Introduction to Computer HW3 Report

B06602037 徐子程

How to run my code ?

Problem A.

```
$ make remainder
$ ./a.out k N m
```

or `$ ./remainder k N m`

Problem B.

```
$ make admission
$ ./a.out student.csv dept.csv output.txt
```

or `$ ./admission student.csv dept.csv output.txt`

Clean

```
$ make clean
```

Problem A.

We adapt Modular exponentiation algorithm .

Based on the property:

$$A^2 \bmod C = (A * A) \bmod C = ((A \bmod C) * (A \bmod C)) \bmod C$$

Pseudo code:

```
function modular_exponent(base, exponent, m)
result = 1
while (exponent > 0):
      if (exponent & 1):
            result = (result * base) mod m
      exponent /= 2
      base = (base ** 2) mod m
return result
```

Problem B.

A. 演算法設計

將所有學生儲存在容器裡，每次取一個學生，按志願序選擇的科系，將此學生設定為已錄取，並讓該科系紀錄此學生的分數及 ID。直到所有學生都被錄取（或是已經用完志願）。
當一個科系人數已經額滿，則將目前該科系錄取最低分數的學生刪除，並設定為未錄取。

Note:

在刪除學生時，只要知道分數最低的候選人即可，故採用 min heap 儲存，而不先進行排序。

B. 資料結構及物件設計

1. 分發系統(admissionSys)
   `vector` 儲存學生/科系的 pointer，index 為學生/科系的編號(ID)
   hash table(`unordered_set`)儲存尚未被分發學生的 ID

2. 學生(student)
   三個分數(g1,g2,g3)
   志願序(1~10)
   是否被分發(bool)
   被分發到的科系 ID
   目前的志願序(在這個 index 之前的科系不錄取)

3. 科系(department)
   錄取人數上限
   分數比重
   min heap(`priority_queue`)儲存已經被錄取學生的分數及 ID

Pseudo code:

```
function admissionSystem()
      vector  student_list
      vector  dept_list
      unordered_set unadmitted

      for id in student_list:
            // insert all the students's ID into unadmitted
            unadmitted.insert(id)

      while (!unadmitted.empty):
            id =  unadmitted.begin
            stu = student_list[id] // get current student

            // assume the student has been admitted first
            unadmitted.erase(id) // remove from unadmitted students
            dept = dept_list[stu→choice[currChoice]]
            stu→set_admitted()
            stu→set_dept(dept→id)

            // add the student into the department's admitted list
            dept→pq.push(score(stu), stu→id)

            // the department is full
            if (dept→full())
                  // get the student who has minimum score than others
                  stu =  student_list[dept→pq.top]
                  // set the student back to unadmitted
                  stu→set_unadmitted()
                  dept→pq.pop()
                  unadmitted→insert(stu→id)
```