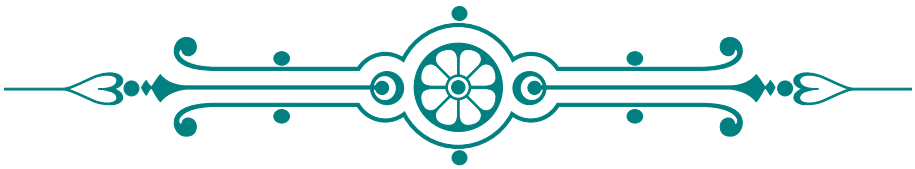


컴퓨터 알고리즘과 실습

학번 : 2016110056

학과 : 불교학부

이름 : 박승원

날짜 : 2017년 3월 22일

문제 1. 합병 정렬 (50 점)

- 수업시간에 순환적(recursive) 합병 정렬 방법에 대해서 배웠다. 이를 비순환적(iterative) 합병정렬로 바꾸어보자.
- 순환적 합병 정렬을 구현해 보아라. (20 점)
  - 코드 제출
  - 입력 A=[30 20 40 35 5 50 45 10 25 15]에 대한 정렬 결과를 출력하여라. 단계별로 값이 어떻게 변화되는지와 최종 결과물을 출력할것.
- 이를 비순환적 합병 정렬로 구현해 보아라. (30 점)
  - 코드 제출
  - 입력 A=[30 20 40 35 5 50 45 10 25 15]에 대한 정렬 결과를 출력하여라. 단계별로 값이 어떻게 변화되는지와 최종 결과물을 출력할것.
- 위 두경우 단계별로 값이 어떻게 변화되는지는 자신만의 방식으로 합병정렬이 되어가고 있다는 과정을 보여주면 됨. Ex) 배열에 값이 바뀔때마다 출력하거나 각단계마다 A 또는 버퍼(수업시간에 B[]) 를 출력하거나 뭐든 합병정렬을 하고있다는 증빙이 되면 됨.

```
// 2016110056   박승원
#include<string>
#include<random>
#include<cstring>
#include<iostream>
using namespace std;

void swap(int* p, int* q) {
    int tmp = *p;
    *p = *q;
    *q = tmp;
}

void merge(int* p, int* m, int* q) {
    int sz = q - p;
    int ar[sz];
    int* const partition = m;
    int* const start = p;
    for(int i=0; i<sz; i++) {
        if(p == partition ) ar[i] = *m++;
```

```
        else if(m == q) ar[i] = *p++;
        else ar[i] = *p < *m ? *p++ : *m++;
    }
    memcpy(start, ar, sizeof(ar));
}

void merge_sort(int* p, int* q) {
    if(q - p <= 1) return;
    int middle = (q - p) / 2;
    merge_sort(p, p + middle);
    merge_sort(p + middle, q);
    merge(p, p + middle, q);

    for(int i=0; i<q-p; i++) cout << p[i] << ' ';
    cout << endl;
}

void iterative_merge_sort (int* p, int* q) {
    int j=2;
    for(; j<q-p; j*=2) {
        for(int* r = p; r +j/2 < q; r += j) merge(r, r + j/2, min(r + j, q));
        for(int i=0; i<10; i++) cout << p[i] << ' ';
        cout << endl;
    }
    merge(p, p + j/2, q);
}

int main(int ac, char** av) {
    int ar[] = {30, 20, 40, 35, 5, 50, 45, 10, 25, 15};
     iterative_merge_sort (ar, ar+10);
    cout << "비순환적 합병 정렬 결과 " << endl;
    for(int i=0; i<10; i++) cout << ar[i] << ' ';

    int ar2[] = {30, 20, 40, 35, 5, 50, 45, 10, 25, 15};
    merge_sort(ar2, ar2+10);
    cout << "순환적 합병 정렬 결과 " << endl;
    for(int i=0; i<10; i++) cout << ar2[i] << ' ';
}
```

```
evince report.pdf
zezeon@ubuntuZ:~/Programming/basicProgramming$ make tex
g++ 1.cpp -o 1.x -g -fmax-errors=1 -lm -std=c++14
//2016110056 박승원
#include<string>
#include<random>
#include<cstring>
#include<iostream>
using namespace std;
void swap(int* p, int* q) {
        int tmp = *p;
        *p = *q;
---------- 문제 1번 실행을 시작합니다. --------------------------
./1.x
20 30 35 40 5 50 10 45 15 25
20 30 35 40 5 10 45 50 15 25
5 10 20 30 35 40 45 50 15 25
비순환적 합병 정렬 결과
5 10 15 20 25 30 35 40 45 50 20 30
5 35
5 35 40
5 20 30 35 40
45 50
15 25
10 15 25
10 15 25 45 50
5 10 15 20 25 30 35 40 45 50
순환적 합병 정렬 결과
5 10 15 20 25 30 35 40 45 50 ---------- 문제 1번 실행을 종료합니다. ---------------
--------------
```

