Assignment 2: Photo Slideshow[1]

Due date: *Wednesday, April 17, 2019 at 22:00*

This is an individual assignment. You may discuss it with others, but your code and documentation must be written on your own. You might receive a bonus for an outstanding solution.

Problem Statement

In this assignment, you are asked to implement a library to determine how interesting a slideshow is. More precisely, given a list of photos and the tags associated with each photo, your implementation must assign a score to a given slideshow formed from those photos.

We define a photo by a set of tags and its orientation. The orientation is either horizontal or vertical. A slideshow is a sequence of slides. Each slide consists of either:

- A single horizontal photo, or
- two vertical photos side-by-side.

Same as the photos, we describe a slide by a set of tags. If the slide contains only a single horizontal photo, the tags of the slide are the same as the tags of the single photo it contains. However, if the slide contains two vertical photos, the tags of the slide are the union of the tags of the photos it contains. Each photo can be used either once or not at all. A slideshow containing duplicated photo is defined not to be interesting at all.

Scoring

The slideshow is scored based on how interesting the transitions between adjacent (neighboring) slides are. In more precise terms, the score of a slideshow is the cumulative value (sum) of the *interest factor* between every two neighboring slides within the slideshow. Considering the tags of two neighboring slides, one can come up with many interesting ways to score a slideshow. In your library, you must implement both a default scoring scheme and also a more flexible one for further changes. Below, you will find a description of the two scoring schemes:

- **Default scheme**: For two subsequent slides S_i and S_{i+1}, the interest factor is the minimum of the number of common tags (intersection) and the number of different tags (symmetric difference) between S_i and S_{i+1}.
- **Flexible scheme**: In this scheme, the user of your library is supposed to come up with his or her own interest factor. To this end, the user passes a function that your library must use to calculate the final score for the given slideshow.

[1] Text and the original idea are partially barrowed from the Google Hash Code competition 2019; https://codingcompetitions.withgoogle.com/hashcode/

Input data set

The set of the photos must be read from a file. The name of this file is given as a parameter to the constructor of your library. The file contains exclusively ASCII characters with lines terminated with a single end-of-line character. The first line of the file contains a single integer N ($1 < N < 10^5$) that indicates the number of photos in the collection. This is followed by N lines, where line i contains a description of the photo with ID i. The description of a photo contains the following data, separated by a single space.

- 'H' or 'V', indicating whether the photo is horizontal or vertical.
- An integer M ($0 < M < 100$) indicating the number of tags for the photo.
- M text strings separated by spaces representing the tags.

The library must define (i.e., implement) all the declarations in the following header file:

slideshow.h

```
#ifndef SLIDESHOW_H_INCLUDED
#define SLIDESHOW_H_INCLUDED

struct photoset;

/* Constructor: returns the null pointer in case of failure.
* Reads the set of photo from the given file.
*/
struct photoset * ps_new(char * filename);

/* Destructor: clear all memory allocated for the given photoset. */
void ps_delete(struct photoset * p);

/* Calculates the score of the given slideshow.
* The slideshow is a string representation of the
* photo IDs concatenated with ','. The interest factor
* between two adjacent slides is the minimum between two
* numbers: number of common tags and number of
* different tags. Return -1 if slideshow contains
* at least a duplicated or a nonpaired vertical photo.
*/
int ps_score_default(struct photoset * p, const char * slideshow);

/* Calculates the score of the given slideshow.
* The slideshow is a string representation of the
* photo IDs concatenated with ','. The interest factor
* between every two adjacent slides are calculated by
* the given input function. It returns -1 if
* slideshow contains at least a duplicated photo or a
* nonpaired vertical photo.
*/
int ps_score(struct photoset * p, const char * slideshow, int
    (*interest_factor)(unsigned intersection, unsigned differences));

#endif
```

Example

Below is an example of a photo database file with $N = 4$ photos.

photoset1.in

```
4
H 3 cat beach sun
V 2 selfie smile
V 2 garden selfie
H 2 garden cat
```

Next, in `usercode.c` you see how your library is used in order to read the DB file and assign a score to arbitrary slideshows.

usercode.c

```c
#include "slideshow.h"

int my_method(unsigned i, unsigned j){
  return (i-j) * (i-j);
}

int main(){
  struct photoset* p = ps_new("photoset1.in");

  int s1 = ps_score_default(p, "0,1,2");         // s1 == 0
  int s2 = ps_score_default(p, "0,3");           // s2 == 1

  int s3 = ps_score(p, "0,1,2", my_method);      // s3 == 36
  int s4 = ps_score(p, "2,3", my_method);        // s4 == -1
}
```

Submission Instructions

Add comments to your code to explain sections of the code that might not be clear. You may use an integrated development environment (IDE) of your choice. However, *do not submit any IDE-specific file,* such as project description files. Make sure to only submit the source **slideshow.c** files and not anything else and submit that file through the iCorsi system.