Fantasy league legend

candidate number: 215959

Table of content

[1 Introduction 3](#_Toc71411520)

[2.Background research 5](#_Toc71411521)

[Top Trumps: NBA All Stars 5](#_Toc71411522)

[Fantasy basketball league 5](#_Toc71411523)

[NBA Fantasy 6](#_Toc71411524)

[Yahoo Fantasy basketball 6](#_Toc71411525)

[The team Elo rating system 6](#_Toc71411526)

[The formula of the Elo rating system 7](#_Toc71411527)

[Per(Player Efficiency Rating) 8](#_Toc71411528)

[3.Professional considerations 9](#_Toc71411529)

[4.Requirement analysis 11](#_Toc71411530)

[Functional requirements 11](#_Toc71411531)

[Non-functional requirements 11](#_Toc71411532)

[5.System design 12](#_Toc71411533)

[5.1 learning objectives 12](#_Toc71411534)

[Modules 12](#_Toc71411535)

[Data Selection 13](#_Toc71411536)

[Modeling team chemistry 13](#_Toc71411537)

[The function to calculate the strength of groups of players 15](#_Toc71411538)

[The function to calculate the team chemistry 16](#_Toc71411539)

[Data memo function 18](#_Toc71411540)

[Tkinter library 18](#_Toc71411541)

[UI design 19](#_Toc71411542)

[Game mode 20](#_Toc71411543)

[Game rule 20](#_Toc71411544)

[6.implementation 23](#_Toc71411545)

[Explore the suitable model for data structure 23](#_Toc71411546)

[The rating implement method change 25](#_Toc71411547)

[Fibonacci sequence recursion 26](#_Toc71411548)

[Memoization 27](#_Toc71411549)

[Implementation in a simple robot 29](#_Toc71411550)

[Implementation issue 29](#_Toc71411551)

[7.Testing 31](#_Toc71411552)

[Unit test 31](#_Toc71411553)

[Integration test 32](#_Toc71411554)

[System test 32](#_Toc71411555)

[8.Evaluation 33](#_Toc71411556)

[9.Conclusion 35](#_Toc71411557)

[Compare with existing work 36](#_Toc71411558)

[Limitation 36](#_Toc71411559)

[Pros 36](#_Toc71411560)

[Future work 37](#_Toc71411561)

[10.Appendix 38](#_Toc71411562)

[Reference 38](#_Toc71411563)

[Data Source 39](#_Toc71411564)

# 1 Introduction

As more and more sports move towards professionalism, sports analytics is gaining more and more attention, especially after the great success of baseball and basketball, and now terms such as "magic ball theory" (Feldman, 2021)and "penalty kicks" have gained considerable fame in both fan and AI circles. In the meantime, "fantasy leagues" have been created, the fantasy league tournament is a model in which players create their teams as professional basketball managers.

The goal of this report is to quantifying chemistry value between players by applying the chemistry value in a card game based on the fantasy league model. Two players will choose their respective players as professional basketball managers and the system will score the players individually and by the bonds between them, with the team with the highest score winning. Another objective is to clarify the difference between the scoring system and other scoring systems. The main difference is that the scoring system takes into account the overall relationship between the players and the impact of the level of understanding on the overall team score. This report will demonstrate in more detail the program requirements, practices, implementation, and evaluation of this project.

Team chemistry in NBA (Unknown, 2019)

Most teams have great surface strengths, but when they don't produce results, they will use the team chemistry as an excuse (Pina, 2020). But it is hard to explain the team chemistry and how it works.

"This is important. If you unite the team together, then it’s like a third of a year’s win. People are not grateful for it, and it doesn’t look like much, but if you have a The team of the year, that becomes one or two games. This will bring you to another round of the playoffs." (Pina, 2020). In detail, it could be concluded as two main part: (Khan, 2013)

A sense of community - when every player on the team finds their place, plays to their strengths, and is willing to contribute to their team.

The level of understanding - players understand each others’ strengths and weaknesses and can work well together e.g. when one player is free to shoot, his teammate would pass the ball to him in time.

In this project, the team chemistry will be considered as a value, and the team rating system will calculate this value. The value will be taken into account in the final team score.

This document describes the project in eight sections:

Research: This section details the preliminary research carried out before starting to design the solution. This includes an outline for some of the scoring systems already in place.

Professional considerations: this section includes how the project will take into account the public interest, professional competence and integrity, and accountability to relevant authorities.

Requirement analysis: this section outlines the requirements of the project to determine the size and focus of the anticipated final project. This includes the functional and non-functional requirements of the final system.

Design: This section describes the overall game and the design process for the rating system. It will also highlight the calculations used for the game, we will explore the chemistry value between players based on team Elo rating and player per data. And strength function is used to calculate the overall rating of the player selected. Furthermore, the selection and comparison of models for player data structures will also be discussed in this section. Finally, the creation of the user interface and the explanation of the rules of the game are explained in detail.

Implementation: this describes all the problems encountered during the implementation of the project. It outlines the general order in which the problems occurred and how they were dealt with.

Testing: the program is tested using unit tests, integration tests, and system tests.

Evaluation: It includes a comparison of this scoring system with other scoring systems and an assessment of the accuracy of the project using correlation. It also includes a self-reflection on the overall results of the project and highlights any other outstanding issues with the system.

Conclusion: this summarizes all the elements covered in this report. Also, it will point out the pros and limitation which compare with the existing work And discusses further work that could be done on this project in the future, reflecting on the outstanding requirements and explaining how they could be implemented.

# 2.Background research

This section will introduce other similar card games include Top Trumps, NBA Fantasy, and Yahoo Fantasy Basketball, and how they calculate their player’s scores. After this, the report will introduce some data relative to the rating system which this project will use.

## Top Trumps: NBA All Stars

Top Trumps: NBA All Stars is a virtual game of Top Trumps based on the National Basketball Association's Top Trumps package. The game begins with two players competing for control of the ball to determine which player will start the game first, and the first player will choose an option to compare to their opponent and will win or lose based on the number of points scored for the same option. Options include height, free throw percentage, scoring average, assist average, rebounding average, and the number of playoff appearances. The more times a player wins, the closer the player gets to the goal until one player scores a dunk and the game finish, what is more as this game is time-limited, so when time runs out the game end. The side that whose player close to the basket will win.



-figure1 The player against each other during the competition

In the graph, both of the player will compare their data of height which could determine who is much stronger.

## Fantasy basketball league

Fantasy basketball is a game similar to player card scoring, where players form their own league and select players, and the points earned by the selected players' facts determine the league win. The players in Fantasy Basketball are selected from real players in the National Basketball Association. Players choose the players they believe will earn the most points by using their knowledge of the players and analyzing their performance. Points are awarded based on the player's performance in the game, such as points, assists, etc.

### NBA Fantasy

The fundamentals of NBA Fantasy consist of selecting players from the NBA league with the main aim of creating your fantasy team. your fantasy team will earn points according to their on-field performance. It allows the player to select a coach, two centers of the team, four forwards and four guards, the player will against several teams in each match day, and the player could Players can decide which team players to play based on their opponent's lineup and tactics and can make substitutions on the bench midway through the game. Players can decide which team players to play based on their opponent's line-up and tactics and can make substitutions on the bench midway through the game. Players can also buy or sell players to improve their team's score.

This is how the relationship between the real player performance and their score in the NBA Fantasy:

Three-Point Field Goals: 3 points

Two Point Field Goals: 2 points

Free Throws Made: 1 point

Rebounds: 1.2 points

Assists: 1.5 points

Blocked Shots: 2 points

Steals: 2 points

Turnovers: -1 point

### Yahoo Fantasy basketball

Another of the better-known fantasy games, yahoo fantasy basketball has a similar main format to NBA Fantasy, where you count the players you choose to score points, based on their performance on the basketball court. Players need 13 players to form a standard yahoo fantasy team, with 3 players on the bench and 10 players active in the game. Those active players need to put in their efforts to help the player's Fantasy team win the total points for the week. Only the points accumulated by the active players will be taken into account. Players need to know and observe each player selected, even if they make adjustments and substitutions to the team so that each player can play to their value and earn a higher score.

This is the correlation between live performance and fantasy points.

Each rebound grabbed equals 1.2 points

Each assist is 1.5 points

Each steal is 3 points

Each blocked shot is 3 points

Each turnover is -1 point

## The team Elo rating system

Elo is a simple measure of the strength of teams based on game-by-game results; it is used to calculate the relative skill level of players in a zero-sum game. A player's Elo score is expressed as a number, which may vary depending on the outcome of the rated game played. At the end of each game, the winning player receives points from the number of points lost. The difference between the ratings of the winners and losers determines the total number of points gained or lost after the basketball game. If the player with the higher rating wins, the player with the lower rating will only gain some rating points. However, if the lower-rated player gets a losing victory, many rating points will be transferred. In the case of a tie, the lower-rated player will also gain some points from the higher-rated player. This means that the rating system is self-correcting. In the long run, players who are rated too low or too high should accordingly perform better or worse than the rating system expects and thus gain or lose rating points before the rating reflects their true strength.

## The formula of the Elo rating system

Function explain:

First of all, we default a sigmoid with a scale of 400. The second formula could be used to indicate the probability of player A wins the game. The third formula is used to demonstrate that the probability that player B will win as we assume A will win, E in the formula stands for Error. RA and RB stand for their rating.

In this program, Elo ratings depend only on the final score of each game and the location of the game (home-field advantage). They include both regular season and playoff games. And this project will use FiveThirtyEight (FiveThirtyEight, 2021) related team Elo statistics as a part of the information which will be used to evaluate a rating in the final team composition. Compared to the pure Elo rating system the FiveThirtyEight model has a partial adjustment for Elo. At the end of each season, the new Elo value for all teams becomes three-quarters of the end of last year's season plus one-quarter of the average. When comparing the probability of winning, the home team gets an extra 100 points for having a home advantage. As the home team wins about 60% of the time, this should probably be +70 (+100 means 64% of the time!). The adjustment for K depends on the margin of victory and the difference in Elo. The more points you win, the more K you get (there is also a small adjustment for team strength, FiveThirtyEight set the K to 20.



-Figure2. this graph is about Los Angeles Lakers team Elo from 1950 to now from FiveThirtyEight (FiveThirtyEight, 2021)

This graph shows the change in Elo scores for the Los Angeles Lakers team from 1950 to the present day, with the scores at each stage and the slope giving a good indication of the overall strength of the team at that time. To work out who would win and who would lose between the two teams, you first have to calculate the difference in points between them. To map this difference into probabilities, Elo is often modelled by the sigmoid function (logistic). Specifically, it is a function that maps (−∞,+∞)→[0,1], where σ(0)=0.5. Elo rating is often used by the sigmoid function (logistic) for statistics and research specifically, it is a function that maps the difference between Elo rating. The sigmoid function is used here to get a better look at the data comparison of a zero-sum game.

