



# 4주차 퀴즈

📅 Date	
☰ Tags	취준스터디
🔗 출처	
📎 참고자료	

## 정렬

1. <https://www.acmicpc.net/problem/1377>

해당문제를 보고 버블정렬의 어떤특징이 문제에 사용되었는지 말하고,문제의 수도코드를 작성하시오.

버블정렬의 swap이 한번도 일어나지 않은 루프가 언제인지 알아내야 하는 문제이다.

swap이 일어나지 않은 경우 = 모든 데이터가 정렬되었다 는 것을 의미한다.

버블정렬의 경우 하나의 원소가 오른쪽으로 이동하는 것은 하나의 루프에서 여러번 발생할 수 있다 그러나, 왼쪽으로 가는 경우는 한 루프에서 한번 발생한다.

왼쪽으로 간 횟수 중 최댓값을 구한다면, 그 횟수는 정렬한 횟수가 될 수 있다.

(마지막에 break를 걸고 출력하기 때문에 +1을 해주어야한다, swap이 필요하지 않은 경우를 한번 거치고 break를 하기 때문이다)

```
bool change = false;
for (int i=1; i<=n+1; i++) {
    change = false;
    for (int j=1; j<=n-i; j++) {
        if (a[j] > a[j+1]) {
            change = true;
            swap(a[j], a[j+1]);
        }
    }
    if (change == false) {
        cout << i << '\n';
        break;//여기에서 출력하기 때문에 +1
    }
}
```

```

N(데이터 개수) A(데이터 리스트)

for N만큼 반복 :
    A리스트 저장

A리스트 정렬

for N만큼 반복 :
    A[i]의 정렬전 index - 정렬 후 index 계산의 최댓값을 찾아 저장

최댓값 + 1 을 정답으로 출력

```

2. 삽입정렬의 특징을 설명하고, 삽입정렬의 시간복잡도를 줄이는 방법에 대해 서술하시오.

삽입정렬의 시간 복잡도를 줄이는 방법으로는 삽입정렬의 과정 중 적절한 삽입 위치를 찾는 알고리즘을 이진탐색으로 이용하는 방법이 있다.

이진탐색을 왜 사용하는지에 대해 말하자면, 이진탐색의 경우 오름차순 혹은 내림차순으로 이미 정렬되어 있는 상태에서 탐색을 하는 알고리즘인데, 현재의 수를 key로 잡아 이전까지 정렬해두었던 앞의 배열에서 key가 들어갈 자리를 찾아 삽입정렬의 시간 복잡도를 낮출 수 있다.

## 가비지컬렉터

1. GC Algorithm 중 한가지를 단점 장점과 함께 설명하세요.

### Mark and Sweep

가비지 컬렉션의 루트에서 시작해 이 루트가 참조하는 모든 오브젝트, 그리고 오브젝트들이 참조하는 다른 오브젝트들을 탐색해 내려가며 마크한다. 해당 단계를 거치고 난 뒤 가비지 컬렉터는 힙 내부 전체를 돌면서 마크되지 않은 메모리들을 해제한다. 이 과정을 스윕이라고 부른다.

### 참고자료

### Parallel Compacting Collector

Old 영역에 대한 GC는 Mark-Summary-Compaction 방식으로 이루어진다.

살아있는 객체를 식별 후, 컴팩션 영역의 살아있는 객체를 조사한 뒤 컴팩션을 수행한다.

시리얼 컬렉터, 병렬 컬렉터의 마크스윕컴팩션과 병렬 컴팩팅 컬렉터의 마크 서머리 컴팩션의 차이점은 스윕과 종합 단계의 차이로 볼 수 있다.

스윕단계의 경우 단일 스레드가 **old 영역 전체를 훑지만** Summary 단계의 경우 여러개의 스레드가 Old영역을 분리하려 훑는다. 또한 수행한 GC에서 컴팩팅 된 영역을 별도로 훑는다 서머리 단계가 차이점이고 복잡하다는 점이 있다.

