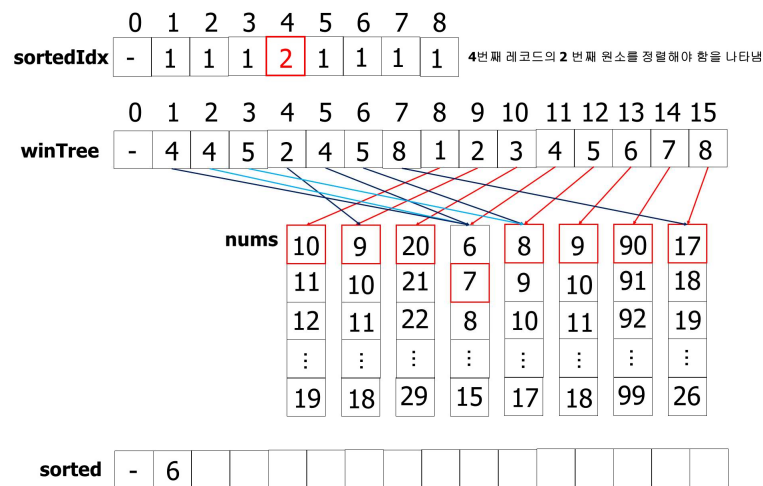
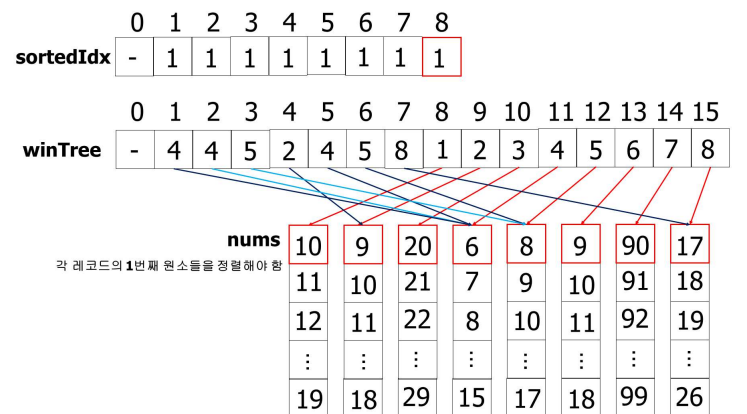
(2) 구현세부사항

① 함수정의 및 트리선언

```
#define MAX_SIZE 100
#define INF_NUM 10000

// min - winner tree
int nums[ MAX_SIZE+1 ][11] = { 0 };           // keys to sort
int winTree[ 2*MAX_SIZE ] = { NULL };         // winner tree
int sorted[ MAX_SIZE*10+1 ] = { 0 };          // sorted result
int initWinner(int cur, int k, int winTree[]);
void adjustWinner(int min, int k, int winTree[]);
```

② k=8일 경우, 초기 min - winner tree 구성 예



4번째 레코드의 key 값 6이 정렬되고 난 후의 sortedIdx, winTree, nums의 상태