****

- Figure 3. The sigmoid function diagram used to distribute the Team Elo rating

As the graph shown, we could find out that the probability of win is 0.5 which 50%, and also the probability of lose is 0.5 which is fit in well with none-sum game setting.

## Per(Player Efficiency Rating)

The Player Efficiency Rating is a per-minute rating developed by John Hollinger. In his word. “The PER sums up all a player’s positive accomplishments, subtract the negative accomplishments, and returns a per-minute rating of a player’s performance.” It is used to measure the productivity of a player per minute. It adds up all the positive contributions a player makes to his team while subtracting the negative contributions in a system of statistical point values. It also adjusts for pace and playing time to make comparisons between players much easier. (Unknown, n.d.)

PER Benefit:

It will show the personal average ability especially the offense part due to the huge weight in the algorithm. The top players can usually be found at the top of the list, so this is good evidence of how the players relate to each other.

PER Limitation:

But it is irrational to create the team solely based on the PER, we cannot just choose five players who are in the top PER List. What is more, the question also pointed by John that the PER always showed that the person who is good at the offense part but not the defense part (Khan, 2013). In detail, there are not many reliable defensive statistics to input into the formula. We would all agree that steals and caps do not necessarily correlate with good defense. As a result, defensive specialists are at a disadvantage in this metric, and great two-way players may rank lower than purely offensive players. Furthermore, it will strongly influence by the number of gameplay of the player, for example, there are a lot of players who have a good PER with only a few gameplay.

# 3.Professional considerations

3.1. BCS Code of Conduct

This is the BCS Code]of Conduct (BCS the Chartered for IT, 2019) which relate to my project:

1.1 have due regard for public health, privacy, security, and wellbeing of others and the environment;

The game is a team rating game, it will store the data about the name for the team, the member of the player’s ‘fantasy team’, and the final rating for their team when they finished their selection. Also, the statistical data for the NBA players could be found through the internet.

1.2 have due regard for the legitimate rights of third parties;  
Due to the free charge of this game, there will have fewer obligations for the third party.

1.3  conduct your professional activities without discrimination on the grounds of sex, sexual orientation, marital status, nationality, color, race, ethnic origin, religion, age or disability, or of any other condition or requirement;

The game is designed for all age groups without any discrimination, the older and children all will be able to enjoy this game, In-game, the children should be supported by their teacher or parents, which could help them expand their team manager ability to create their team.

1.4  promote equal access to the benefits of IT and seek to promote the inclusion of all sectors in society wherever opportunities arise.  
This game is a free charge game which means it will promote the benefit of IT equally. It will not consider anything about economic background.

2.5. respect and value alternative viewpoints and seek, accept and offer honest criticisms of work;

I will show my biggest respect to the feedback I received and if there are any alternative viewpoint or any criticisms, I will consider and accept if reasonable

4.6 encourage and support fellow members in their professional development.  
If any of my fellows get any questions in their professional development, I will do my best to help and support them.

2.2. Ethical Issues (School of Engineering and informatics, The University of Sussex, 2020)

This project is a specific type of game, so it is necessary to consider ethical approval to make sure it will not damage social healthy. This project is only a rating game that does not include any feature about risk, for the participants will have a try on this rating game and could provide the feedback for this project if they want. Due to this game do not have an age limitation, this game will make sure there is nothing harmful to the health of minors. The 6 basic rules of the game and the rating of the user’s fantasy team will be shown in the interface, and the information about their fantasy team only be used after the participants sign the consent form.

# 4.Requirement analysis

The main use of the scoring system is to allow users to choose as well as match their 'fantasy league' and get feedback. This part will consider a rational rating system, in which the functional requirement and non-functional requirement will be outlined.

## Functional requirements

* The project shall be playable by two players on one computer
* The system should provide the user the order of pick
* Users will be able to choose the player from the random set
* Users will be able to see the information and image of the player
* The system should provide the user the order of pick
* The system should able to provide the user the team rating of the combined player
* The system should able to constrain the user to choose the same player
* The system should able to consider the team chemistry of chosen player rather like the existing games
* The program can combine players and be able to determine a combined rating
* The system will rate the team based on the team chemistry value
* The game system should determine the winner
* The game system should remind the user about the total cost and the rest of the cost after they chose an NBA player
* The game should show the 20 random player’s information clearly(include their name and their cost)

## Non-functional requirements

* The project will be developed using Python and Python Graphical User Interface
* The data should be reliable
* The User could go through the user interface easily
* The User could go through the Rule page
* The project will be compatible with a desktop device
* The system should rate the team fairly

# 5.System design

## 5.1 learning objectives

* Explore and figure out the most suitable model for players data structure
* Strength python knowledge
* Learn to build the User Interface by Python Tkinter library
* Explore how to calculate the value of the relationship and how the player relation will impact the final rating
* Come up with a reasonable function to rate the player group

The main learning objectives of this project were to learn and find the most suitable model for the player data structure and to build a team scoring system by finding the core of the relational ability values between players. As the final form of the project will be a game, it was necessary to find a suitable framework library. The last and most important part of the project is to find a suitable scoring method for the players. I found that memoization is the easiest and most efficient way to Implement and integrate the results rating of each team, so understanding how to use memoization became another learning objective. The last learning objective was to learn more about the use of the python library and to increase proficiency in python. The choice of programming language for this project was simple as python is simple and easy to understand with many libraries for data analysis and statistics.

### Modules



* data flow diagram

The picture above shows how the whole system works and how each part interacts with each other，As this project would eventually be presented as a game. Firstly the overall flow of the rating system, the player information, and the team Elo rating are the main data used to calculate the final team rating. Player rating and team Elo rating are used to calculate the team chemistry value, which is the main goal of the project. The resulting team chemistry values are stored in the Data memo, which provides some information to the Strength function.

The UI is indispensable as the project will eventually be presented as a game, the overall presentation of the UI is dependent on the rules of the game, the player will select 5 players to form a team through the UI and receive a score when each of the two players has selected a team the selection data will be sent to the Strength function. Then Strength function will calculate the final team scores from the data in the Data memo.

### Data Selection

The project required data that represented the strength of teams, the strength of players within those teams, and the chemistry between individual players. For the selection of data, initially, 14 notorious combinations were chosen from internet research, and team chemistry was approximated heuristically by determining if the players in question are in some sort of famous combination. Almost all these 14 combinations were two- or three-player teams, so the initial data size consisted of approximately 40 players. The rating of these players and teams is taken from the 2k rating, which is the website of providing players data in the game NBA 2k21.

I initially used the NBA2k21 data which is available on the 2k rating page which included both player and team data.

However, the website did not provide a function of how the ratings are evaluated, and I wanted a more reliable set of data whose distributions and calculating functions are known during the development of the chemistry function. I switched to use Team Elo since it was the most suitable choice. Team Elo is based on the team's era and the highest Elo score earned by the team in that year, while team Elo is obtained through FiveThirtyEight's History of the NBA.

Finally, when I try to calculate the chemistry values, it was found that it was not accurate to use only the team Elo and combination ratings, as the team Elo considers not only the stats of the famous players but also all the bench players and role players in the team. By expanding the player's stats to consider every player on every team selected, the formula is partially adjusted to find the chemistry value of these combinations by overall strength.

However, as the 2k rating page does not mention how player ratings are calculated and what the overall distribution of player ratings is, it was decided to use PER instead of Player Rating as the main data for calculating team chemistry value. All data for PER are taken from the basketball- reference website.

### Modeling team chemistry

This part is used to introduce the process of developing the model of team chemistry, the data used in this session are mentioned in the background research session.

The model for team chemistry has evolved through three stages.

Initially, the model only considered the relationship between individual team members.



- Figure4. the player chemistry relation display

As shown in the picture, it is the example of player chemistry in which the lines between each player represent their team chemistry, and the value on each line represents the level of chemistry between the players, the higher the value the higher the chemistry between the players, in other words, the higher the value the better the relationship between the players, which makes it easier for the players to produce good results. And this is where the main focus of this project will be. The details about the calculation will be mention in the later section.

The previous model was refined to also consider the relationship between groups of people or individuals



- Figure5. the relationship graph between groups

Unlike the first type, the graph is an example of player chemistry of the second type, which is an evolution of the first type that describes the chemistry value between groups and individuals, as expressed in the diagram, where player A and player B are a combination, and player C is associated with this combination in some way, and the addition of player C will potentially make the combination stronger

Because the relationship between large groups within an encompassing group will generate many chemistry values without adding extra information, the model was changed to evaluate the strength of each subset of the players.



- Figure6. the graph shows the final type

The final team chemistry design will show the relationship between groups. As the diagram above shows, player A is a group with player B, and player A is a group with player E. When group AB is combined with group AE, the effect this has on the group ABE is modeled in the value of the group ABE.

### The function to calculate the strength of groups of players

The function to calculate the strength of groups of players recurses on the strength of the subgroups of players within the group.



Figure7. The player rating implement by fib method

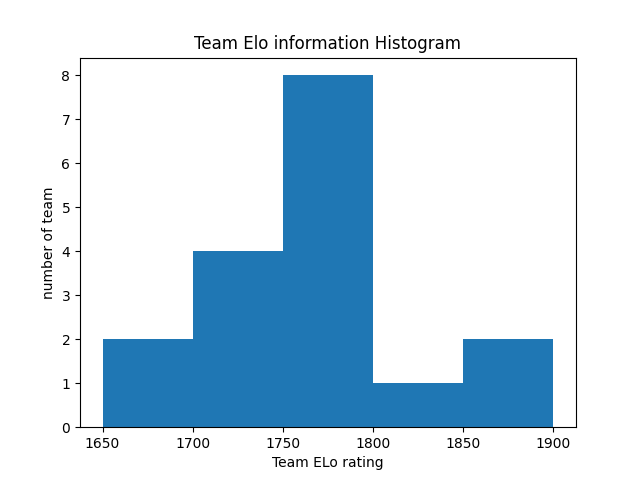
As the diagram above shows, the objective is to try to obtain the rating of a team of three players (Player1,2,3). To calculate the rating of this team, firstly we need to obtain the individual values of each player, and secondly, we need to take into account the possibility that each two of the three players will be combined, as shown in the second row of the diagram above. In simple terms, the strength of the second row is the strength (which is based on the PER rating of players)of the third row plus k, e.g. (player 1,2) = strength (player 1) + strength (player 2) + k, and so on. The formula given in formula 1 is the recursion made by this method so that the final team rating can be derived.

Here the Strength function is used to describe the rating of a player or team, S is a subset of the pool of players and T is a subset of S.