문제 2. 힙정렬 (50점)

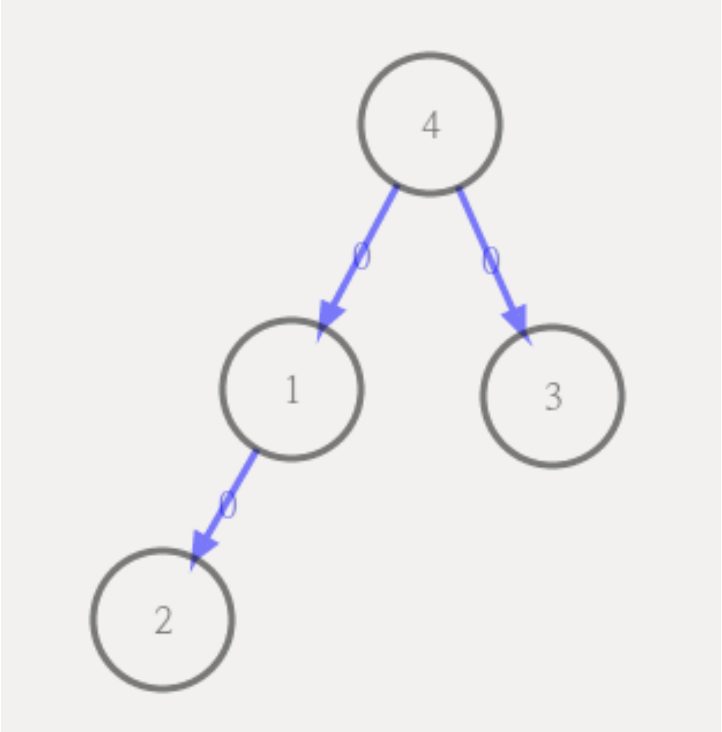
- 수업시간에 배운 힙정렬을 구현해보자. A = [4, 1, 3, 2, 16, 9, 10, 14, 8, 7]
  - 힙생성 (25점)
    - 입력 A 에대해 힙생성 알고리즘(수업시간에 배운 알고리즘 2를 이용할 것)을 이용했을때 트리구조가 각 단계별로 어떻게 변화하는지 이진트리를 그려 보시오. 이는 수업때 Lecture_3 p20 에서 그려보았습니다.
    - 이를 코드로 구현해보고 자신이 구현한 코드를 이용하여 생성된 힙을 일차원 배열로 출력하시오.
  - 생성된 힙을 이용하여 힙정렬 (25점)
    - 루트로 부터 하나씩 값을 바꾸어가며 단계별로 힙이 어떻게 변화하는지 이진트리를 그려보시오. 이는 수업때 Lecture_3 p23 에서 그려보았습니다.
    - 이를 코드로 구현해보고 자신이 구현한 코드를 이용하여 정렬된 일차원 배열을 출력하시오.

