SOGANG UNIVERSITY

멀티미디어 프로그래밍
Mid term Report

V2017110 김병준

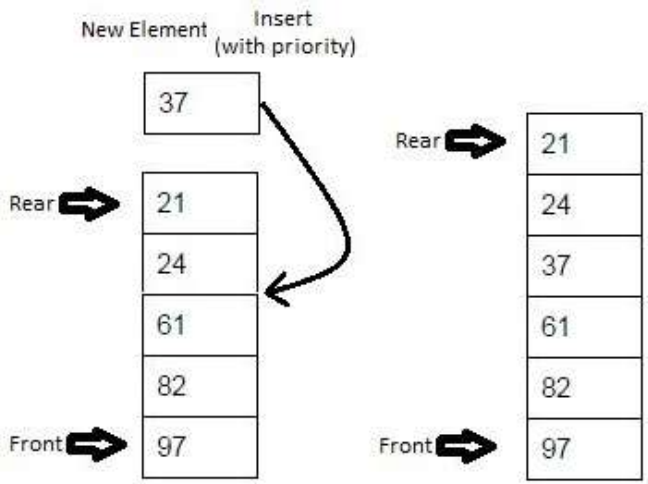
Index

- Priority Queue & Heap Sorting
- Binary Indexed Tree (Fenwick Tree)
- Dijkstra Algorithm for Single Source Shortest Path

Priority Queue & Heap Sorting

- Queue
  - FIFO(first in first out), 순서화된 리스트
  - front에서 삭제되고 rear(back)에서 입력된다.

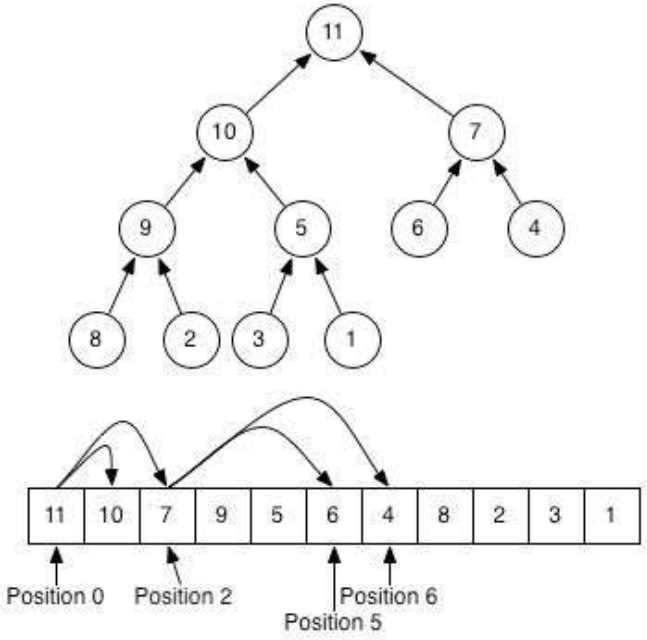
- Priority Queue
  - 일반적인 queue와 같이 입력은 rear(back)
  - 최소값 or 최대값이 우선순위를 가지고 나온다.
  - priority queue의 구현
    - 배열 비교
    - 연결 리스트 이용
    - heap structure 이용
  - 배열, 연결리스트 이용은 최대 시간복잡도가 크다.