In the function, Strength(s) means the total Strength of a group, and Strength(T) means the Strength of each subset of the group. The recursion will consider and go through the powerset of the groups of players. If we want to get the overall strength values of the group, exploring the strength values between players and players is an indispensable step

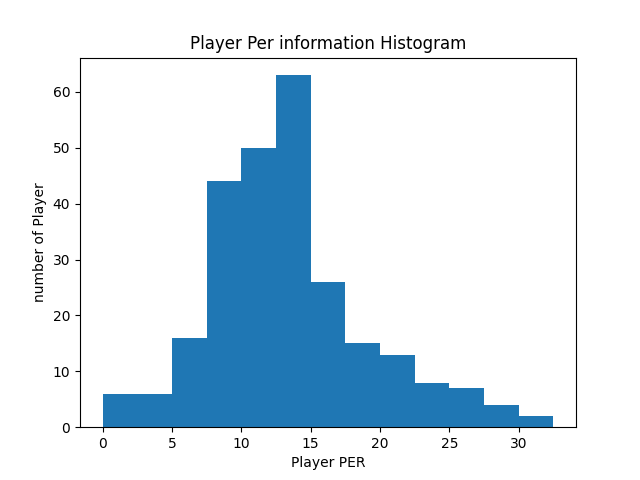
### The function to calculate the team chemistry

In the background research, we found the Elo rating could be a good way to represent the strength of each team. I searched the Elo ratings of some teams and selected the highest Elo rating when the players were all in this team to calculating the team chemistry value.



* Figure 8. the graph shows the team Elo histogram(the team selected as data)

As mentioned in the background research part, this project uses the highest Elo of the period for the teams which contain the player combinations as the data. As the graph shows, most of the teams have an Elo score between 1750 and 1800, since the average Elo is 1500, the teams selected are stronger than average. For this project, we need to find the team chemistry values of the best combinations from these 14 teams. Each of the 14 teams will have several role players in addition to the famous combinations. The chart below summarizes the information on all players, including the famous combinations, which will be used to calculate the chemistry value.



-Figure 9. the player PER information histogram

According to the background research, Elo rating is often used by the sigmoid function (logistic) for statistics and research, so we can conclude that Elo ratings are a logistic distribution. However, it is hard to find a distribution that fits the player's PER, but the PER is likely to be normally distributed, due to it increases approximately linearly with the time of each game. So in the project, the distribution of PER is considered as Normal distribution. And the Team Elo is Logistic distribution, in which the logistic distribution looks like a normal distribution This suggests that we can use the normal distribution to

approximate the logistic distribution or we can approximate the normal distribution with a logistic. (Cook, 2010) So we could normalize the Team Elo, which could help to find the team chemistry value.

Here is the formula to calculate the team chemistry value:

In the first step, the normalized team Elo is calculated from team Elo and player Elos. It is made to fit the normal distribution of the players.

The normalized team Elo can then be thought of as an average strength between the players considering the team chemistry. The team chemistry + the strength value of individual players between a set of players is approximated by the strength of the team normalized to player strength, multiplied by the number of players, taking away the players who are in the team but not in the subset of players which we try to calculate chemistry for.

### Data memo function

As mentioned in the module above, class set for powerset and team chemistry are used to integrate and calculate some information which will be used in the class Strength. class team chemistry is used to calculate the team relation value k.

As mentioned earlier, the Strength function is implemented and calculates team scores using memoization. The main feature of memoization is that there is a cache object, which is used to store some of the computed data. While the recursion is in progress, the data can be used directly, if the required data can be found directly in the cache. In this project, the data memo function can be understood as a cache function. the parameter of the data memo function is a set. This set is obtained using the powerset function, which gives all the possibilities for grouping players. The data memo function uses the team chemistry value to score and store each grouping.

Here is the pseudocode of Data memo:

Def Data\_memo(set):

New Memo={ }

For set in set:

If len(set)==1:

New Memo[set]= player rating

If len(set)==2:

New memo[set] = team chemistry value + team

If len(set)==3:

New memo[set] = team chemistry value + team

Return New Memo

The powerset size is set to a maximum of three, which means that the maximum possible combination is three players per set, as shown above. The reasons for this are explained in part 6.

### Tkinter library

For methods of making UIs, Tkinter (Geeks, 2021) is the most common library for developing GUIs (graphical user interfaces) in Python. It is a standard Python interface that provides a toolkit for making UIs. The main language used for this project is python, and Tkinter is more accessible and efficient, so finally, Tkinter was chosen to build the user interface.

### UI design



-Figure 10. the front page for UI

This is the starting page for the game, both of the players could type the name they prefer to their team, and start the game, the question mark at the bottom is used to show the rule and gameplay of this game.



-Figure11 Main UI diagram

The final user interface has been partially simplified and compared to the initial prototype it shows less detailed information about the players, such as their abilities and base positions. The overall layout of the UI depends mainly on the rules and gameplay of the game. Before the game start, the player will give the name of their team and given the order by rolling the dice. The player could choose five players which would be displayed with images and combine their fantasy league team. When the player has selected the final player, they will be taken to the final scoring page, which will show the final winner and the scoring of their respective team. From this page, players will be informed of the reasons for the high and low ratings. Which is shown below:



-Figure 12. the finished page of the UI which include the rating and team name

### Game mode

For the setting of the game modes, the project decided to have two modes for players to choose from and be entertained, the first being player versus player. However, if the player wants to play alone, the game has an AI mode for the player to choose from. Both modes are available to give players a high level of playability.

### Game rule

For the rules of the game, the order of the players is determined by rolling the dice. To ensure the competitiveness of the game, the players are selected in a fixed order, but the players are free to mix and match and to use the order rationally, for example, to break up the opponents' upcoming combinations. The order in which players are selected is shown in figure xxx. To add interest to the game, we set and filter the players in the player pool in different tiers, all players with a rating of 80 or more will be selected as data, while the final reality is 20 players selected randomly from the data, and the players are divided into 5 tiers, tier 1 players with an individual rating of 96-100, tier 2 players with a rating of 92-96, and Tier 3 players are rated 88-92, tier 4 players are rated 84-88, tier 5 players are rated 80-84. Different salaries are set for each tier, for example, tier 1 players are paid 5 coins, tier 2 players are paid 4 coins, and so on. To make it more challenging to build a team, we give each player 19 coins, In general, each player will choose two tier-one NBA players, so they will both be left with 9 coins, which will test the player's ability to match and make trade-offs. For example, if a player chooses two 4 coin cards, then he can only choose one more 1 coin card. and the player has to use a limited number of coins to build a team.



Figure 13. -the order for the players to choose from a pool of basketball players

# 6.implementation

## Explore the suitable model for data structure

For the basic setting of the data, the main objective of this project was to explore and calculate the tacit values of the relationships between players (team chemistry). Based on the background research, I selected 14 famous combinations and compiled their team and teammate data for this project.

The initial stage of the model is considering which method to use to present the graph

-Figure14. The player relation model

This image shows how the relationships between players can be represented by the adjacency list.

the adjacency list is a chain structure of the graph it will have the vector, states, and a line connecting the state. In this diagram, Player A connects to Player B, which means Player A has a relationship with Player B. And the result for the adjacency list is easy to represent Here is the graph to show the relation result:



-Figure 15. adjacency list to show the relationship

However, there are other ways to represent the relation between players.

Adjacency list can also be represented as arrays:



Figure 16.-The adjacency array

In this table, the horizontal rows represent the number of connections of that point and the vertical columns represent each unit, with column 0 indicating how many connections there are.

The graph is used to be designed as the model to show the relationship combine between each player. A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph. And it is consists of a finite set of vertices(or nodes) and a set of edges that connect a pair of nodes. The Adjacency Matrix is the main factor that gives us a way to represent our player relation graph in an efficient and structured procedure. It is easy to represent nodes and edges by creating a matrix table as the graph shown below:



Figure 17-adjacency matrix graph shows the value

Adjacency list memory usage depends more on the number of edges (rather than the number of nodes), if the data is sparse, using an adjacency list will save a lot of space compared to an adjacency matrix, but the data in this project is denser, so the adjacency matrix is the better choice. In addition, the adjacency matrix can be used to find and check the values of specific edges more quickly, which is important for getting the team chemistry values.

Finally, the power set model is set up, which not only identifies relationships between individuals and groups but also between groups and groups



Figure 18-adjacency matrix graph shows the value between power set and power set

In the final stage, both horizontal and vertical nodes will be power sets, which represent the possibility of all player combinations and the function represented can be shown as P(p) \* P(p) = Graph. And edges are used in this project to show the relationship between each power group, if there is no relationship between the groups then the edge will be entered as null, otherwise, the actual value will be entered and saved in the table to show that a relationship does exist between the selected power groups.

### The rating implement method change

The project initially used pure recursion to implement the final scoring of the integrated teams, but during the run, it was found that the program took a long time to run using pure recursion, and when testing with larger data, it was found that there was a Stack overflow situation.

Here is the function of the Strength method.

Strength(s) means the total rating of the group, and Strength(T) means the rating of a subset in the group.

The pseudocode code for this function by using recursion is shown below, As it relies on the way of recursion, it is necessary to set up the base case, so for this project’s pure recursion is if the number of set one, it will straight return that player’s rating.

Define Strength(set)

If num == 1

Return Rating(set[0])

Else

strength = 0;

For each subset in itertools.combination(set, len(set)-1)

Strength += Strength(subset)

Strength / (len(set))

Return Strength()

It will cause stack overflow condition, which is the program uses more memory space than is available on the call stack, for example like the graph to calculate the overall rating of (player1,player2,player3). The algorithm will calculate the rating of (player1,player2) and (player1,player3) and (player2,player3) then combine their rating to get the combined rating of three players, if there are requests to rate a group of four then it will do a recursion until it finds the strength of the subsets it needs to calculate which will be explained in detail in the implementation part, In this case, the recursion will do a lot of redundant work and have low efficiency. So I try to find the more efficient way to solve the calculation (improve the fib recursion), which is the Memoization, the basic idea for memoization is an attempt to save all of the results obtained from the recursion. And take out the result directly when we use it to calculate further information. Therefore, an array is defined to store the calculated data and then query from the array when needed, which will save unnecessary calculations and improve the efficiency of the program.

## Fibonacci sequence recursion

Fibonacci sequence (Ghose, 2019) is a sequence in which each number is the sum of the first two numbers. The number at a specific position in the Fibonacci sequence can be obtained using recursive methods.

