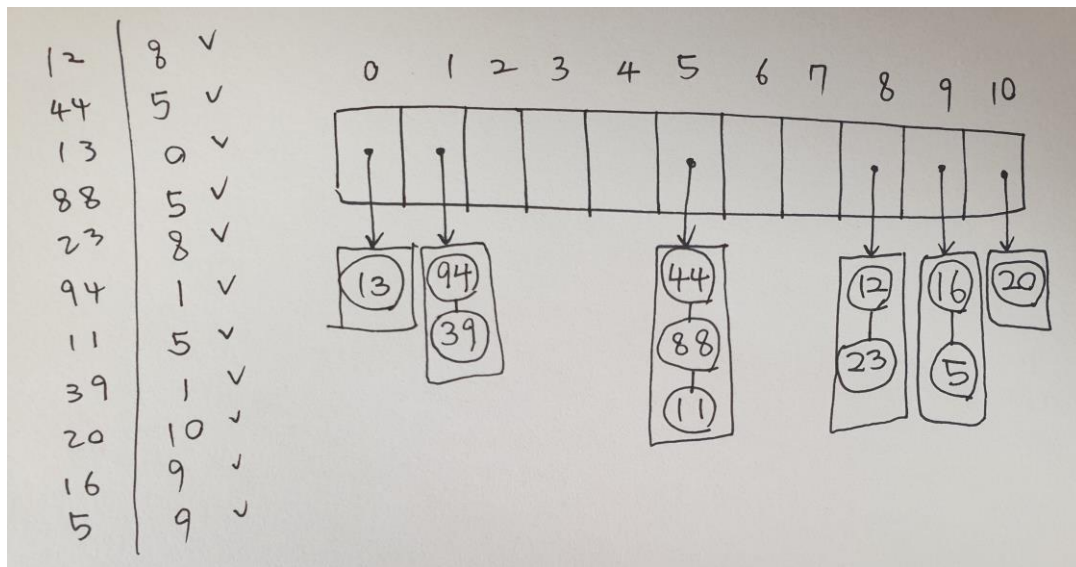


데이터 구조 Assn7

20180551 이준석

POVIS ID : ljs9904ljs

Problem 1 (R-9.7)



Problem 2 (R-9.8)

0	1	2	3	4	5	6	7	8	9	10	11	12
13	94	39			44	88	11	12	23	20	16	5

Problem 3 (R-9.15)

- 최악의 경우는 모든 값들의 key가 같은 경우이다.
 - n개의 원소를 삽입하므로 그 때는 $O(n^2)$ 이다.

- 최고의 경우는 모든 값들의 key가 다른 경우이다.
 - n개의 원소를 삽입하므로 그 때는 $O(n)$ 이다.

Problem 4 (R-9.16)

R-9.16 Draw an example skip list that results from performing the following series of operations on the skip list shown in Figure 9.12: `erase(38)`, `insert(48,x)`, `insert(24,y)`, `erase(55)`. Record your coin flips, as well.

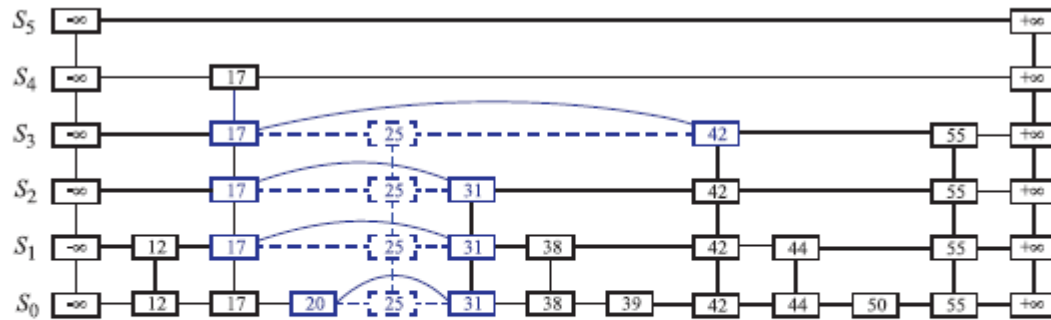


Figure 9.12: Removal of the entry with key 25 from the skip list of Figure 9.11. The positions visited after the search for the position of S_0 holding the entry are highlighted in blue. The positions removed are drawn with dashed lines.

