1st Python (Hard) Week — EPITA 2019 — — MSc CS — Wednesday : Google Hash Code – GHC —

EPITA (MSc) Team

April 10, 2019

GHC: Google Hash Code

GHC: Google Hash Code

- Challenge organized every year by Google
- Teams from all around the world
- High computation problems
- Objective: maximize a given function to obtain the best score.
- 4h of competition.

GHC: Objective

Slideshow of pictures

- The goal of this challenge is to arrange a list of photos into a slideshow that is the most "interesting"
- The interest of a slideshow is represented by a score
- Objective: get the highest score.

GHC: Subject and Data file

Slideshow of pictures

You can download all the files you need at:
https://codingcompetitions.withgoogle.com/hashcode/archive

Please look at **2019 Qualification Round**.

GHC: Photos

Photos

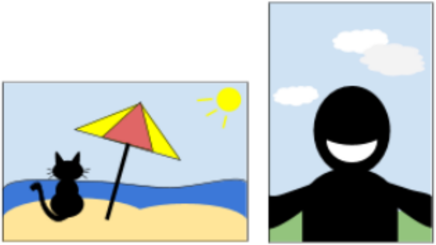
- Two types of photos: Horizontal ('H'), Vertical ('V')
- A photo is represented by a list of tags
- No duplicates of tags on the same photo
- Each photo can be used either once or not at all.

Example

For example, a photo with a cat on a beach, during a sunny afternoon could be tagged with the following tags: [cat, beach, sun].

GHC: Example of photos

Example of photos



the photo on the left is horizontal, while the photo on the right is vertical

GHC: Slides

Slides composition

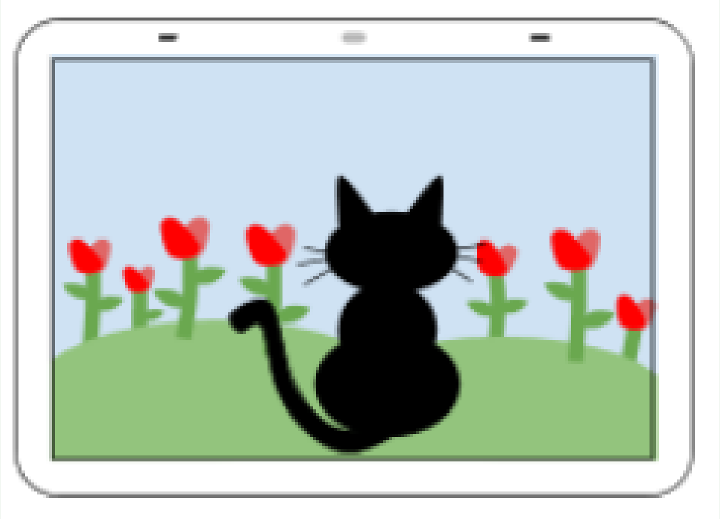
- A slide has one horizontal photo **or** two vertical photos
- If the slide contains a single horizontal photo, the tags of the slide are the same as the tags of the single photo it contains
- If the slide contains two vertical photos, the tags of the slide are all the tags present in any or both of the two photos it contains
- No duplicates of tags in a slide.

Example

- For example, a slide containing a single horizontal photo with tags [cat, beach, sun], has tags [cat, beach, sun].
- For example, a slide containing two vertical photos with tags [selfie, smile] for the first photo and tags [garden, selfie] for the second photo, has tags [selfie, smile, garden].