The pseudocode for fib can be conclude as:

Define Fib(n)

If n = 0

Return 0

Else

Return Fib(n-1)+Fib(n-2)

****

**-** Figure 19.the fib diagram shows the fib work process

## Memoization

In computing, memoization (Koroleva, 2018) is an optimization technique that focuses on speeding up computer programs by storing the results of expensive function calls and returning the cached results when the same input occurs again. memoization is an optimization of the recursion function. Unlike the fib function, it sets up an object called a cache, which stores all the return values we have obtained, and when the function is executed it will check whether the cache has the required data, and if it does, it can be called directly, if not, the next calculation will be done, which can effectively reduce the number of function calls.

****

* **Figure 20.the graph shows the memoization working process**



-Figure 21. how to implement the memoization in this project

The data in the green part of the image can be used without calculation because memoization defines a cache to store the calculated data, which can be used if the same data is needed again during a loop, which is not possible with pure recursion.

Here is the pseudocode for Memoization:

Define Strength(set)

Memo = keys:powerset5 -> values: zeros(len(powerset5(players)))

If num == 1

Memo[set] = Rating(set[0])

Return Rating(set[0])

Else

If (Memo[set] == 0)

strength = 0;

For each subset in itertools.combination(set, len(set)-1)

strength += Strength(subset)

strength / (len(set))

Memo[set] = strength

Return Memo[set]

The pseudocode provides an implementation of the Strength function using a memo on the previously evaluated Strengths, and calls recursively the strength function for all the Strengths it needs, and will calculate the value if it is not in the Memo already. Team chemistry between subsets of the group is modeled by changing the values in the memo to be greater or less than the average of the strengths of the subsets.

### Implementation in a simple robot

In this project, there are two modes are considered which the player could choose to against with a player, and the player could choose to against with the computer choice. if the user wants to play on its own, then he/she can choose to play with AI. During the playing, AI will be able to choose its team, just like the second player. However, In this project, the robot has not been set up to be very complex, due to time constraints. To ensure the cost of AI will not more than 19 points the setting of AI choose the player is that the AI player will only choose the basketball player from T3 which means the player in that level will cost 3 points. So at this level, the AI player will not be very strong. But this mode could help the player to get to know more about how will the rating be and the whole process of playing this game.

### Implementation issue

During the development of the project, some problems were encountered, mainly due to the availability of player data. The initial set-up was to use the most famous combinations of players, but the rating system was built to require more data, so the data size and entry method were changed.

The data was initially set to be small, so it was entered manually, but team Elo was used to calculate the team chemistry value when the project was underway, as the team Elo rating was based on the strength of the team based on everyone in the team, but only the team Elo rating was used to This revealed that the data was not comprehensive enough and that it would be very difficult to add more information manually. Therefore, we chose to expand the overall data by searching for all players in the 14 teams and adding them to the CSV, using the CSV to store and modify the data.

Another problem arose during the testing of the Data memo when several teams ended up with values that differed significantly from the true value, and it was thought that there might be a problem with the application of the algorithm. However, after doing the most basic calculation of the average player ability values, the problem was discovered. As the project selected 14 teams with a good mix of players at different times, some players were present in several teams at the same time, and the final ability value of the player depended on the last team entered, which led to errors in the scores of several teams.

When designing the relationships between players, the final choice was to use the powerset to find the relationship values between players. Initially, the maximum size of the powerset was set to 4.

As the team combination is four, we will get millions of possibilities, which will seriously affect the running time of the program, and as a large part of the data used is from combinations of two to three players, it was decided to reduce the maximum size to 3.

# 7.Testing

## Unit test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Module | Normal Value | | Extreme Value | | Anomalous Value | |
| Value | Test result | Value | Test result | Value | Test result |
| Team Elo  Get team Elo() | ‘Utah Jazz’ | 1764 | - | - | ‘Good’ | null |
| Team Elo Get team Average() | ‘ Nuggets’ | 81.375 | ’14 team’ | Error | ‘Not team’ | Error |
| Player  Get\_wholeGP | ‘Chris Bosh’ | Heat | - | - | ‘other people’ | Error |
| Player  Whole\_rating | ‘Ray Allen’ | 17.6 | - | - | ‘Hello’ | Can’t find |
| Set for Powerset  Powerset() | Player List | A list includes all possible powerset for player | A huge Player List | Stack over  flow | [1,2,3,4] | List all possible powerset for the number in the List. |
| Team\_infor() | 'Utah Jazz', 'Karl Malone', 'John Stockton', 'Jeff Hornacek' | 23.9613 | - | - | 'Utah Jazz', 'A', 'B', 'C' | Error |
| Set for Powerset  Data memo() | Player powerser List | Rating the score for each powerset | A huge Player List | Stack overflow |  |  |
| Strength() | Karl Malone', 'John Stockton | 104.6507 | ’30 player’ | Stack overflow | ‘nobody’ | Value Error: r must be non-negative |
| Strength easy() | ‘Heat’ | 79.0687 | - | - | ‘Kobe Bryant’ | Error |

## Integration test

|  |  |
| --- | --- |
| Integration | condition |
| Powerset-Data memo  Strength – data memo | A cache memo set as expected  Strength function can get the value from cache data |
| Data memo- Team Infor | The value calculated from team infor can be stored into Data memo |
| Team\_infor-whole rating | In the team infor function, the player rating can be obtained by using the whole rating function |
| Team infor – Team Elo rating | In the team infor function, each team Elo rating can be got from the Team elo function |
| Strength-Final Rating page | The final rating page does show the rating of the player team |

## System test

|  |  |
| --- | --- |
| Case | Condition |
| Player select teams normally | A wins as expected |
| Players choose players worth more than their budget | Can not choose, forced to choose another one |
| The Player want to check the rule | A page of Rule are shown |
| Player re-choose the NBA players during the choose section | The player could be changed but the money of the player may be lost |
| The User chooses the same player the whole time | The player cannot be chosen, as it was already fixed, but the money still costs. |
| The player does not choose the player and press ok | The player presses the ok button with doing nothing, he needs to re-choose. |
| The player chooses the player but does not have enough money | The player can not choose the player which can not afford it. And the team have to be fixed |

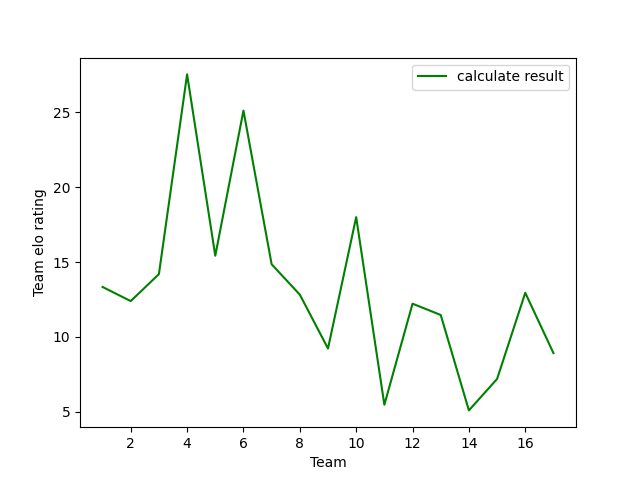
# 8.Evaluation

This section will contain self-reflection and some quantitative analysis of the calculated scores to assess the accuracy of the scores. In addition to this, there will be a comparison of this scoring system with other scoring systems that exist. These were chosen over user testing because the project is more focused on the back-end functionality of the system rather than the end-user experience.



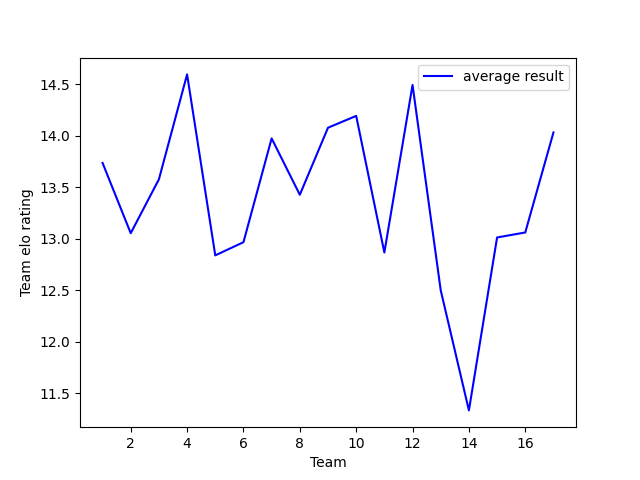
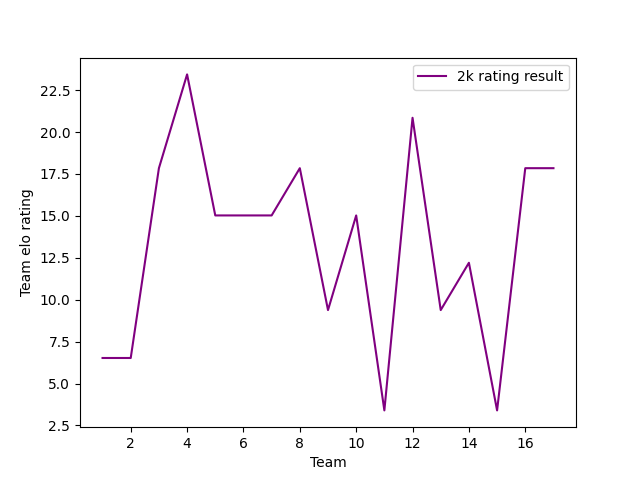
-Figure 22.confusion matrix graph

Above is the confusion matrix graph, The horizontal and vertical rows show the team strength rating obtained from 2krating and the strength of a team that obtained by using our formula, the middle row shows the difference between the two data and is colored to represent the size of the difference, with lighter colors indicating smaller differences and darker colors indicating larger differences

****

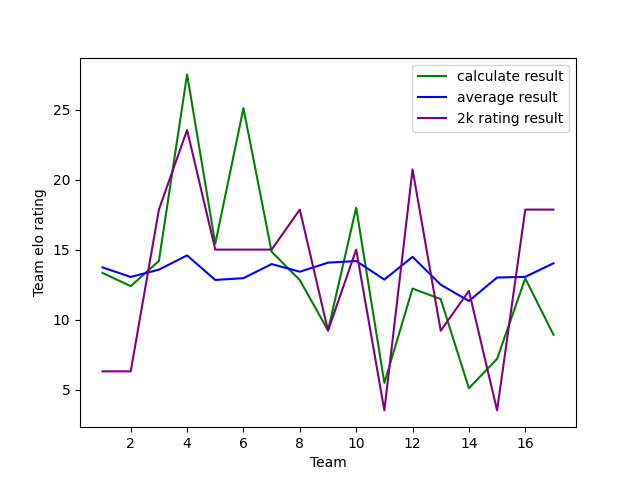
-Figure 23.calculate value

This value is what we calculated for the rating of the whole team from our formula, which is the team that includes the famous combination or the team that wins is the champion, as we can see there are 17 teams.