Parallel GC는 Minor GC에 대해서 멀티 스레딩으로 수행하는 것에 반해, Parallel Old GC는 Major GC에 대해서도 멀티 스레딩으로 수행한다.

## 2. G1GC 동작 방법을 간단히 설명하세요.

이해가 안감 → 추후 조사 필요

참고

# 브라우저

## 1. 브라우저의 동작과정 중 Operation에 대해 설명해주세요

Layout operation - 렌더트리 노드의 레이아웃을 생성한다. 레이아웃은 각 노드의 크기로 구성되고 화면에 포지션 별로 프린트 된다. 이 과정은 각 노드의 레이아웃들을 계산하기 때문에 레이아웃으로 불리운다.

그림을 그리기 전에 미리 얼마만큼 공간을 할당하며, 그 공간에 어떤 크기로 요소를 배치해야 될지 계획하는 단계

이 작업은 윈도우를 리사이징 할때 , 돔 요소를 조작할 때 발생한다. 그래서 웹페이지가 레이아웃으로 여러번 호출하는 것을 피해야한다 → 비용문제

Paint operation - 모양,위치,애니메이션을 자주 변경하는 CSS속성들 때문에 브라우저는 이에 대한 레이어를 만든다. 이 레이어는 브라우저가 웹페이지의 전체적인 주기를 효율적으로 수행할 수 있도록 도움을 준다.

각각의 요소에 대해서 레이어별로 어떻게 그릴지에 대해서 paint를 준비만 해놓는 것이다.

## 2. 브라우저 구성요소 중 렌더링 엔진의 역할을 짧게 설명해주세요.

웹서버로부터 응답받은 자원을 웹 브라우저 상에 나타내는 역할을 한다.

브라우저는 서버로부터 HTML 문서를 응답받으면 렌더링 엔진의 HTML 파서와 CSS 파서에 의해 파싱되어 돔,css돔 트리로 변환되고 렌더 트리로 결합한다. 이렇게 생성된 렌더트리를 기반으로 브라우저가 웹페이지를 나타낸다.

렌더트리는 돔과 css돔을 합해 최종적으로 브라우저에 표기될 것 들만 렌더트리에 선별하여 생성하는 것이다. 이 렌더트리는 표시되는 각 요소의 레이아웃을 계산하는데 사용되고 픽셀을 화면에 렌더링 하는 페인트 프로세스에 대한 입력으로 처리가 된다.

즉, 렌더트리는 화면에 보이는 모든 노드와 콘텐츠 스타일 정보를 병합한 최종 출력에 해당된다.

브라우저마다의 렌더링 방식이 조금 다르다.

## GIT

해당 화면은 master로 happy\_new\_year 브랜치를 squash merge한 그래픽입니다.

처음 master의 파일은 happy.md 파일만 존재하였습니다.

각 happy\_new\_year의 브랜치의 커밋은 커밋메시지와 동일한 md파일을 생성하였습니다.

(ex ) 커밋메시지 : new -> 생성된 파일 : new.md

마스터 브랜치에 happy.md파일만 존재할시

그림과 같은 커밋메시지를 가진 happy\_new\_year 브랜치를 생성하여 master 브랜치에 squash merge 하고 싶을때 사용되는 명령어의 순서를 적어주십시오.