Priority Queue & Heap Sorting

❖ Heap structure

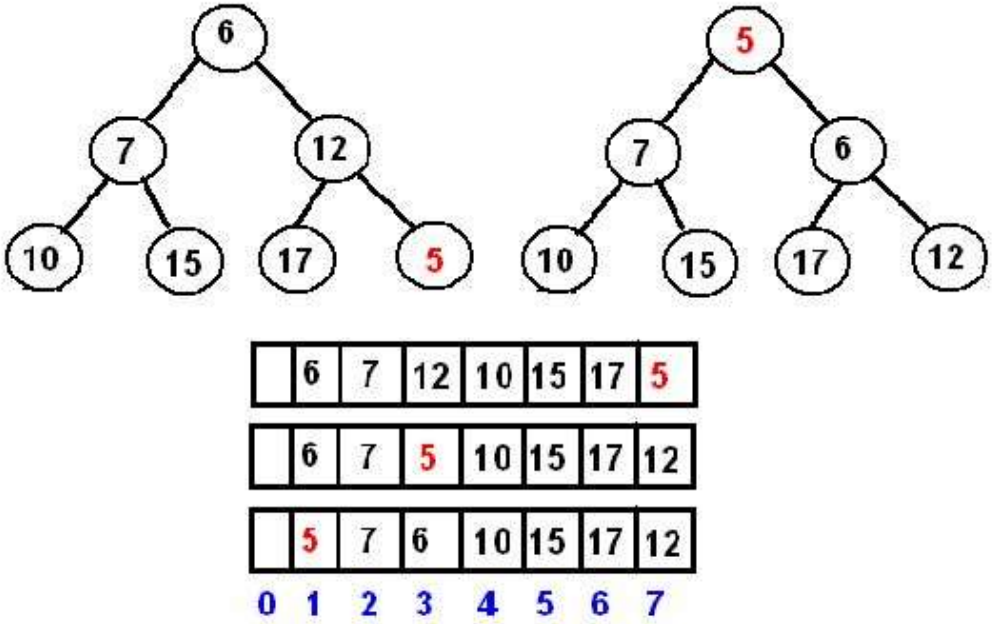
- 완전 이진 트리 구조
- 하나의 노드는 2개의 노드에 상속된다.
- 시간 복잡도가 최소, 최대, 평균이 모두 같다.
- Max/Min Heap 로 우선순위에 따라 차례로 OUT.
- Max Heap 인 경우 루트 노드(최상단)의 값이 max
- 각 노드의 값은 자식 노드의 값과 비교하여 정렬



Priority Queue & Heap Sorting

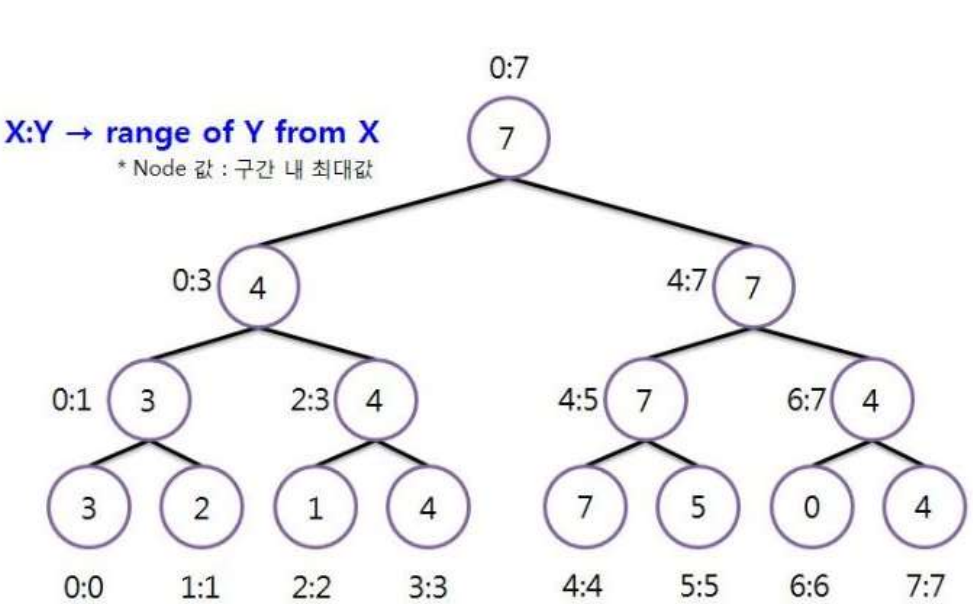
❖ Heap Sorting
- 새로운 data가 입력 되면 해당 노드와 부모 노드의 값을 비교하여 정렬한다.
- max heap 이면 더 클 때, min heap이면 더 작을 때 교환 하여 정렬한다.
- data가 OUT 될 때 루트 노드의 data가 사용되고 마지막 노드의 값이 시작 노드로 온 후 자식 노드과 비교하여 정렬

