Bookmarks

- Overview
- Entrance Survey
- Week 1
- Week 2
- Week 3
- Week 4
- Quiz
- Week 5
- Week 6
- Week 7

Lecture 13 - Trees - Time 51:54
Lecture Sequence

Wrap up - Time 33:39

Problem Set 7
Problem Set due Aug 04, 2016 at 23:30 UTC

- Sandbox

Week 7 > Problem Set 7 > Part 3 - Connecting Adopters and Adoption Centers

Bookmark

Part 3 - Connecting Adopters and Adoption Centers

(20 points possible)
Now that you have implemented both the AdoptionCenter and the different types of Adopters, it is time to try to adopt out some pets!

We will deal with two scenarios, one from the perspective of the `Adopter` type, and one from the perspective of an `AdoptionCenter`.

Help an Adopter visit AdoptionCenters in the Best Order

An Adopter or Adopter Subclass has a list of AdoptionCenters in the area, as well as information on what animals each AdoptionCenter has that day. Write a method that will return an organized list of the AdoptionCenters in such a way that the scores unique to the Adopter or Adopter Subclass for the AdoptionCenter will be ordered from highest score to lowest score.

Write the method `get_ordered_adoption_center_list(adopter, list_of_adoption_centers)` with the following parameters:

- `adopter` - A single Adopter or Adopter Subclass instance
- `list_of_adoption_centers` - A list of AdoptionCenter instances.

The method returns a list of an organized `adoption_center` such that the scores for each `AdoptionCenter` to the `Adopter` will be ordered from highest score to lowest score. In case of ties, order the adoption center names alphabetically.

Help an AdoptionCenter Select Adopters

Using the methods that you have been given, you want to help organize a list of Adopter types for an AdoptionCenter to send advertisements which will invite them to visit the AdoptionCenter. The AdoptionCenters may have limited funds and can only send out mail to a select few Adopters in their database, so want to select the best candidates to advertise to in order increase the odds of adoption.

Your task is to write a method `get_adopters_for_advertisement(adoption_center, list_of_adopters, n)`. The method should return a list of length up to n that represents the highest scoring Adopters/Adopter Subclasses for the specific AdoptionCenter (You want to find the top n best Adopter matches). Write the method `get_adopters_for_advertisement(adoption_center, list_of_adopters, n)` with the following parameters:

- `adoption_center` - A single `AdoptionCenter` instance
- `list_of_adopters` - A list of `Adopter` (or a subclass of `Adopter`) instances.
- `n` - The number of adopters, up to a maximum of n, who will be sent advertisements. Note that n >= 0 and may be longer than the `list_of_adopters`, in which case less than n advertisements will be sent out.

The function returns a list of the top `n` scoring Adopters from `list_of_adopters` (in numerical order of score). In case of ties, order the Adopter names alphabetically.

Advanced hints on sorting!

Some test cases

Paste both functions `get_ordered_adoption_center_list` and `get_adopters_for_advertisement` in the box.

Unanswered

You have used 0 of 30 submissions