

21000549-001/002 알고리즘

Programming Assignment #3

Due: 2020년 11월 8일 23:59PM

단어들 중 일부들만을 이용해서 문장 만들기

1. Mission Objective

우선순위를 나타내는 음이 아닌 정수와 단어로 구성되어 있는 순서쌍 n 개가 주어져 있다. 그 순서쌍을 $A_0, A_1, A_2, \dots, A_{n-1}$ 이라고 하자. $A_i = (p_i, w_i)$ 라고 할 때, p_i 가 우선순위, w_i 가 단어이다. $w_0, w_1, w_2, \dots, w_{n-1}$ 을 다음 조건들을 만족하면서 출력하는 프로그램을 만든다 (제한 조건: $p_i < 2^{20}, n < 10^6$):

- 1) 각 단어들은 한 칸씩 띄어서 출력한다.
- 2) $p_i < p_j$ 이면 w_i 가 w_j 보다 먼저 출력된다.
- 3) $p_i = p_j, i < j$ 이면 w_i 가 w_j 보다 먼저 출력된다.
- 4) $i < \left\lfloor \frac{n}{2} \right\rfloor$ 이면 w_i 대신 하이픈 - 가 출력된다.

예를 들어, 다음과 같이 15개의 순서쌍이 있다고 하자:

(11, Hello), (10, World), (9, interesting), (0, algorithm), (0, Sookmyung),
(4, Women's), (1, University), (8, a), (2, a), (1, I), (1, am), (8, boy), (6, are),
(4, you), (2, girl)

이를 위의 규칙대로 출력하면 다음과 같다:

- - - I am a girl - you are a boy - - -

2. 파일 및 함수 설명

여러분의 리눅스 계정에 **PA3** 디렉토리가 있고, 그 안에 다음과 같은 파일들이 있을 것이다.

pa3.cpp: **main** 함수가 있는 소스코드 파일로, **concat.cpp**에 있는 소스코드를 실행할 때 사용하면 된다. 이 파일의 내용은 점수에 아무런 영향을 미치지 않으므로 마음껏 편집해도 된다 (심지어 컴파일이 되지 않아도 됨)

pri.h, pri.cpp: 음이 아닌 정수와 단어와의 순서쌍을 저장하기 위한 클래스. 각 멤버 변수 및 멤버 함수의 의미는 다음과 같다.

- **pr**: 우선순위를 나타내는 음이 아닌 정수값
- ***word**: 단어
- **pri()**, **pri(int p, char* w)**: 생성자
- **~pri()**: 소멸자
- **operator= (pri &pair)**: A와 B가 **pri** 타입의 변수일 때, **A = B**;가 등장했을 때 실행되는 함수.
- **setPair(int p, char* w)**: **pr**과 **word**를 정해주는 함수.
- **printPair(void)**: 순서쌍을 화면에 출력해주는 함수.

concat.h: 단어들을 연결하여 문장을 만들어내기 위한 클래스가 담긴 헤더파일로, 각 멤버 변수 및 멤버 함수의 의미는 다음과 같다.

- ****pa**: 순서쌍들의 배열
- **num**: 순서쌍의 개수
- **concat(void)**, **concat(int* priority, char **words, int n)**: 생성자. 이 때, 후자의 경우, 우선순위의 배열과 단어들의 배열이 들어가게 되며, 세 번째 인자는 배열의 원소의 개수이다.
- **~concat(void)**: 소멸자

- **concatenate(void)** : 단어들을 연결한 문장을 반환하는 함수로, 여러분이 작업해야 할 곳이다.

이외에도 멤버 변수나 멤버 함수가 더 필요하면 선언 및 정의해서 써도 된다.

concat.cpp : **concat.h**에서 선언된 메소드들을 정의해놓은 파일. 이 중에서 **char* concatenate(void)**의 내용을 편집하면 된다. 또한, **concat.h**에서 메소드를 추가로 선언했다면, **concat.cpp**에서 추가로 정의해줘야 한다. (생성자, 소멸자는 절대로 변경하면 안 됨)

score.o : 채점을 위한 파일. 수정이나 삭제 금지.

Makefile : **make** 명령을 위한 매크로 파일. **make**라는 명령을 내리며 **run**이라는 실행파일이 만들어진다. 이들을 지우고 싶으면 **make clean** 명령을 내려주면 된다.

3. 실시 예

make 명령 후, **pa3.cpp**를 수정하지 않았을 경우, 문장을 만들어내는 프로그램을 잘 만들었다면, 실행했을 때 다음과 같이 떠야 할 것이다:

```
- - - I am a girl - you are a boy - - -
```

4. 채점 방식

프로그래밍 후, 여러분이 점수를 예측하기 위해서는 **score.o** 파일을 활용하면 되며,

```
$ make score
```

이와 같이 컴파일하면 **score**라고 하는 실행파일이 만들어진다. 이는 **main.cpp**와는 무관하게 만들어지며, 실행하게 되면 여러분들이 받게 될 점수가 출력된다. **make clean** 명령을 내리면 이 파일 역시 지워진다.

채점은 1) 알고리즘의 정확성 평가 2) Stress test로 구성되어 있으며, 각각 3.2, 4.8점이다. 모든 테스트가 성공적으로 끝나면 8점을 보장받게 된다. 그 중 stress test에서는 순서쌍의 개수가 100만개가 될 때까지 테스트를 해 보며, 모든 테스트가 각각 1초 안에 끝나야 8점을 받게 된다.

채점 후, 001, 002반 전체 학생들 중에서 8점 받은 학생들끼리 수행 시간을 비교해서 빠른 순서대로 최종적으로 10점부터 8점까지 부여할 예정이다.

5. 제출

- 11월 8일 일요일 23:59까지. 시계는 제출용 서버의 시계를 따른다.

- 여러분의 계정에 **PA3** 디렉토리가 있고, 이 안에 다음과 같은 파일들이 있을 것이다:

```
pri.cpp
pri.h
concat.cpp
concat.h
pa3.cpp
score.o
Makefile
```

이 중, **concat.h** 파일과 **concat.cpp** 파일의 최종 수정 시각으로 delay 여부를 판별할 것임.

- 만일 위의 파일들 중 어느 하나를 지웠으면 담당 교수에게 문의할 것.
- 하루 Delay 되면 본인 획득 점수의 반을 받게 되며, 이를 이상 Delay 되면 0점.
- Copy는 해당 PA 0점. 문제 해결 아이디어를 내기 위해서는 어떤 것을 참조해도 상관 없지만, 소스코드를 찔 때는 다른 사람 것이나 웹사이트를 보면 안 됨. 반드시 본인이 프로그램을 설계하고 타이핑 하여 만들어야 함.