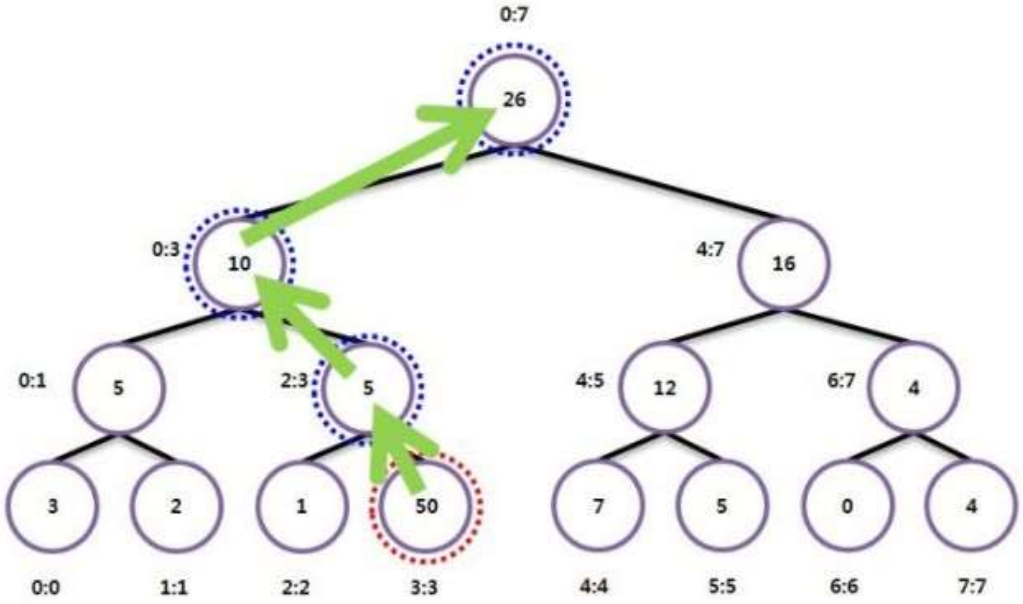
- 평균 시간 복잡도가 우수
- 왼쪽 노드의 INDEX
  - 2k
- 오른쪽 노드의 INDEX
  - 2k+1



Binary Indexed Tree (Fenwick Tree)

- Segment Tree 의 종류
- Segment Tree
  - 완전 이진트리 형태
  - 부모노드는 자식 노드의 특정값 (ex. 최대값/ 합)
  - data에서 특정 구간에 대한 연산에 유리