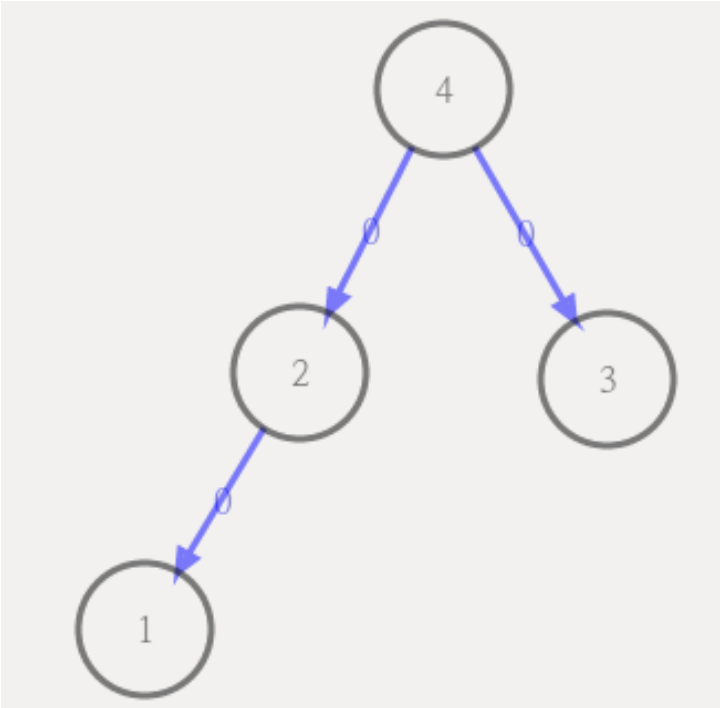
```
// 2016110056   박승원
#include<iostream>
#include<stdio.h>
#define MAX 300
using namespace std;
typedef int element;
element heap[MAX];
int sz = 0;

void swap(element* a, element* b) {
    element tmp = *a;
    *a = *b;
    *b = tmp;
}

int threesome(int i) {
    if(heap[i] < heap[i * 2] || heap[i] < heap[i * 2 + 1]) {
        int big = heap[i*2] > heap[i*2+1] ? i*2 : i*2+1;
        swap(&heap[big], &heap[i]);
        return big;
```

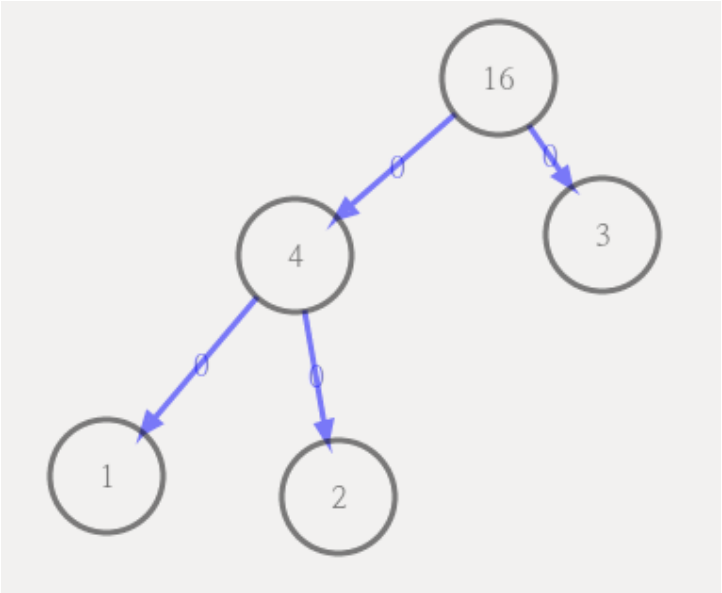
```
    } else return 0;
}

void insert(element n) {
    heap[++sz] = n;
    for(int i=sz; i>1; i/=2)
        if(heap[i] > heap[i/2]) swap(&heap[i], &heap[i/2]);
        else break;
}

void show() {
    for(int i=1; i<=sz; i++) printf("%d ", heap[i]);
}

element pop() {
    heap[0] = heap[1]; // use for return
    heap[1] = heap[sz--];
    int n = 1;
    while(n) n = threesome(n);
    return heap[0];
}


int main()
{
    int A[] = {4, 1, 3, 2, 16, 9, 10, 14, 8, 7};
    for(auto& a : A) insert(a);
    for(int i=0; i<sz; i++) cout << heap[i] << ' ';
    cout << endl;
    int B[10];
    for(int i=0; sz; i++) B[i] = pop();
    for(int i=0; i<10; i++) cout << B[i] << ' ';
}
```