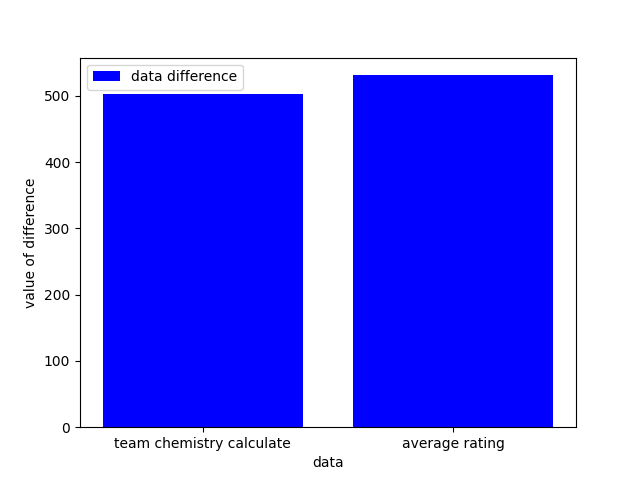
-Figure 24.rating calculate from average - Figure 25.the rating from 2krating website

The graph with a blue line is the 17 teams rating simply calculate by adding the per of all of the players in the team and divide the number of players. The graph with the purple line is the 2k rating for these 17 teams, which the data have been normalized. Although a clear trend change can be seen in each of these three plots, the magnitude of their y-axis ranges are different. To better observe the differences between these three results, I have put the three results into a single plot.



-Figure26. shows the calculated team strength and the average of team strength and 2krating result

Three different results are shown in the figure, the calculated result is that we calculate the team strength by using our formula and the 2k rating result is that the data get from the 2krating website, and the data normalized by per, according to the graph we could find most of the value are similar to each other, except the team 6 and team 14, and our calculate value follow a similar path to the true value.



-Figure 27.total data difference value bar chart

The data difference value is calculated by the square different of the data from the true data, the true data being the data from 2krating. It is used to check how far the data we calculate differ from 2krating, as the graph shows there is two bar in the graph, the first one is the data we calculate by using our team chemistry and Strength function, the second one is the average rating calculated by simply add all the player per in that team and divide the number of players, we could see that our team chemistry calculate is more close to the 2krating, the lower the height the smaller the gap to the 2krating data.

# 9.Conclusion

In conclusion, this report shows the overall process of a simple game based on a scoring system. It shows that this scoring system entails thorough background research, it covers the design methods used for development as well as the design of the UI and the rules of the game. The implementation phase explains some of the model or methodological changes adopted, some of the problems encountered during implementation, and finally includes the evaluation and testing of the data.

The main aim of this project was to first, design a reasonable model to describe the relationship between players. Secondly, generate a team rating system based on exploring player-player relationships(team chemistry vale). In the evaluation section, there is a graph with two bars representing the deviation from the real data. The first one is based on the average Elo rating and the average level of the players as the real data. The other one takes the results of another rating system and normalizes them and compares them with our ratings as real data. It was found that some of the famous combinations had a positive team chemistry value and some had a negative relationship value. It was found that most of the combinations with a negative value were those where the individual players were very good, but the team did not end up playing very well that season. The data used for this project is not large enough at the moment, so there is a risk of inaccuracy. There are also some role players whose details were not available when the player information was entered. So there is a good chance that inaccuracies will occur. Of course, the final data will have to be compared and improved by setting different real data. However, based on the ratings and data obtained so far I believe that the objectives and requirements defined in this article have been achieved.

## Compare with existing work

### Limitation

* The Player PER data may still not be accurate

The PER could represent the player's efficiency during the game. However, It will be influenced by the number of games the player has played. In the data, some teams have a few players who have a high PER due to the lack of games played. This is a source of inaccuracy.

* There are exist some combinations which the player may choose that may harm the team

The rating system is focusing on the team chemistry between the player. However, It is normal to have positive or negative team chemistry values in a team, and according to the calculation, there do exist some famous groups that do not have a good team chemistry value. This will impact the final rating of the team.

* The team Elo rating may not be accurate

In the background section, Team Elo is introduced as one of the main data, but the team Elo could change depending on the daily competition. The project chose the peak of Team Elo, this may cause the data to not be accurate as well

* Purely based on previous records

As mentioned in the background research section, all of the data are based on the historical records from the website, so the player performance is never up to date.

Both NBA fantasy and Yahoo basketball fantasy use data based on how players perform in each game and their databases are updated daily, meaning they have more accurate and up-to-date data. And what this leads to is that their predictions of players will be more accurate.

### Pros

Compared to NBA Fantasy and Top Trumps and Yahoo sports, this project will take more into account the player-to-player chemistry. Unlike Yahoo sports and NBA Fantasy, this project will include more data on older players. Which help the user to get more information about those old famous team.

## Future work

1. Expand data

It is possible that the final results of this project are not very accurate because the data stored is too small. To ensure the accuracy of the data, it is necessary to store more data. It is also crucial to devise a method to calculate individual player ability values, as they are currently calculated by player efficiency.

1. Improve Against AI mode

For the moment for this project, the mode only supports confrontations between two players, and a simple AI mode in which the AI player could choose the NBA player but not smart enough. The future work for this part could be to add different levels for the AI which the AI player could choose the player like a real human. This might be done by learning TensorFlow and using TensorFlow to implement machine learning to set up how the Ai works. The AI is currently set up to work in two ways, the first being brute force hacking and the second being using tensor flow to train the machine and then pit the player against the AI. The player can choose the mode and difficulty according to their needs.

1. Optimize the UI

The current UI of this project is still very plain and the layout is not yet well set up. Players can only see a very limited amount of information about the player when selecting a player and are not yet able to see the player's ability stats such as three points, two points, defense, etc. It would be helpful if this feature was added to the UI, especially for those who don't know the NBA, so that they can have a better understanding of each player's characteristics and outstanding abilities. In addition, the structure of the UI could be adjusted and made more colorful, which would greatly improve the user experience.

1. Expand the model

This project can now give a rating based on the player's choice of player, With more information and time, it is possible not only to display each team's rating on the final page but also to set a prediction of the number of points a player will receive during this simulated rating. For example, how many points Player A has scored, how many goals he has stolen.

1. Extent prediction function

As mentioned in the limitations section, this system can only be used to calculate team chemistry between former teams, but not to predict team chemistry between new teams.

The future work for this part is to try to enter some new data such as comparable players, try to find the closest previous player models to modern players, do this for each player in the team, and see if there are any play records for similar types of players. This will allow you to make predictions for new teams.

# 10.Appendix

# Reference

BCS the Chartered for IT, 2019. *BCS Code of Conduct.* [Online]   
Available at: https://www.bcs.org  
[Accessed 1 11 2020].

Cook, J. D., 2010. *Normal approximation to logistic distribution.* [Online]   
Available at: https://www.johndcook.com/blog/2010/05/18/normal-approximation-to-logistic/

Cook, J. D., 2010. *Normal approximation to logistic distribution.* [Online]   
Available at: https://www.johndcook.com/blog/2010/05/18/normal-approximation-to-logistic/

Fein, Z., 2009. *Cracking the Code: How to Calculate Hollinger's PER Without All the Mess.* [Online]   
Available at: https://bleacherreport.com/articles/113144- cracking-the-code-how-to-calculate-hollingers-per-without-all-the-mess  
[Accessed 10 11 2020].

Feldman, D., 2021. *Daryl Morey says 3-pointers worth too much: ‘It skews the game’.* [Online]   
Available at: https://nba.nbcsports.com/2021/04/06/daryl-morey-says-3-pointers-worth-too-much-it-skews-the-game/?partner=MSN  
[Accessed 11 2020].

FiveThirtyEight, 2021. *The Complete History Of The NBA.* [Online]   
Available at: https://projects.fivethirtyeight.com/complete-history-of-the-nba/#lakers  
[Accessed 2 2021].

Geeks, G. f., 2021. *Python Tkinter Tutorial.* [Online]   
Available at: https://www.geeksforgeeks.org/python-tkinter-tutorial/#:~:text=Tkinter%20is%20the%20most%20commonly%20used%20library%20for,applications%20with%20Tkinter%20becomes%20the%20fastest%20and%20easiest  
[Accessed 3 2021].

Ghose, T., 2019. *What is fibonacci Sequence.* [Online]   
Available at: https://www.livescience.com/37470-fibonacci-sequence.html  
[Accessed 2 2021].

Khan, E., 2013. *Advanced NBA Stats for Dummies: How to Understand the New Hoops Math.* [Online]   
Available at: https://bleacherreport.com/articles/1813902-advanced-nba-stats-for-dummies-how-to-understand-the-new-hoops-math

Koroleva, L., 2018. *Memoization. Improving Recursive Solution for Fibonacci Sequence Problem..* [Online]   
Available at: https://medium.com/@nothingisfunny/memoization-improving-recursive-solution-for-fibonacci-sequence-problem-c02dab7a74e5  
[Accessed 2 2020].

OpenGenus IQ, 2021. *Learn Computer Science.* [Online]   
Available at: <https://iq.opengenus.org/graph-representation-adjacency-matrix-adjacency-list/  
[Accessed 1 2021].

Pina, M., 2020. *The NBA has a chemistry problem.* [Online]   
Available at: https://www.sbnation.com/nba/2020/4/23/21229237/nba-roster-chemistry-free-agency-team-turnover  
[Accessed 12 2020].

School of Engineering and informatics, The University of Sussex, 2020. *information for studens.* [Online]   
Available at: http://www.sussex.ac.uk/ei/internal/forstudents/informatics/undergraduate/finalyearprojects/i%20nformationforstudents  
[Accessed 2 11 2020].

Unknown, 2019. *The Basketball Movement.* [Online]   
Available at: https://thebasketballmovement.com/blog/2019/4/27/what-team-chemistry-means  
[Accessed 11 2020].

Unknown, n.d. *Calculating PER.* [Online]   
Available at: https://www.basketball-reference.com/about/per.html  
[Accessed 7 11 2020].

