CSC111 Project Proposal: Look and Cook

Dana Al Shekerchi, Nehchal Kalsi, Kathy Lee, and Audrey Yoshino

Tuesday, March 16, 2021

Problem Description and Research Question

In response to the pandemic's start last spring, many people resorted to exploring hobbies and interests to occupy their abundance of free time. As staying at home was mandatory, people alternated rooms within their homes rather than touring the city. One's second favourite place after their bed was the kitchen. Therefore, people started to try out new recipes to put their cooking skills to the test. Cooking allowed one to fulfill their boredom, improve their skills, and maybe get a good meal out of it.

The endless number of complicated recipes found online drove our motivation to create an application that endangers the misery of not having the right ingredients. Additionally, not all of us are ready to dedicate plenty of time to a meal; we are not even sure it will end up being tasty. Our application will allow the user to input the ingredients they crave or have, and the time they have to create their ideal meal. In doing so, users may step out of their culinary comfort zones by creating something new out of their everyday ingredients. Once the user inputs all the needed information, they would get all the recipes that meet their criteria.

Our project aims to conveniently allow one to create a meal using their desired ingredients and time, without leaving their house last minute and spending more time than wanted.

Dataset Description

There is one relevant dataset that we will be using for this project. This dataset is from Kaggle's "Recipe Ingredients and Reviews". There are four CSV files, but we plan to use only one file, viz., `clean_recipes.csv`. The primary separator of this file is a semicolon (;). This dataset has ten columns representing various recipe attributes to organize data and includes over 12200 rows constituting recipes. For computations, we will mainly depend on the columns `ingredients` and `total_time`. However, to present our findings, all the columns will be made use of.

Computational Plan

In this project, we envisage that trees would filter and search the recipes based on user-specified ingredients. The ingredients would be the nodes of the tree.

Based on the user's ingredients, the program would select the first ingredient and assign it as the root node. All the recipes containing only this ingredient (as in root node) will be added to a loop accumulator or a results list. The other recipes that do not contain the first ingredient will branch off and be utilized as the second recursive call input. Then, the program would pick up the second ingredient and test the conditions on the input data, and a similar process would be carried out. All recipes containing only the second ingredient and only the first and second ingredient would be appended to the results list this time. This process would continue until the recipes corresponding to the last ingredient have been added. It should be noted that every time a new ingredient is added, all recipes with valid combinations of the new and the previous ingredients will be appended to the loop accumulator. Also, it is obvious that ingredients that the viable recipes satisfy are found at the leaf node level. (Refer Fig 1)

Apart from this, if the user also specifies the total cooking time, in the first step, all recipes would be filtered based on that condition. And then, a process similar to the one explained above would follow.

While our dataset is available as a CSV file, to start, we will be required to perform a data transformation, obtaining material to work within Python. Simultaneously, data filtering will help limit the scope of information, extracting data that would be meaningful for our purposes. Once our dataset is suitable for use, we may use it to build trees. With this, we intend to form a tree of the searched ingredients, which will aid in finding recipes to fit the user's preferences. Another algorithm to configure is the sorting of search results, which should be prioritized according to the relevance to the user's search criteria. To make our program interactive, evidently, we will be required to design functions to build our user interface and link it with our tree structure.

We plan to use the library tkinter to display our graphical user interface. Users will be presented an input page with a list of potential ingredients divided into subcategories, i.e. pantry items, fresh produce, etcetera, accompanied by a check box. This list of possible ingredients will remain on the left half of the input page, and as users check off items, a list will appear on the right-hand side displaying items that have been checked off. We plan to provide users with an option to check off "household items," or, in other words, everyday household ingredients (e.g. salt, certain seasonings, oil, flour, etc.) so users do not have to input these ingredients at each usage repeatedly. There will be a button at the bottom of the list of checked off ingredients provided on the right that will display the user with a list of recipes containing *only* the provided ingredients, alongside an image of the recipe as well as details such as cook time, recipe name, and author. The further down the list, the fewer ingredients are used out of those provided. Once users click on the recipes, a separate window will open, providing the user with instructions, measurements, and further details on their chosen recipe.

Some tkinter modules and functions we plan to include are tkinter.scrolledtext (to display necessary lists of recipes and ingredients), the tkinter window manager (to control our display windows), the Button class (to be used at multiple stages in our program) and the Checkbutton class (to create check button widgets for users to input their available ingredients).

References

Burdurlu, Y. (2019, February 18). Recipe ingredients and reviews. Retrieved March 14, 2021, from `https://www.kaggle.com/kanaryayi/recipe-ingredients-and-reviews/activity`.

Tkinter - python interface to tcl/tk. (n.d.). Retrieved March 14, 2021, from `https://docs.python.org/3/library/tkinter.html`

Appendix

The Central Tree Structure of our Program

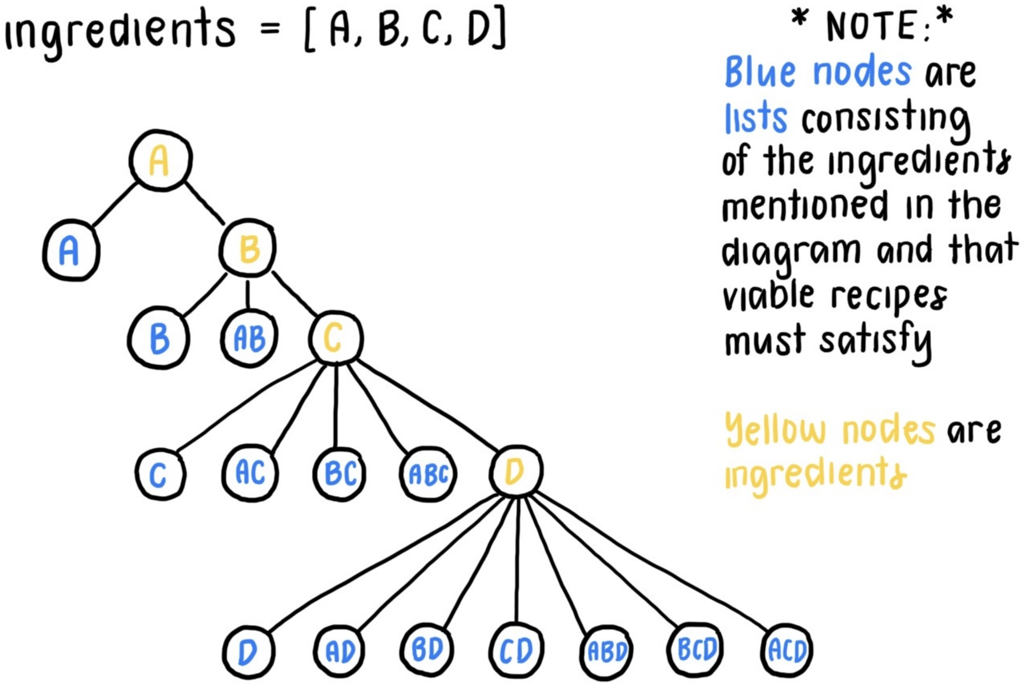
ingredients = [A, B, C, D]
A
A
B
B
AB
C
C
AC
BC
ABC
D
D
AD
BD
CD
ABD
BCD
ACD
* NOTE:*
Blue nodes are lists consisting of the ingredients mentioned in the diagram and that viable recipes must satisfy
Yellow nodes are ingredients

Figure 1: Tree structure

The **yellow** nodes of the tree represent individual ingredients. The **blue** nodes are a list of combinations of individual ingredients involving the current root of the subtree. The recipes containing *only* the specified ingredient combinations at the leaf level are added to the results.