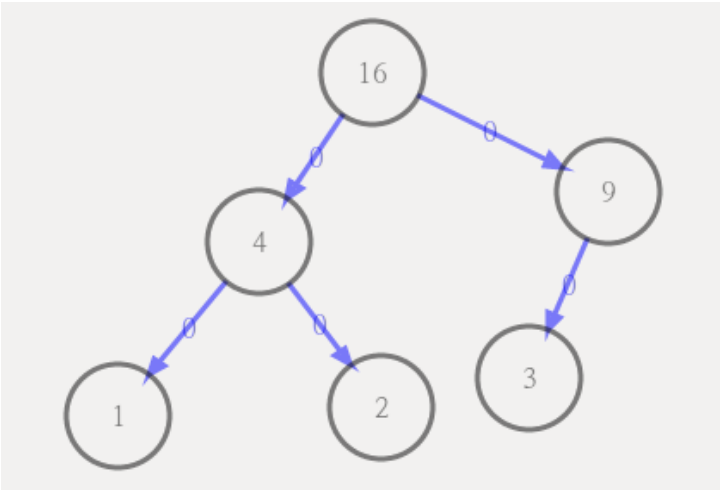
```
10-21.pdf  12.x       3num.x    6.png   Makefile     func.c       sscan.x
10.png     2.c        4.c       6.x     Session.vim  func.png     varg
10.x       2.cpp      4.cpp     7.c     boo.c        func.x
11-11.pdf  2.png      4.png     7.png   boo.png      logo.jpg
zezeon@ubuntuZ:~/Programming/basicProgramming$ make tex
g++ 2.cpp -o 2.x -g -fmax-errors=1 -lm -std=c++14
//2016110056 박승원
#include<iostream>
#include<stdio.h>
#define MAX 300
using namespace std;
typedef int element;
element heap[MAX];
int sz = 0;
void swap(element* a, element* b) {
---------- 문제 2번 실행을 시작합니다. ----------
./2.x
0 16 14 10 8 7 3 9 1 4
16 14 10 9 8 7 4 3 2 2 ---------- 문제 2번 실행을 종료합니다. ----------
---------
```