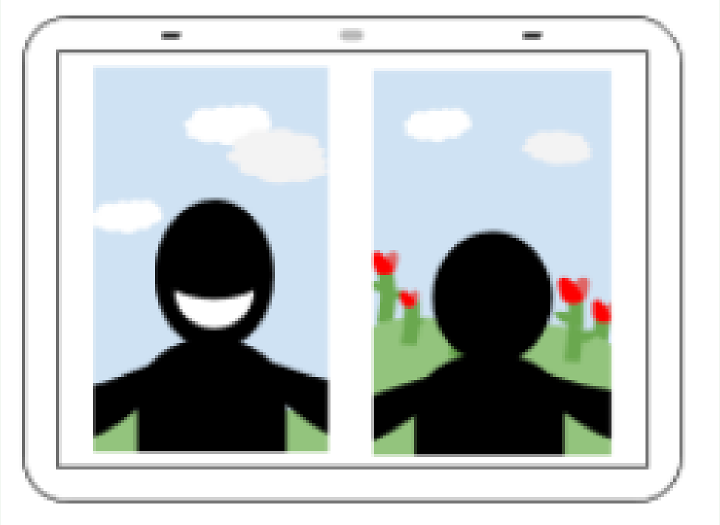
Example of Horizontal slide

Example of a slide with an horizontal photo



Example of Vertical slide

Example of a slide with two vertical photos



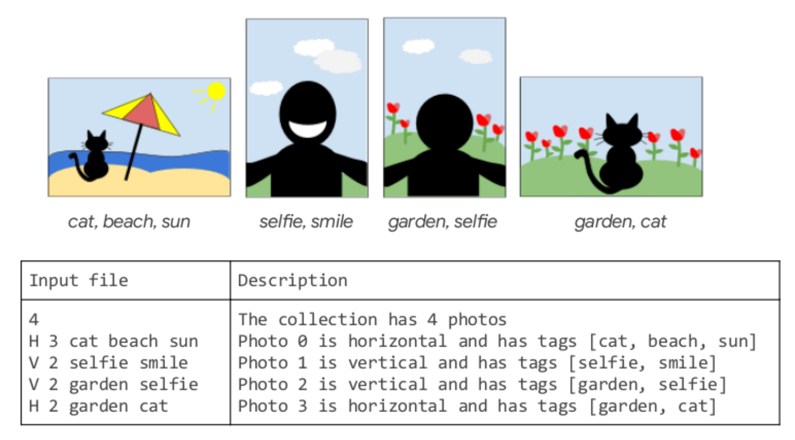
GHC: Datas

File format

- Each input data set is provided in a plain text file containing exclusively ASCII characters with lines terminated with a single '\n' character (UNIX- style line endings)
- The first line of the data set contains a single integer N ($1 \leq N \leq 10^5$): 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 ($0 \leq i \leq N$)
- The description of the photo i contains the following data, separated by a single space:
  - A single character H if the photo is horizontal, or V if it is vertical
  - An integer M_i ($1 \leq M_i \leq 100$): the number of tags for that photo
  - M_i text strings: the tags for photo i (between 1 and 10 of lowercase ASCII letters and digits for each tag)

GHC: Example of photos

Example of photos

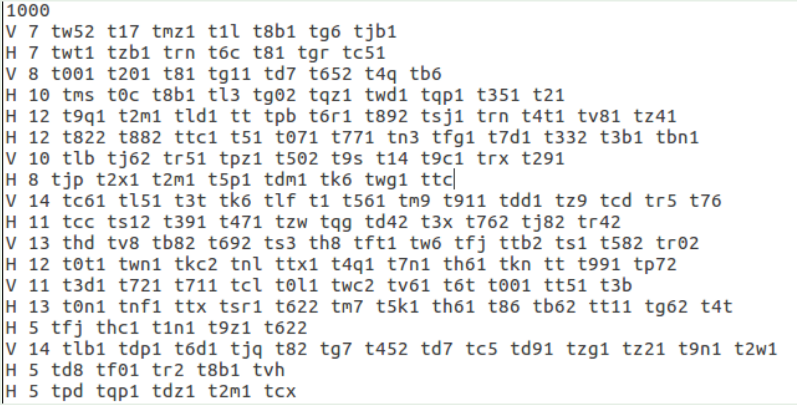


cat, beach, sun *selfie, smile* *garden, selfie* *garden, cat*

Input file	Description
4	The collection has 4 photos
H 3 cat beach sun	Photo 0 is horizontal and has tags [cat, beach, sun]
V 2 selfie smile	Photo 1 is vertical and has tags [selfie, smile]
V 2 garden selfie	Photo 2 is vertical and has tags [garden, selfie]
H 2 garden cat	Photo 3 is horizontal and has tags [garden, cat]