원본

S5												
S4		17										
S3		17					42			55		
S2		17		31			42			55		
S1	12	17		31	38		42	44		55		
S0	12	17	20	31	38	39	42	44	50	55		

`erase(38)` 수행

S5												
S4		17										
S3		17					42			55		
S2		17		31			42			55		
S1	12	17		31			42	44		55		
S0	12	17	20	31		39	42	44	50	55		

insert(48,x) 수행

S5												
S4		17										
S3		17					42				55	
S2		17		31			42		(48,x)		55	
S1	12	17		31			42	44	(48,x)		55	
S0	12	17	20	31		39	42	44	(48,x)	50	55	

head가 2번 나왔다.

insert(24,y) 수행

S5												
S4		17										
S3		17					42				55	
S2		17			31		42		(48,x)		55	
S1	12	17			31		42	44	(48,x)		55	
S0	12	17	20	24	31	39	42	44	(48,x)	50	55	

head가 0번 나왔다.

erase(55) 수행

S5												
S4		17										
S3		17					42					
S2		17			31		42		(48,x)			
S1	12	17			31		42	44	(48,x)			
S0	12	17	20	24	31	39	42	44	(48,x)	50		

Problem 5 (C-9.1)

firstEntry(k): Return an iterator to the entry with smallest key value; if

the map is empty, it returns end.

lastEntry(k): Return an iterator to the entry with largest key value; if

the map is empty, it returns end.

ceilingEntry(k): Return an iterator to the entry with the least key value greater than or equal to k ; if there is no such entry, it returns end.

floorEntry(k): Return an iterator to the entry with the greatest key value less than or equal to k ; if there is no such entry, it returns end.

lowerEntry(k): Return an iterator to the entry with the greatest key value less than k ; if there is no such entry, it returns end.

higherEntry(k): Return an iterator to the entry with the least key value greater than k ; if there is no such entry, it returns end.

이것들을 스킵 리스트로 구현하는 것이 문제이다.

s는 skip list의 가장 왼쪽 상단이다.

<firstEntry(k)>

input : skip list

output : the smallest key value

p = s

while(below(p)가 NULL이 아닐 때까지)

{

p = below(p)

}

p = after(p)

return key(p)

<lastEntry(k)>

input : skip list

output : the largest key value

```

p = s
while ( below(s)가 NULL이 아닐 때까지 )
{
    p = below(p)
}
while( after(p)가 NULL이 아닐 때까지 )
{
    p = after(p)
}
return key(p) // p가 가리키는 것의 key value

```

<ceilingEntry(k)>

input : skip list

output : the least key value greater than or equal to k

```

p = s
while ( below(s)가 NULL이 아닐 때까지 )
{
    p = below(p)
}
while( k >= key( after(p) )
{
    p = after(p)
}
return key(p)

```

<floorEntry(k)>

input : skip list

output : the greatest key value less than or equal to k

$p = s$

while (below(s)가 NULL이 아닐 때까지)

{

$p = \text{below}(p)$

}

while (after(p)가 NULL이 아닐 때까지) // 가장 마지막 key값으로 간다.

{

$p = \text{after}(p)$

}

while($k \leq \text{key}(\text{before}(p))$) // 가장 마지막 key값에서 왼쪽 방향으로 탐색한다.

{

$p = \text{before}(p)$

}

return key(p)

<lowerEntry(k)>

input : skip list

output : the greatest key value less than k

$p = s$

while (below(s)가 NULL이 아닐 때까지)

{

$p = \text{below}(p)$

}

```
while( k > key( after(p) )
```

```
{
```

```
p = after(p)
```

```
}
```

```
return key(p)
```

<higherEntry(k)>

input : skip list

output : the least key value greater than k

```
p = s
```

```
while ( below(s)가 NULL이 아닐 때까지 )
```

```
{
```

```
p = below(p)
```

```
}
```

```
while ( after(p)가 NULL이 아닐 때까지 ) // 가장 마지막 key값으로 간다.
```

```
{
```

```
p = after(p)
```

```
}
```

```
while( k < key( before(p) )
```

```
{
```

```
p = before(p)
```

```
}
```

```
return key(p)
```