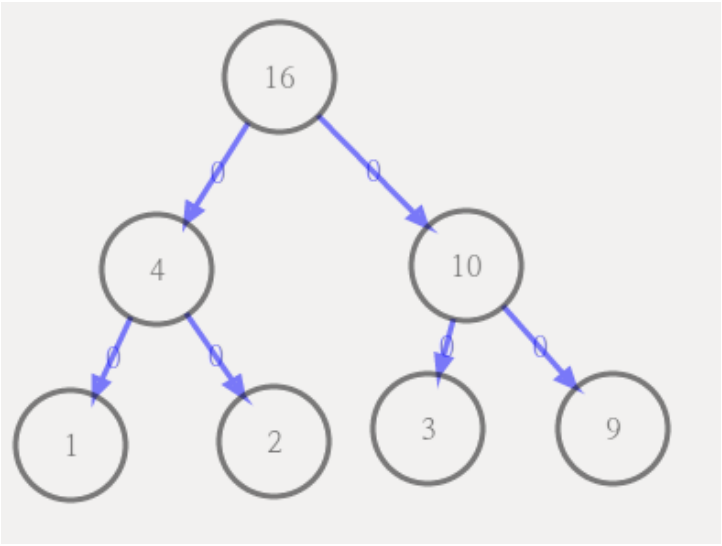
제 1 절　힙을 형성하는 과정에서의 힙의 형태

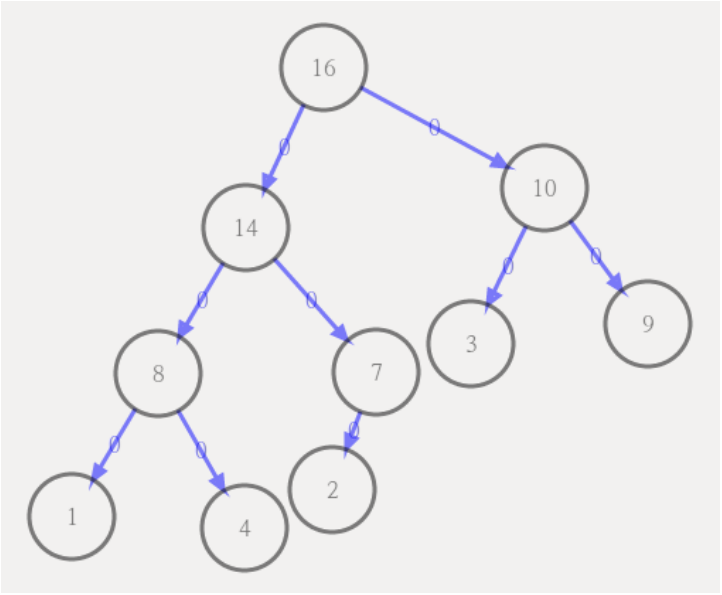




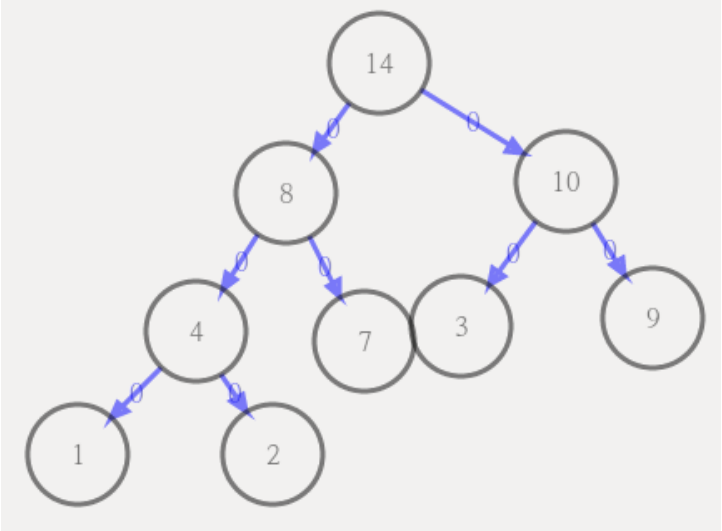


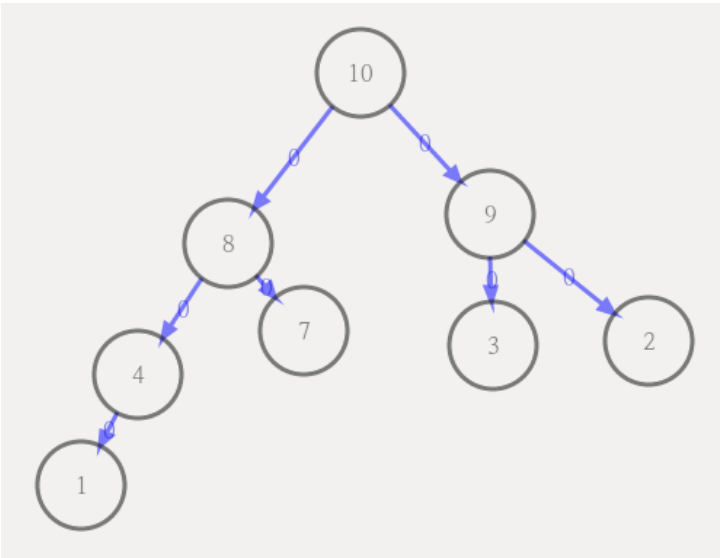


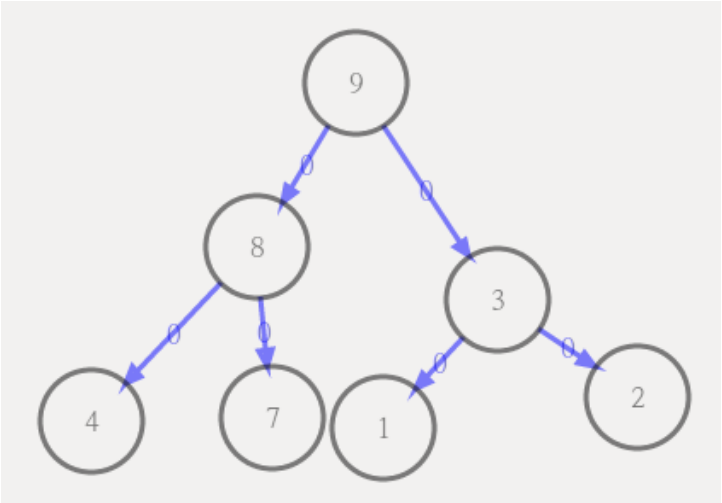