## Data Source

Team Elo rating Available at: https://projects.fivethirtyeight.com/complete-history-of-the-nba/#mavericks

Player rating Available at: <https://www.2kratings.com>

Player PER Available at: <https://www.basketball-reference.com>

Data used:

PER:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rk | name | G | MP | PER | Team |  |
| 1 | Karl Malone\malonka01 | 81 | 3030 | 27.9 | 1997-98 Utah Jazz | |
| 2 | Jeff Hornacek\hornaje01 | 80 | 2460 | 19.3 | 1997-98 Utah Jazz | |
| 3 | Bryon Russell\russebr01 | 82 | 2219 | 12.9 | 1997-98 Utah Jazz | |
| 4 | Adam Keefe\keefead01 | 80 | 2047 | 14.5 | 1997-98 Utah Jazz | |
| 5 | John Stockton\stockjo01 | 64 | 1858 | 21.8 | 1997-98 Utah Jazz | |
| 6 | Howard Eisley\eisleho01 | 82 | 1726 | 13.6 | 1997-98 Utah Jazz | |
| 7 | Shandon Anderson\andersh01 | 82 | 1602 | 15.7 | 1997-98 Utah Jazz | |
| 8 | Greg Foster\fostegr01 | 78 | 1446 | 8.9 | 1997-98 Utah Jazz | |
| 9 | Greg Ostertag\ostergr01 | 63 | 1288 | 12.5 | 1997-98 Utah Jazz | |
| 10 | Antoine Carr\carran01 | 66 | 1086 | 9.8 | 1997-98 Utah Jazz | |
| 11 | Chris Morris\morrich01 | 54 | 538 | 14.7 | 1997-98 Utah Jazz | |
| 12 | Jacque Vaughn\vaughja01 | 45 | 419 | 6.9 | 1997-98 Utah Jazz | |
| 13 | William Cunningham\cunniwi01 | 6 | 38 | 8.7 | 1997-98 Utah Jazz | |
| 14 | Troy Hudson\hudsotr01 | 8 | 23 | 19.1 | 1997-98 Utah Jazz | |
| 1 | Wilt Chamberlain\chambwi01 | 82 | 3630 | 20.3 | 1970-71 Los Angeles Lakers | |
| 2 | Happy Hairston\hairsha01 | 80 | 2921 | 16.2 | 1970-71 Los Angeles Lakers | |
| 3 | Jerry West\westje01 | 69 | 2845 | 23.5 | 1970-71 Los Angeles Lakers | |
| 4 | Gail Goodrich\goodrga01 | 79 | 2808 | 14.6 | 1970-71 Los Angeles Lakers | |
| 5 | Keith Erickson\erickke01 | 73 | 2272 | 11.5 | 1970-71 Los Angeles Lakers | |
| 6 | Jim McMillian\mcmilji01 | 81 | 1747 | 12.2 | 1970-71 Los Angeles Lakers | |
| 7 | Willie McCarter\mccarwi01 | 76 | 1369 | 8.5 | 1970-71 Los Angeles Lakers | |
| 8 | Rick Roberson\roberri01 | 65 | 909 | 11.8 | 1970-71 Los Angeles Lakers | |
| 9 | Fred Hetzel\hetzefr01 | 59 | 613 | 12.4 | 1970-71 Los Angeles Lakers | |
| 10 | Pat Riley\rileypa01 | 54 | 506 | 12.3 | 1970-71 Los Angeles Lakers | |
| 11 | John Tresvant\tresvjo01 | 8 | 66 | 22.7 | 1970-71 Los Angeles Lakers | |
| 12 | Elgin Baylor\bayloel01 | 2 | 57 | 9 | 1970-71 Los Angeles Lakers | |
| 13 | Earnie Killum\killuea01 | 4 | 12 | -5.3 | 1970-71 Los Angeles Lakers | |
| 1 | Tony Parker\parketo01 | 80 | 2735 | 18 | 2004-05 San Antonio Spurs | |
| 2 | Bruce Bowen\bowenbr01 | 82 | 2627 | 9.5 | 2004-05 San Antonio Spurs | |
| 3 | Tim Duncan\duncati01 | 66 | 2203 | 27 | 2004-05 San Antonio Spurs | |
| 4 | Manu Ginóbili\ginobma01 | 74 | 2193 | 22.3 | 2004-05 San Antonio Spurs | |
| 5 | Rasho Nesterović\nestera01 | 70 | 1785 | 12 | 2004-05 San Antonio Spurs | |
| 6 | Brent Barry\barrybr01 | 81 | 1742 | 14 | 2004-05 San Antonio Spurs | |
| 7 | Robert Horry\horryro01 | 75 | 1396 | 14.9 | 2004-05 San Antonio Spurs | |
| 8 | Devin Brown\brownde02 | 67 | 1238 | 14.6 | 2004-05 San Antonio Spurs | |
| 9 | Beno Udrih\udrihbe01 | 80 | 1149 | 14.3 | 2004-05 San Antonio Spurs | |
| 10 | Malik Rose\rosema01 | 50 | 862 | 14 | 2004-05 San Antonio Spurs | |
| 11 | Tony Massenburg\masseto01 | 61 | 699 | 8.5 | 2004-05 San Antonio Spurs | |
| 12 | Nazr Mohammed\mohamna01 | 23 | 414 | 14.2 | 2004-05 San Antonio Spurs | |
| 13 | Mike Wilks\wilksmi01 | 48 | 278 | 11 | 2004-05 San Antonio Spurs | |
| 14 | Sean Marks\marksse01 | 23 | 244 | 8.7 | 2004-05 San Antonio Spurs | |
| 15 | Glenn Robinson\robingl01 | 9 | 157 | 17.2 | 2004-05 San Antonio Spurs | |
| 16 | Dion Glover\glovedi01 | 7 | 68 | 14 | 2004-05 San Antonio Spurs | |
| 17 | Linton Johnson\johnsli01 | 2 | 15 | -3.4 | 2004-05 San Antonio Spurs | |
| 1 | Klay Thompson\thompkl01 | 78 | 2649 | 17.4 | 2016-17 Golden State Warriors | |
| 2 | Stephen Curry\curryst01 | 79 | 2638 | 24.6 | 2016-17 Golden State Warriors | |
| 3 | Draymond Green\greendr01 | 76 | 2471 | 16.5 | 2016-17 Golden State Warriors | |
| 4 | Kevin Durant\duranke01 | 62 | 2070 | 27.6 | 2016-17 Golden State Warriors | |
| 5 | Andre Iguodala\iguodan01 | 76 | 1998 | 14.3 | 2016-17 Golden State Warriors | |
| 6 | Shaun Livingston\livinsh01 | 76 | 1345 | 10.1 | 2016-17 Golden State Warriors | |
| 7 | Zaza Pachulia\pachuza01 | 70 | 1268 | 16.1 | 2016-17 Golden State Warriors | |
| 8 | Ian Clark\clarkia01 | 77 | 1137 | 13.1 | 2016-17 Golden State Warriors | |
| 9 | Patrick McCaw\mccawpa01 | 71 | 1074 | 8.6 | 2016-17 Golden State Warriors | |
| 10 | David West\westda01 | 68 | 854 | 16.6 | 2016-17 Golden State Warriors | |
| 11 | JaVale McGee\mcgeeja01 | 77 | 739 | 25.2 | 2016-17 Golden State Warriors | |
| 12 | James Michael McAdoo\mcadoja01 | 52 | 457 | 13.1 | 2016-17 Golden State Warriors | |
| 13 | Kevon Looney\looneke01 | 53 | 447 | 13.6 | 2016-17 Golden State Warriors | |
| 14 | Matt Barnes\barnema02 | 20 | 410 | 10.7 | 2016-17 Golden State Warriors | |
| 15 | Anderson Varejão\varejan01 | 14 | 92 | 9.4 | 2016-17 Golden State Warriors | |
| 16 | Damian Jones\jonesda03 | 10 | 85 | 5.3 | 2016-17 Golden State Warriors | |
| 17 | Briante Weber\weberbr01 | 7 | 46 | 5.9 | 2016-17 Golden State Warriors | |
| 1 | LeBron James\jamesle01 | 76 | 2877 | 31.6 | 2012-13 Miami Heat | |
| 2 | Chris Bosh\boshch01 | 74 | 2454 | 20 | 2012-13 Miami Heat | |
| 3 | Dwyane Wade\wadedw01 | 69 | 2391 | 24 | 2012-13 Miami Heat | |
| 4 | Mario Chalmers\chalmma01 | 77 | 2068 | 13.3 | 2012-13 Miami Heat | |
| 5 | Ray Allen\allenra02 | 79 | 2035 | 14.7 | 2012-13 Miami Heat | |
| 6 | Shane Battier\battish01 | 72 | 1786 | 10.7 | 2012-13 Miami Heat | |
| 7 | Norris Cole\coleno01 | 80 | 1590 | 7.9 | 2012-13 Miami Heat | |
| 8 | Udonis Haslem\hasleud01 | 75 | 1414 | 9.9 | 2012-13 Miami Heat | |
| 9 | Mike Miller\millemi01 | 59 | 900 | 13.5 | 2012-13 Miami Heat | |
| 10 | Rashard Lewis\lewisra02 | 55 | 792 | 11.1 | 2012-13 Miami Heat | |
| 11 | Chris Andersen\anderch01 | 42 | 624 | 17.4 | 2012-13 Miami Heat | |
| 12 | Joel Anthony\anthojo01 | 62 | 566 | 10.4 | 2012-13 Miami Heat | |
| 13 | James Jones\jonesja02 | 38 | 221 | 8.1 | 2012-13 Miami Heat | |
| 14 | Juwan Howard\howarju01 | 7 | 51 | 9.9 | 2012-13 Miami Heat | |
| 15 | Jarvis Varnado\varnaja01 | 8 | 40 | -4.1 | 2012-13 Miami Heat | |
| 16 | Josh Harrellson\harrejo01 | 6 | 31 | 8.5 | 2012-13 Miami Heat | |
| 17 | Terrel Harris\harrite01 | 7 | 29 | 7.2 | 2012-13 Miami Heat | |
| 18 | Dexter Pittman\pittmde01 | 4 | 12 | 17 | 2012-13 Miami Heat | |
| 1 | Michael Jordan\jordami01 | 82 | 3090 | 29.4 | 1995-96 Chicago Bulls | |
| 2 | Scottie Pippen\pippesc01 | 77 | 2825 | 21 | 1995-96 Chicago Bulls | |
| 3 | Toni Kukoč\kukocto01 | 81 | 2103 | 20.4 | 1995-96 Chicago Bulls | |