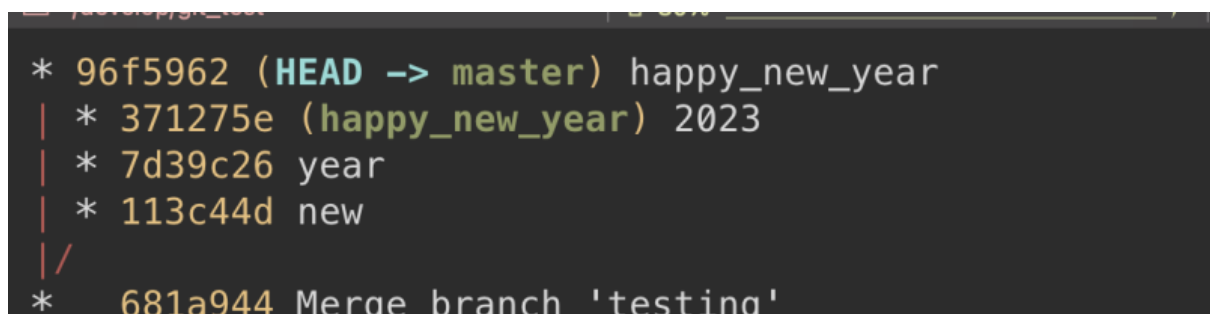
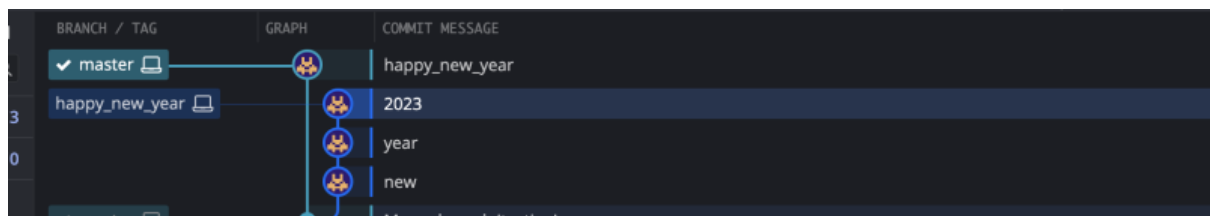
현재 브랜치는 master입니다.

(squash merge commit message는 happy\_new\_year 입니다. )

파일 생성 명령어

echo 내용 >> 파일

ex ) echo "cat" >> cat.md



git checkout -b happy\_new\_year

```
echo "new" >> new.md
git add new.md
git commit -m "new"
git checkout master
git merge --squash happy_new_year
git commit -m "happ_new_year"
```

---

## API

1) 아래의 두가지 URI을 보고 둘 중에 바람직한 URI는 어떤것인지 설명하고 다른 하나는 어떤 점을 고쳐야 하는지 URI 작성의 규칙을 인용하여 설명하세요

(1)

<http://www.itivillage.tistory.com/coffees> (O)

(2)

<http://www.itivillage.tistory.com/coffee>

도큐먼트(단수) - 디비의 row 단위 컬렉션에서 하나의 객체단위로 하는 단일 정보

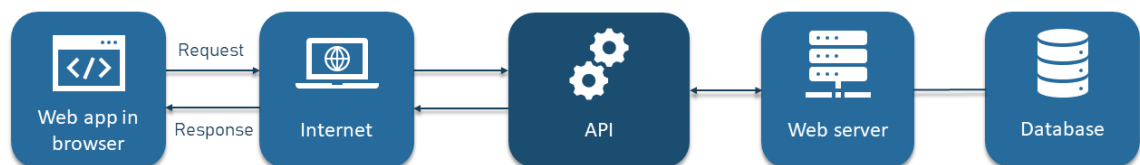
컬렉션(복수) - 도큐먼트들의 집합 / 관리주체는 서버, 어떤 리소스가 도큐먼트의 집합이며, 관리 주체가 서버인 것

스토어(복수) - 도큐먼트들의 집합 / 관리주체는 클라이언트

→ 컬렉션,스토어 구분할 필요는 크게 없다 관리주체의 차이일뿐

2) 다음의 그림을 보고 API의 작동 방식을 설명하세요

### HOW API WORKS



클라이언트가 정보를 받기 위해 request 요청을 날린다

api는 유효한 요청인지 확인한 뒤 웹서버를 호출한다

서버는 요청받은 데이터를 데이터베이스를 조회해 가져와 api에 응답한다

데이터는 api를 통해 클라이언트앱에 response 로 전달된다.

---