GHC: How photos looks like in the data

How a photo looks like in the data



```
1000
V 7 tw52 t17 tmz1 t1l t8b1 tg6 tjb1
H 7 twt1 tzb1 trn t6c t81 tgr tc51
V 8 t001 t201 t81 tg11 td7 t652 t4q tb6
H 10 tms t0c t8b1 tl3 tg02 tqz1 twd1 tqp1 t351 t21
H 12 t9q1 t2m1 tld1 tt tpb t6r1 t892 tsj1 trn t4t1 tv81 tz41
H 12 t822 t882 ttc1 t51 t071 t771 tn3 tfg1 t7d1 t332 t3b1 tbn1
V 10 tlb tj62 tr51 tpz1 t502 t9s t14 t9c1 trx t291
H 8 tjp t2x1 t2m1 t5p1 tdm1 tk6 twg1 ttc
V 14 tc61 tl51 t3t tk6 tlf t1 t561 tm9 t911 tdd1 tz9 tcd tr5 t76
H 11 tcc ts12 t391 t471 tzw tqg td42 t3x t762 tj82 tr42
V 13 thd tv8 tb82 t692 ts3 th8 tft1 tw6 tfj ttb2 ts1 t582 tr02
H 12 t0t1 twn1 tkc2 tnl ttx1 t4q1 t7n1 th61 tkn tt t991 tp72
V 11 t3d1 t721 t711 tcl t0l1 twc2 tv61 t6t t001 tt51 t3b
H 13 t0n1 tnf1 ttx tsr1 t622 tm7 t5k1 th61 t86 tb62 tt11 tg62 t4t
H 5 tfj thc1 t1n1 t9z1 t622
V 14 tlb1 tdp1 t6d1 tjq t82 tg7 t452 td7 tc5 td91 tzg1 tz21 t9n1 t2w1
H 5 td8 tf01 tr2 t8b1 tvh
H 5 tpd tqp1 tdz1 t2m1 tcx
```

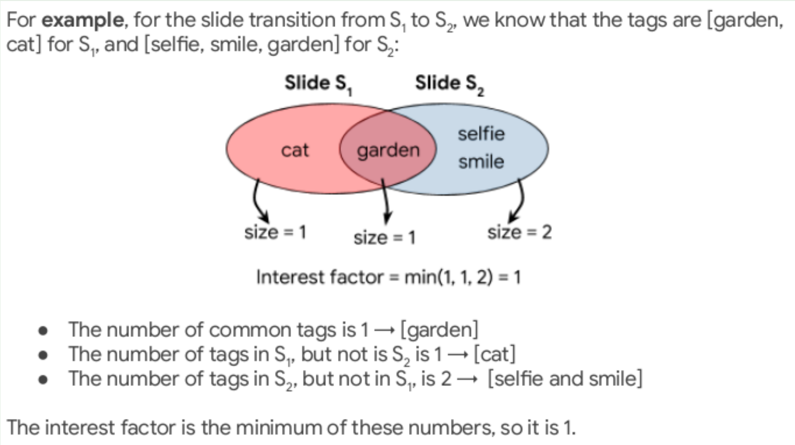
GHC: Scoring

Scoring function

- The slideshow is scored based on how interesting the **transitions** between each pair of subsequent slides are
- For two subsequent slides S_i and S_{i+1}, the **interest factor** is the minimum between the three values of:
  1. the number of common tags between S_i and S_{i+1}
  2. the number of tags in S_i but not in S_{i+1}
  3. the number of tags in S_{i+1} but not in S_i
- For a slideshow of S slides, the score will be equal to the sum of interest factors of each transition of two neighboring slides
- A slideshow with only one slide has a score of zero.

GHC: Example of scoring

Example of scoring



For **example**, for the slide transition from S_1 to S_2, we know that the tags are [garden, cat] for S_1, and [selfie, smile, garden] for S_2:

Slide S_1
Slide S_2
cat
garden
selfie
smile
size = 1
size = 1
size = 2

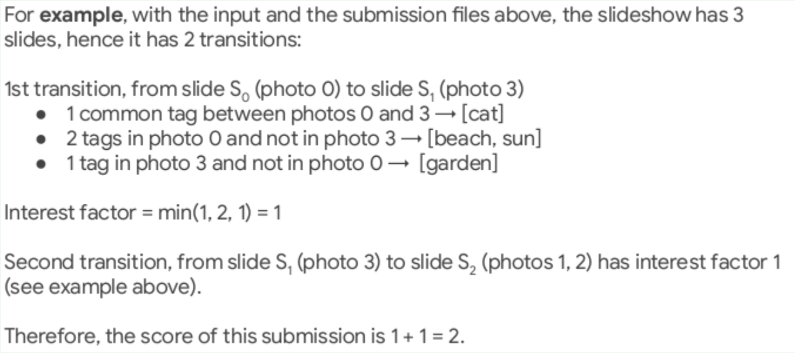
Interest factor = min(1, 1, 2) = 1

- The number of common tags is 1 ⟶ [garden]
- The number of tags in S_1, but not is S_2 is 1 ⟶ [cat]
- The number of tags in S_2, but not in S_1, is 2 ⟶ [selfie and smile]

The interest factor is the minimum of these numbers, so it is 1.

