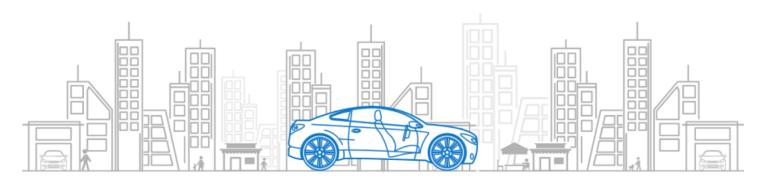


# Join Algorithm





- ❖ 3가지 Join Algorithm (Nested-loop, Merge, Hash)중 주어진 상황에 가장 적합한 Join함수를 구현 및 적용하여 output을 출력
  - 가장 적합하다는 의미는 open함수를 최소로 호출함을 의미합니다
  - 주어진 상황에 대해 <u>해당 Join이 왜 가장 적합한 Join Algorithm인지 자신의 논리를</u> wiki에 자세히 서술하여야 합니다

# ❖ 제약 조건

- 주어진 코드는 변형 불가능
- 주어진 class, function, variable, directory는 자유롭게 활용 가능
- int형 변수 이외의 type은 선언 및 사용 불가
- 각 문제에 대한 Join 결과는 하나의 파일로 출력
  - case1 → output1.csv
  - case2 → output2.csv
  - case3 → output3.csv

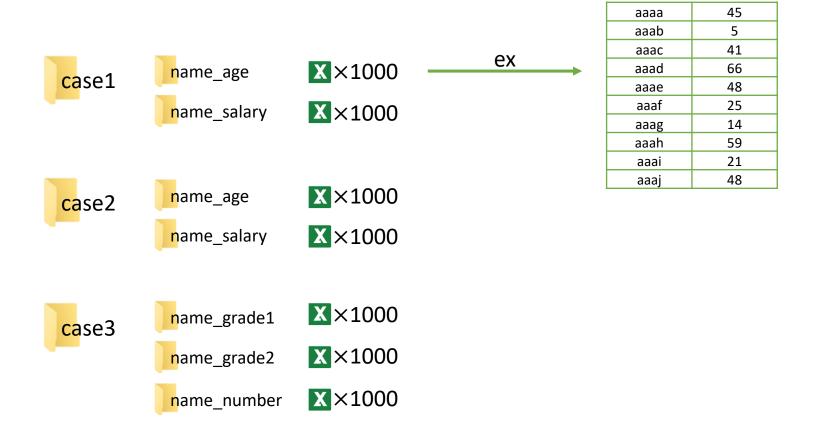


- 파일은 HYU LMS에서 다운받을 수 있으며 아래와 같은 구조 입니다
  - (Ass2)problem
    - buckets/
    - case1/
      - name\_age/
      - name\_salary/
      - test1.cpp
    - case2/
      - name\_age/
      - name\_salary/
      - test2.cpp
    - case3/
      - name\_grade1/
      - name\_grade2/
      - name\_number/
      - test3.cpp

0.csv



- 하나의 relation은 1000개의 파일에 나뉘어져 있습니다
- 하나의 파일에는 튜플이 10개씩 들어있습니다
- name은 unique한 attribute입니다





### Case 1

- 첫 번째 relation의 attribute는 name, age
- 두 번째 relation의 attribute는 name, salary
- 모든 tuple은 name을 기준으로 정렬되어 있음
- name을 기준으로 natural join 수행
- "(name),(age),(salary)" 형태로 output 파일에 출력

case1/name\_age/0.csv

аааа	45
aaab	5
aaac	41
aaad	66
aaae	48
aaaf	25
aaag	14
aaah	59
aaai	21
aaaj	48

case1/name\_age/1.csv

case1/name\_salary/0.csv

aaaa	5544
aaab	7404
aaac	1240
aaad	9065
aaae	9247
aaaf	2624
aaag	8413
aaah	4358
aaai	6320
aaaj	4647

case1/name salary/1.csv

#### case1/output1.csv

aaaa, 45, 5544 aaab, 5, 7404 aaac, 41, 1240 aaad, 66, 9065 aaae, 48, 9247 aaaf, 25, 2624 aaag, 14, 8413 aaah, 59, 4358 aaai, 21, 6320 aaaj, 48, 4647

:



# Case 2

- 첫 번째 relation의 attribute는 name, age
- 두 번째 relation의 attribute는 name, salary
- 첫 번째 relation의 tuple은 age를 기준으로 정렬되어 있음
- 두 번째 relation의 tuple은 salary를 기준으로 정렬되어 있음
- name을 기준으로 natural join 수행
- "(name),(age),(salary)" 형태로 output 파일에 출력



### Case 3

- 첫 번째 relation의 attribute는 학생 이름과 1학기 성적
  - student\_name, korean, math, english, science, social, history
- 두 번째 relation의 attribute는 학생 이름과 2학기 성적
  - student\_name, korean, math, english, science, social, history
- 세 번째 relation의 attribute는 student\_name, student\_number
- 모든 data는 무작위로 저장되어 있음
- 각 과목에 대해서 1학기 보다 2학기 때 성적 향상이 일어난 과목의 개수가 2개 이상인 학생의 이름과 학번을 "(student\_name),(student\_number)" 형태로 output 파일에 출력
  - 성적은 등급을 의미하므로 숫자가 작아진 것이 성적 향상을 의미

#### case3/output3.csv 형태 예시

igib,20224197

bcaa, 20227061

jeid, 20229140

egbc, 20226811

hgag, 20221869



### Code

- 각 문제에 대한 code는 문제번호.cpp 파일로 제출 ex. case1번 문제에 대한 풀이 코드 파일 이름은 test1.cpp
- Submission via HYU GitLab "git push"

# Output

- 모든 문제에 대한 출력 데이터는 문제번호.csv 파일로 제출 ex. case1번 문제에 대한 output 파일 이름은 output1.csv
- Submission via HYU GitLab "git push"
- (your git repo)
  - ass1/
  - proj1/
  - proj2/
  - ass2/
    - test1.cpp
    - output1.csv
    - test2.cpp
    - output2.csv
    - test3.cpp
    - output3.csv



## Report(Wiki)

- 각 상황에 대해 <u>해당 Join Algorithm이 왜 가장 적합한지 자신의 논리를 자세히 서술</u>
- 이론수업을 토대로 실제 데이터를 바탕으로 한 구현과정에서의 Trouble Shooting
- hwp / ppt / word / pdf / your git repo wiki page
  - notion에서 .md 파일 추출 시 이미지가 깨지므로 pdf로 추출하여 제출하세요
  - gitlab의 wiki 페이지에 wiki 작성 시, LMS에는 git repo wiki 페이지 주소 링크만 올려주시면 됩니다

#### Submission via HYU LMS

### Deadline

• 추가 제출 기한은 없습니다

~12/17 11:59 pm