| 4 | Dennis Rodman\rodmade01 | 64 | 2088 | 13.6 | 1995-96 Chicago Bulls | |
| 5 | Steve Kerr\kerrst01 | 82 | 1919 | 15.2 | 1995-96 Chicago Bulls | |
| 6 | Ron Harper\harpero01 | 80 | 1886 | 14.4 | 1995-96 Chicago Bulls | |
| 7 | Luc Longley\longllu01 | 62 | 1641 | 11.9 | 1995-96 Chicago Bulls | |
| 8 | Bill Wennington\wennibi01 | 71 | 1065 | 11 | 1995-96 Chicago Bulls | |
| 9 | Jud Buechler\buechju01 | 74 | 740 | 14.1 | 1995-96 Chicago Bulls | |
| 10 | Dickey Simpkins\simpkdi01 | 60 | 685 | 10.4 | 1995-96 Chicago Bulls | |
| 11 | Randy Brown\brownra02 | 68 | 671 | 11.2 | 1995-96 Chicago Bulls | |
| 12 | Jason Caffey\caffeja01 | 57 | 545 | 7.9 | 1995-96 Chicago Bulls | |
| 13 | James Edwards\edwarja01 | 28 | 274 | 3.5 | 1995-96 Chicago Bulls | |
| 14 | John Salley\sallejo01 | 17 | 191 | 8.7 | 1995-96 Chicago Bulls | |
| 15 | Jack Haley\haleyja01 | 1 | 7 | 1.8 | 1995-96 Chicago Bulls | |
| 1 | Shaquille O'Neal\onealsh01 | 74 | 2924 | 30.2 | 2000-01 Los Angeles Lakers | |
| 2 | Kobe Bryant\bryanko01 | 68 | 2783 | 24.5 | 2000-01 Los Angeles Lakers | |
| 3 | Horace Grant\grantho01 | 77 | 2390 | 14.3 | 2000-01 Los Angeles Lakers | |
| 4 | Rick Fox\foxri01 | 82 | 2291 | 13.8 | 2000-01 Los Angeles Lakers | |
| 5 | Brian Shaw\shawbr01 | 80 | 1833 | 10.9 | 2000-01 Los Angeles Lakers | |
| 6 | Robert Horry\horryro01 | 79 | 1587 | 11 | 2000-01 Los Angeles Lakers | |
| 7 | Isaiah Rider\rideris01 | 67 | 1206 | 11.8 | 2000-01 Los Angeles Lakers | |
| 8 | Ron Harper\harpero01 | 47 | 1139 | 12.5 | 2000-01 Los Angeles Lakers | |
| 9 | Mike Penberthy\penbemi01 | 53 | 851 | 11.2 | 2000-01 Los Angeles Lakers | |
| 10 | Derek Fisher\fishede01 | 20 | 709 | 14 | 2000-01 Los Angeles Lakers | |
| 11 | Mark Madsen\madsema01 | 70 | 641 | 9.2 | 2000-01 Los Angeles Lakers | |
| 12 | Devean George\georgde01 | 59 | 593 | 6.2 | 2000-01 Los Angeles Lakers | |
| 13 | Tyronn Lue\luety01 | 38 | 468 | 8.7 | 2000-01 Los Angeles Lakers | |
| 14 | Greg Foster\fostegr01 | 62 | 451 | 9.5 | 2000-01 Los Angeles Lakers | |
| 15 | Stanislav Medvedenko\medvest01 | 7 | 39 | 21.8 | 2000-01 Los Angeles Lakers | |
| 1 | Paul Pierce\piercpa01 | 80 | 2874 | 19.6 | 2007-08 Boston Celtics | |
| 2 | Ray Allen\allenra02 | 73 | 2624 | 16.4 | 2007-08 Boston Celtics | |
| 3 | Kevin Garnett\garneke01 | 71 | 2328 | 25.3 | 2007-08 Boston Celtics | |
| 4 | Rajon Rondo\rondora01 | 77 | 2306 | 15.6 | 2007-08 Boston Celtics | |
| 5 | Kendrick Perkins\perkike01 | 78 | 1912 | 13.3 | 2007-08 Boston Celtics | |
| 6 | James Posey\poseyja01 | 74 | 1821 | 12 | 2007-08 Boston Celtics | |
| 7 | Eddie House\houseed01 | 78 | 1480 | 13 | 2007-08 Boston Celtics | |
| 8 | Tony Allen\allento01 | 75 | 1373 | 10.7 | 2007-08 Boston Celtics | |
| 9 | Glen Davis\davisgl01 | 69 | 940 | 11.3 | 2007-08 Boston Celtics | |
| 10 | Leon Powe\powele01 | 56 | 809 | 20.9 | 2007-08 Boston Celtics | |
| 11 | Brian Scalabrine\scalabr01 | 48 | 512 | 5.1 | 2007-08 Boston Celtics | |
| 12 | Sam Cassell\cassesa01 | 17 | 299 | 11 | 2007-08 Boston Celtics | |
| 13 | P.J. Brown\brownpj01 | 18 | 209 | 10.2 | 2007-08 Boston Celtics | |
| 14 | Scot Pollard\pollasc01 | 22 | 173 | 8.7 | 2007-08 Boston Celtics | |
| 15 | Gabe Pruitt\pruitga01 | 15 | 95 | 8.3 | 2007-08 Boston Celtics | |
| 1 | Gary Payton\paytoga01 | 81 | 3162 | 19.6 | 1995-96 Seattle SuperSonics | |
| 2 | Hersey Hawkins\hawkihe01 | 82 | 2823 | 16.1 | 1995-96 Seattle SuperSonics | |
| 3 | Shawn Kemp\kempsh01 | 79 | 2631 | 22.6 | 1995-96 Seattle SuperSonics | |
| 4 | Detlef Schrempf\schrede01 | 63 | 2200 | 17.3 | 1995-96 Seattle SuperSonics | |
| 5 | Sam Perkins\perkisa01 | 82 | 2169 | 15.4 | 1995-96 Seattle SuperSonics | |
| 6 | Vincent Askew\askewvi01 | 69 | 1725 | 11.9 | 1995-96 Seattle SuperSonics | |
| 7 | Ervin Johnson\johnser02 | 81 | 1519 | 13.1 | 1995-96 Seattle SuperSonics | |
| 8 | Nate McMillan\mcmilna01 | 55 | 1261 | 12.9 | 1995-96 Seattle SuperSonics | |
| 9 | Frank Brickowski\brickfr01 | 63 | 986 | 8.6 | 1995-96 Seattle SuperSonics | |
| 10 | David Wingate\wingada01 | 60 | 695 | 8.1 | 1995-96 Seattle SuperSonics | |
| 11 | Eric Snow\snower01 | 43 | 389 | 10.7 | 1995-96 Seattle SuperSonics | |
| 12 | Steve Scheffler\schefst01 | 35 | 181 | 10.3 | 1995-96 Seattle SuperSonics | |
| 13 | Sherell Ford\fordsh01 | 28 | 139 | 16.4 | 1995-96 Seattle SuperSonics | |
| 1 | Larry Bird\birdla01 | 82 | 3113 | 25.6 | 1985-86 Boston Celtics | |
| 2 | Dennis Johnson\johnsde01 | 78 | 2732 | 14.7 | 1985-86 Boston Celtics | |
| 3 | Robert Parish\parisro01 | 81 | 2567 | 18.8 | 1985-86 Boston Celtics | |
| 4 | Danny Ainge\aingeda01 | 80 | 2407 | 13.6 | 1985-86 Boston Celtics | |
| 5 | Kevin McHale\mchalke01 | 68 | 2397 | 21.6 | 1985-86 Boston Celtics | |
| 6 | Jerry Sichting\sichtje01 | 82 | 1596 | 11.9 | 1985-86 Boston Celtics | |
| 7 | Bill Walton\waltobi01 | 80 | 1546 | 17 | 1985-86 Boston Celtics | |
| 8 | Scott Wedman\wedmasc01 | 79 | 1402 | 12.5 | 1985-86 Boston Celtics | |
| 9 | Rick Carlisle\carliri01 | 77 | 760 | 7.9 | 1985-86 Boston Celtics | |
| 10 | Greg Kite\kitegr01 | 64 | 464 | 4.6 | 1985-86 Boston Celtics | |
| 11 | Sam Vincent\vincesa01 | 57 | 432 | 10 | 1985-86 Boston Celtics | |
| 12 | David Thirdkill\thirdda01 | 49 | 385 | 12.1 | 1985-86 Boston Celtics | |
| 1 | Rafer Alston\alstora01 | 82 | 3040 | 12.9 | 2006-07 Houston Rockets | |
| 2 | Shane Battier\battish01 | 82 | 2988 | 12 | 2006-07 Houston Rockets | |
| 3 | Tracy McGrady\mcgratr01 | 71 | 2539 | 23.2 | 2006-07 Houston Rockets | |
| 4 | Luther Head\headlu01 | 80 | 2211 | 13.7 | 2006-07 Houston Rockets | |
| 5 | Juwan Howard\howarju01 | 80 | 2123 | 12.8 | 2006-07 Houston Rockets | |
| 6 | Chuck Hayes\hayesch01 | 78 | 1714 | 13.9 | 2006-07 Houston Rockets | |
| 7 | Yao Ming\mingya01 | 48 | 1624 | 26.5 | 2006-07 Houston Rockets | |
| 8 | Dikembe Mutombo\mutomdi01 | 75 | 1289 | 14.4 | 2006-07 Houston Rockets | |
| 9 | Bonzi Wells\wellsbo01 | 28 | 590 | 8.9 | 2006-07 Houston Rockets | |
| 10 | Kirk Snyder\snydeki01 | 39 | 563 | 12.2 | 2006-07 Houston Rockets | |
| 11 | John Lucas III\lucasjo02 | 47 | 383 | 11.8 | 2006-07 Houston Rockets | |
| 12 | Vassilis Spanoulis\spanova01 | 31 | 272 | 5.4 | 2006-07 Houston Rockets | |
| 13 | Scott Padgett\padgesc01 | 24 | 198 | 5.6 | 2006-07 Houston Rockets | |
| 14 | Steve Novak\novakst01 | 35 | 191 | 7.7 | 2006-07 Houston Rockets | |
| 15 | Jake Tsakalidis\tsakaja01 | 13 | 132 | 12 | 2006-07 Houston Rockets | |
| 1 | Magic Johnson\johnsma02 | 80 | 2904 | 27 | 1986-87 Los Angeles Lakers | |
| 2 | James Worthy\worthja01 | 82 | 2819 | 18.4 | 1986-87 Los Angeles Lakers | |
| 3 | Byron Scott\scottby01 | 82 | 2729 | 16 | 1986-87 Los Angeles Lakers | |
| 4 | Kareem Abdul-Jabbar\abdulka01 | 78 | 2441 | 17.9 | 1986-87 Los Angeles Lakers | |
| 5 | Michael Cooper\coopemi01 | 82 | 2253 | 14 | 1986-87 Los Angeles Lakers | |