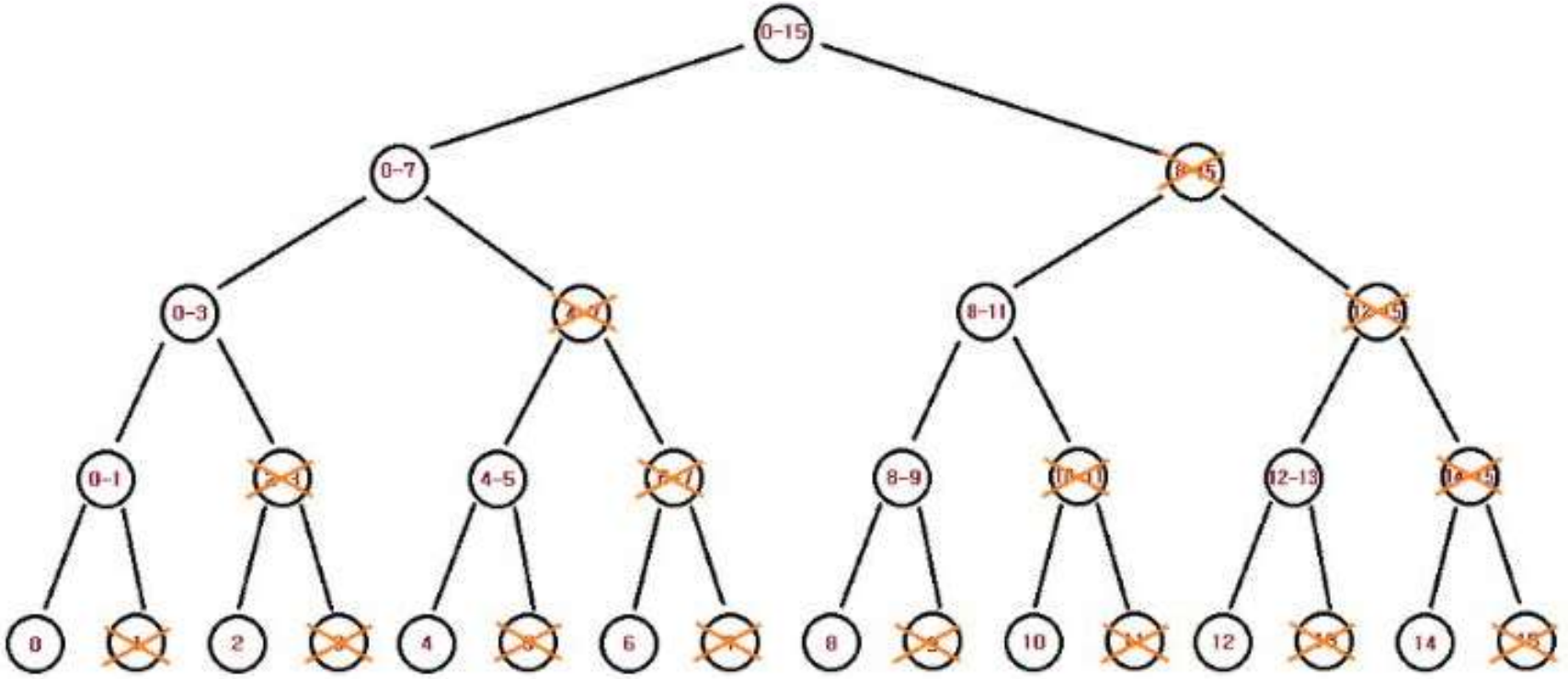




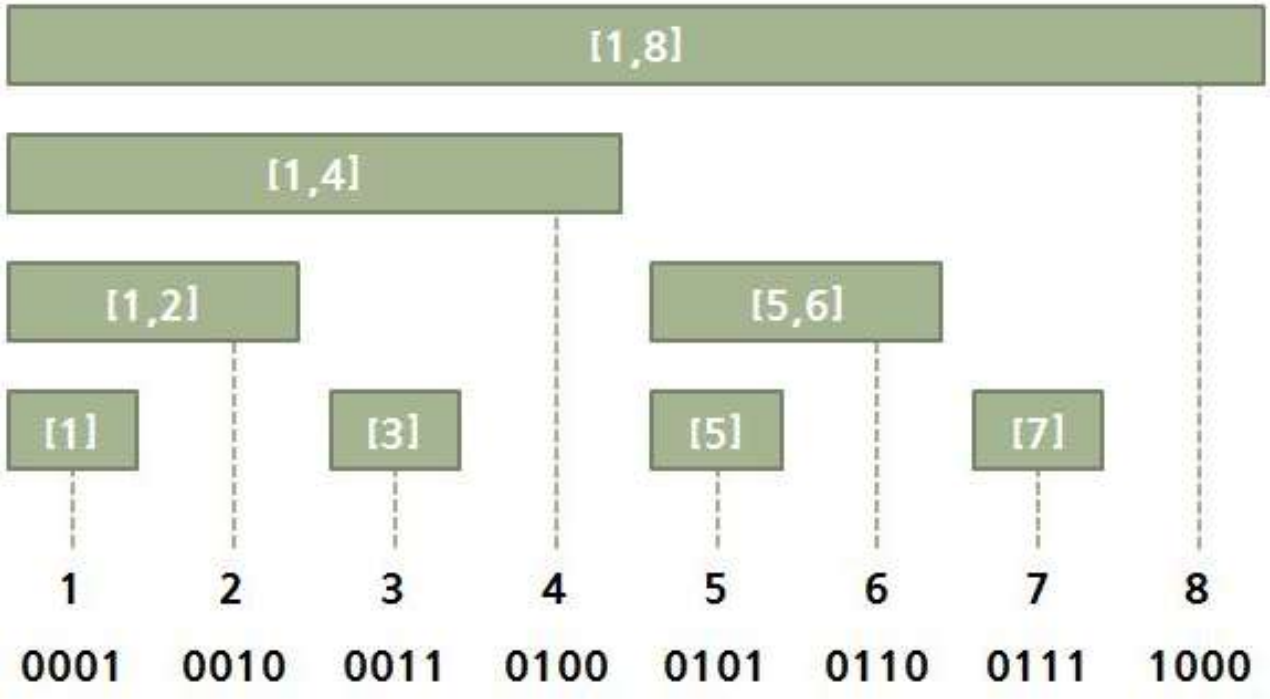
Binary Indexed Tree (Fenwick Tree)

❖ Fenwick Tree

- 특정구간의 특정값을 효율적으로 연산
- [a,b] : a번째 ~ b번째의 값의 특정값은 [1:b] - [1:a-1] 로 구할 수 있다.
  이때 각 노드가 binary로 index가 주어져 있어 [1:x]의 값을 특정 연산 없이 얻는다.