GHC: Example of scoring

Example of scoring



For **example**, with the input and the submission files above, the slideshow has 3 slides, hence it has 2 transitions:

1st transition, from slide S_0 (photo 0) to slide S_1 (photo 3)

- 1 common tag between photos 0 and 3 ⟶ [cat]
- 2 tags in photo 0 and not in photo 3 ⟶ [beach, sun]
- 1 tag in photo 3 and not in photo 0 ⟶ [garden]

Interest factor = min(1, 2, 1) = 1

Second transition, from slide S_1 (photo 3) to slide S_2 (photos 1, 2) has interest factor 1 (see example above).

Therefore, the score of this submission is 1 + 1 = 2.

GHC: Work to do ?

Work to do ? Well . . .

1. Write a (light, simple) code that reads the input file (all data files have the same structure)
2. Create slides from the data
3. Create a slideshow from the slides
4. Code the scoring function
5. Optimize the slides and their order to get a better score.
6. Save the result!

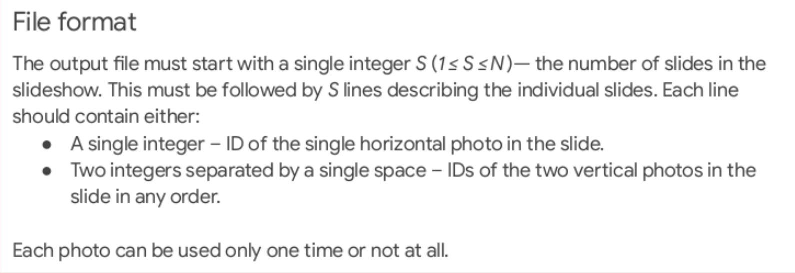
GHC: Rendering of work

Rendering of work

- Send **one** .zip file to PythonMscWeek_wednesday@protonmail.com for each group, containing 6 files
- One file for each slideshows (one for each data file)
- A 6th file must contain all the code you made to get the slideshows
- Don't forget to give the number of the group and all the students names!

Formating

Formating asked for the rendering



File format

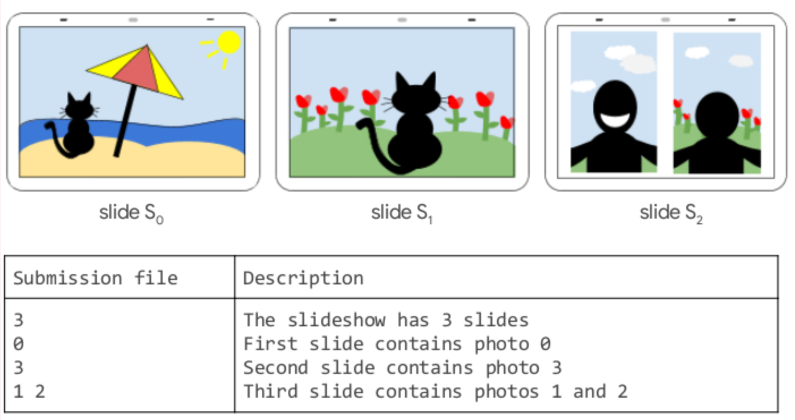
The output file must start with a single integer S ($1 \leq S \leq N$)— the number of slides in the slideshow. This must be followed by S lines describing the individual slides. Each line should contain either:

- A single integer – ID of the single horizontal photo in the slide.
- Two integers separated by a single space – IDs of the two vertical photos in the slide in any order.

Each photo can be used only one time or not at all.

Formating: Example

Formating: Example



slide S_0 slide S_1 slide S_2

Submission file	Description

3
0
3
1 2
	The slideshow has 3 slides
	First slide contains photo 0
	Second slide contains photo 3
	Third slide contains photos 1 and 2

GHC: Tips and Tricks

Beware!

- Photos are not really pictures, just a list of tags
- Tags are not always real words, most of them are just alphanumerical characters without meaning