DRAFT LAB

Draft day is one of the most important events of the year for professional sports teams. In addition to drafting individual players after months of evaluation, teams will also wheel and deal over the picks themselves. Your assignment is to build a program that will evaluate a given pick trade as successful (your team stands to either gain or maintain value), or as a mistake (your team stands to lose value).

Requirements:

The good news is that actionability is already baked into the concept of this lab. Your focus will be on using the provided data effectively, and clearly conveying your strategy. You will be submitting your trade evaluator program and a write-up (approximately 1 page) that explains the strategy that your program employs.

Provided Data:
playerDB.csv – A list of every qualified NBA player from 1980-2019
draftDB.csv – A list of every draft pick from 1949-2019

Write-Up:
Your write-up will comprise half of your grade on this assignment. It will be evaluated for clarity and presentation, as well as the overall soundness of your strategy. There are many reasonable paths to take in this assignment. Your goal is to convey yours clearly, pointing out relevant assumptions and limitations along the way.

Program:
Your trade evaluator program will comprise the other half of your grade. While there are many trades for which there is no universally "correct" answer, your program is expected to perform reasonably and reliably, and in a manner that is consistent with the strategy described in your write-up.

You have been provided with starter code in `draft.py` that performs the basic work of reading in the data files and messaging the user. You have also been provided with a bash script, `run_trade.sh`, that is currently set up to run `draft.py` on Halligan. I will be evaluating your program by automatically running `run_trade.sh` against different inputs on the Halligan servers. Thus, while you are welcome to alter the file configuration that `run_trade.sh` invokes, or any of the internal logic of your program, you may not alter any of the user-facing messaging.

To help you verify this, I have provided two test input files and their corresponding, ground truth outputs. Thus, you should be able to run:

- `./run_trade.sh < test_input1.in > test_input1.out`
- `diff test_input1.out test_input1.gt`
- `./run_trade.sh < test_input2.in > test_input2.out`
- `diff test_input2.out test_input2.gt`

Without seeing any output produced by those calls to `diff`.

Due Date: March 9th, 2021 at 9:00am

This assignment will be submitted in two parts:

- Write-up via Gradescope.
- Code via: **`provide comp152san draftlab run_trade.sh draft.py`**

You are required to provide **`run_trade.sh`**, which will be used to run your submission. You are welcome to provide files in addition to **`draft.py`**, or a substitute for **`draft.py`**. Simply list all of the necessary file(s) after **`run_trade.sh`** in the **`provide`** command.