Binary Indexed Tree (Fenwick Tree)

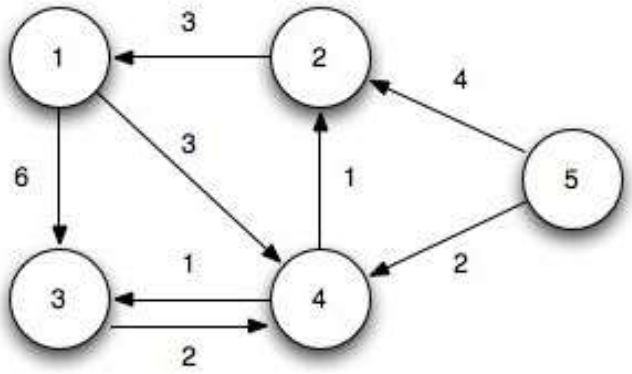


- 각 노드는 binary index에서 첫번째 1을 가지는 자릿수 만큼의 data 길이의 특정값을 나타낸다.
- 입력 data가 변할때 해당 상위 노드만 변경
- binary index를 이용하기에 구현 및 연산 최소화
- 시간 복잡도(구간 연산 & 수정) : O(nlogn)

Dijkstra Algorithm for Single Source Shortest Path

- 하나의 출발점에서 모든 점으로의 최단 경로를 구하는 문제
- 첫 시작점에서 연결되있는 경로를 추가하며 최단거리를 갱신해나간다.
- 초기화 값은 무한대를 가진다.
- 각 경로는 weight를 가진다.

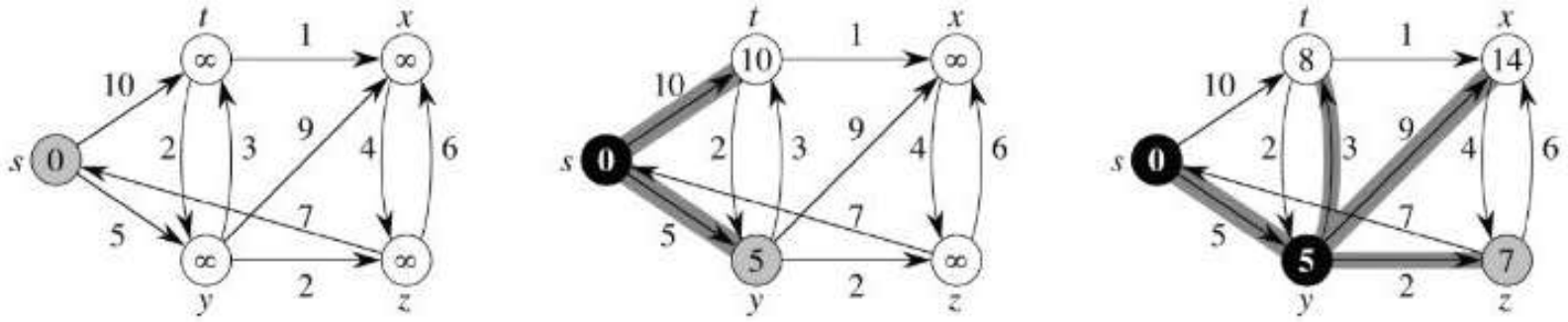
  – 시작점이 5인 경우 연결된 각 경로의 가중치들을 합하여 갱신한다.



5	4	3	2	1
0	INF	INF	INF	INF
	2		4	
		3	3	
				6

Dijkstra Algorithm for Single Source Shortest Path

- 기본적으로 BFS(너비우선탐색)을 발전시킨 알고리즘
- 가중치가 음수일 경우에는 불가능



- 경로 탐색의 순서는 연결된 노드의 가중치를 기록 하고 방문은 경로 합이 가장 적은 연결로 진행한다.
- 모든 경로를 방문 후에 하나의 시작점에서 모든 점으로의 최단 경로가 기록된다.