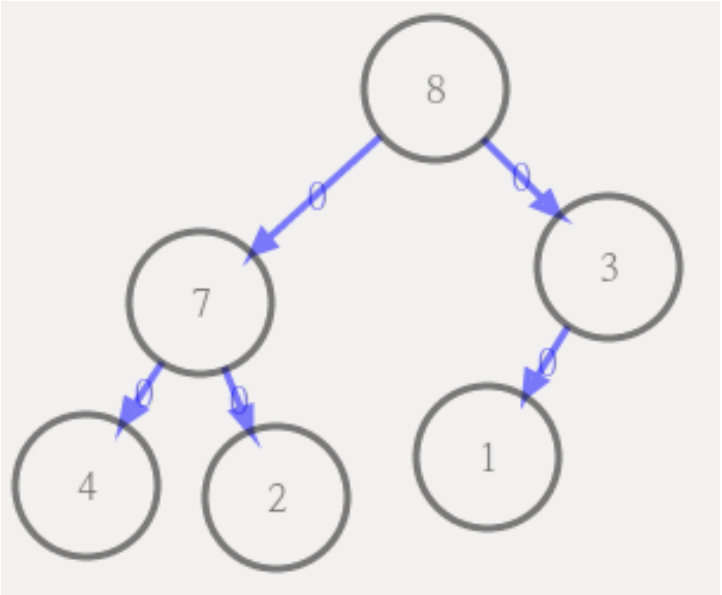


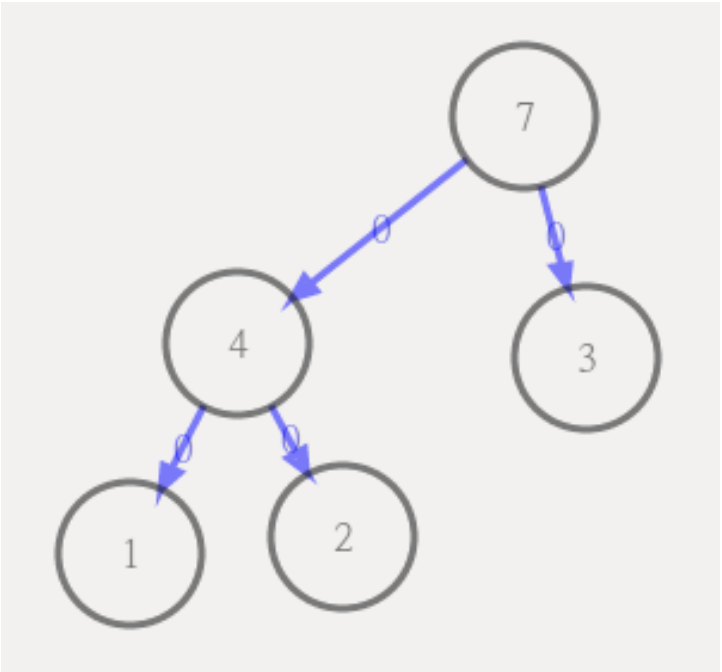
제 2 절　힙에서 원소를 빼어내는 과정에서의 힙의 형태

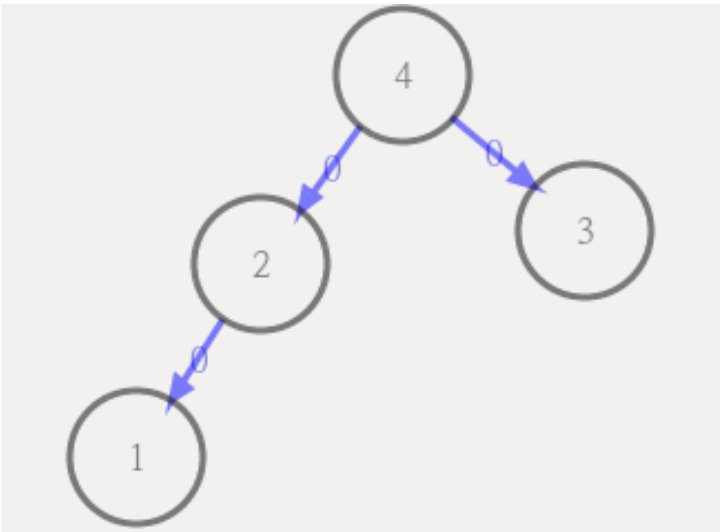





9
8
3
4
7
1
2


8
7
3
4
2
1


7
4
3
1
2


4
2
3
1