| 6 | A.C. Green\greenac01 | 79 | 2240 | 15.7 | 1986-87 Los Angeles Lakers | |
| 7 | Kurt Rambis\rambiku01 | 78 | 1514 | 12.6 | 1986-87 Los Angeles Lakers | |
| 8 | Billy Thompson\thompbi01 | 59 | 762 | 12.8 | 1986-87 Los Angeles Lakers | |
| 9 | Mychal Thompson\thompmy01 | 33 | 680 | 12.6 | 1986-87 Los Angeles Lakers | |
| 10 | Wes Matthews\matthwe01 | 50 | 532 | 11.7 | 1986-87 Los Angeles Lakers | |
| 11 | Frank Brickowski\brickfr01 | 37 | 404 | 10.2 | 1986-87 Los Angeles Lakers | |
| 12 | Mike Smrek\smrekmi01 | 35 | 233 | 5 | 1986-87 Los Angeles Lakers | |
| 1 | Joe Johnson\johnsjo02 | 82 | 3240 | 15.1 | 2004-05 Phoenix Suns | |
| 2 | Shawn Marion\mariosh01 | 81 | 3146 | 21.7 | 2004-05 Phoenix Suns | |
| 3 | Amar'e Stoudemire\stoudam01 | 80 | 2889 | 26.6 | 2004-05 Phoenix Suns | |
| 4 | Quentin Richardson\richaqu01 | 79 | 2839 | 13.6 | 2004-05 Phoenix Suns | |
| 5 | Steve Nash\nashst01 | 75 | 2573 | 22 | 2004-05 Phoenix Suns | |
| 6 | Leandro Barbosa\barbole01 | 63 | 1087 | 12.8 | 2004-05 Phoenix Suns | |
| 7 | Steven Hunter\huntest01 | 76 | 1046 | 14.7 | 2004-05 Phoenix Suns | |
| 8 | Jim Jackson\jacksji01 | 40 | 997 | 11.4 | 2004-05 Phoenix Suns | |
| 9 | Casey Jacobsen\jacobca01 | 40 | 768 | 8.4 | 2004-05 Phoenix Suns | |
| 10 | Jake Voskuhl\voskuja01 | 38 | 360 | 8.7 | 2004-05 Phoenix Suns | |
| 11 | Walter McCarty\mccarwa01 | 28 | 353 | 6.9 | 2004-05 Phoenix Suns | |
| 12 | Bo Outlaw\outlabo01 | 39 | 214 | 8.1 | 2004-05 Phoenix Suns | |
| 13 | Maciej Lampe\lampema01 | 16 | 119 | 5 | 2004-05 Phoenix Suns | |
| 14 | Jackson Vroman\vromaja01 | 10 | 57 | 8.2 | 2004-05 Phoenix Suns | |
| 15 | Smush Parker\parkesm01 | 5 | 34 | 4.3 | 2004-05 Phoenix Suns | |
| 1 | Allen Iverson\iversal01 | 82 | 3424 | 20.9 | 2007-08 Denver Nuggets | |
| 2 | Carmelo Anthony\anthoca01 | 77 | 2806 | 21.1 | 2007-08 Denver Nuggets | |
| 3 | Marcus Camby\cambyma01 | 79 | 2758 | 17.2 | 2007-08 Denver Nuggets | |
| 4 | Kenyon Martin\martike01 | 71 | 2159 | 14.7 | 2007-08 Denver Nuggets | |
| 5 | Anthony Carter\cartean01 | 70 | 1960 | 12.8 | 2007-08 Denver Nuggets | |
| 6 | Linas Kleiza\kleizli01 | 79 | 1889 | 14.4 | 2007-08 Denver Nuggets | |
| 7 | Eduardo Nájera\najered01 | 78 | 1664 | 12.1 | 2007-08 Denver Nuggets | |
| 8 | J.R. Smith\smithjr01 | 74 | 1421 | 18.1 | 2007-08 Denver Nuggets | |
| 9 | Yakhouba Diawara\diawaya01 | 54 | 542 | 8.1 | 2007-08 Denver Nuggets | |
| 10 | Chucky Atkins\atkinch01 | 24 | 352 | 8.9 | 2007-08 Denver Nuggets | |
| 11 | Nenê Hilário\hilarne01 | 16 | 266 | 11.1 | 2007-08 Denver Nuggets | |
| 12 | Bobby Jones\jonesbo02 | 25 | 222 | 8.5 | 2007-08 Denver Nuggets | |
| 13 | Mike Wilks\wilksmi01 | 8 | 122 | 6.4 | 2007-08 Denver Nuggets | |
| 14 | Steven Hunter\huntest01 | 19 | 120 | 8.6 | 2007-08 Denver Nuggets | |
| 15 | Von Wafer\wafervo01 | 21 | 90 | -1.6 | 2007-08 Denver Nuggets | |
| 16 | Jelani McCoy\mccoyje01 | 6 | 33 | 2.6 | 2007-08 Denver Nuggets | |
| 17 | Taurean Green\greenta01 | 9 | 30 | 8.8 | 2007-08 Denver Nuggets | |
| 1 | Ben Wallace\wallabe01 | 81 | 3050 | 17.3 | 2003-04 Detroit Pistons | |
| 2 | Richard Hamilton\hamilri01 | 78 | 2772 | 16.8 | 2003-04 Detroit Pistons | |
| 3 | Chauncey Billups\billuch01 | 78 | 2758 | 18.6 | 2003-04 Detroit Pistons | |
| 4 | Tayshaun Prince\princta01 | 82 | 2701 | 13.3 | 2003-04 Detroit Pistons | |
| 5 | Mehmet Okur\okurme01 | 71 | 1580 | 18.3 | 2003-04 Detroit Pistons | |
| 6 | Corliss Williamson\willico02 | 79 | 1574 | 14.4 | 2003-04 Detroit Pistons | |
| 7 | Elden Campbell\campbel01 | 65 | 892 | 13.8 | 2003-04 Detroit Pistons | |
| 8 | Chucky Atkins\atkinch01 | 40 | 751 | 10.1 | 2003-04 Detroit Pistons | |
| 9 | Bob Sura\surabo01 | 53 | 707 | 13 | 2003-04 Detroit Pistons | |
| 10 | Rasheed Wallace\wallara01 | 22 | 673 | 18.8 | 2003-04 Detroit Pistons | |
| 11 | Lindsey Hunter\hunteli01 | 33 | 661 | 9 | 2003-04 Detroit Pistons | |
| 12 | Mike James\jamesmi01 | 26 | 512 | 14.3 | 2003-04 Detroit Pistons | |
| 13 | Darvin Ham\hamda01 | 54 | 484 | 8.3 | 2003-04 Detroit Pistons | |
| 14 | Tremaine Fowlkes\fowlktr01 | 36 | 261 | 6.5 | 2003-04 Detroit Pistons | |
| 15 | Željko Rebrača\rebraze01 | 21 | 222 | 9.6 | 2003-04 Detroit Pistons | |
| 16 | Darko Miličić\milicda01 | 34 | 159 | 6.1 | 2003-04 Detroit Pistons | |
| 1 | LeBron James\jamesle01 | 76 | 2709 | 27.5 | 2015-16 Cleveland Cavaliers | |
| 2 | Kevin Love\loveke01 | 77 | 2424 | 19 | 2015-16 Cleveland Cavaliers | |
| 3 | J.R. Smith\smithjr01 | 77 | 2362 | 12.4 | 2015-16 Cleveland Cavaliers | |
| 4 | Tristan Thompson\thomptr01 | 82 | 2269 | 15.9 | 2015-16 Cleveland Cavaliers | |
| 5 | Matthew Dellavedova\dellama01 | 76 | 1867 | 11.3 | 2015-16 Cleveland Cavaliers | |
| 6 | Kyrie Irving\irvinky01 | 53 | 1667 | 19.9 | 2015-16 Cleveland Cavaliers | |
| 7 | Richard Jefferson\jefferi01 | 74 | 1326 | 9.7 | 2015-16 Cleveland Cavaliers | |
| 8 | Timofey Mozgov\mozgoti01 | 76 | 1326 | 14.6 | 2015-16 Cleveland Cavaliers | |
| 9 | Iman Shumpert\shumpim01 | 54 | 1316 | 8.4 | 2015-16 Cleveland Cavaliers | |
| 10 | Mo Williams\willima01 | 41 | 748 | 12.4 | 2015-16 Cleveland Cavaliers | |
| 11 | James Jones\jonesja02 | 48 | 463 | 11.6 | 2015-16 Cleveland Cavaliers | |
| 12 | Channing Frye\fryech01 | 26 | 446 | 14.9 | 2015-16 Cleveland Cavaliers | |
| 13 | Jared Cunningham\cunnija01 | 40 | 355 | 5.9 | 2015-16 Cleveland Cavaliers | |
| 14 | Anderson Varejão\varejan01 | 31 | 310 | 11.2 | 2015-16 Cleveland Cavaliers | |
| 15 | Jordan McRae\mcraejo01 | 15 | 113 | 14.2 | 2015-16 Cleveland Cavaliers | |
| 16 | Sasha Kaun\kaunsa01 | 25 | 95 | 12.5 | 2015-16 Cleveland Cavaliers | |
| 17 | Dahntay Jones\jonesda02 | 1 | 42 | 10.3 | 2015-16 Cleveland Cavaliers | |
| 18 | Joe Harris\harrijo01 | 5 | 15 | 3.4 | 2015-16 Cleveland Cavaliers | |
| 1 | Jason Kidd\kiddja01 | 80 | 2653 | 14.4 | 2010-11 Dallas Mavericks | |
| 2 | Jason Terry\terryja01 | 82 | 2564 | 15.9 | 2010-11 Dallas Mavericks | |
| 3 | Dirk Nowitzki\nowitdi01 | 73 | 2504 | 23.4 | 2010-11 Dallas Mavericks | |
| 4 | Shawn Marion\mariosh01 | 80 | 2253 | 17 | 2010-11 Dallas Mavericks | |
| 5 | Tyson Chandler\chandty01 | 74 | 2059 | 18.4 | 2010-11 Dallas Mavericks | |
| 6 | J.J. Barea\bareajo01 | 81 | 1669 | 14.8 | 2010-11 Dallas Mavericks | |
| 7 | Brendan Haywood\haywobr01 | 72 | 1331 | 11.7 | 2010-11 Dallas Mavericks | |
| 8 | DeShawn Stevenson\stevede01 | 72 | 1158 | 9.8 | 2010-11 Dallas Mavericks | |
| 9 | Caron Butler\butleca01 | 29 | 867 | 14.2 | 2010-11 Dallas Mavericks | |
| 10 | Brian Cardinal\cardibr01 | 56 | 618 | 9.7 | 2010-11 Dallas Mavericks | |
| 11 | Peja Stojaković\stojape01 | 25 | 506 | 14.1 | 2010-11 Dallas Mavericks | |
| 12 | Rodrigue Beaubois\beaubro01 | 28 | 496 | 11.4 | 2010-11 Dallas Mavericks | |
| 13 | Ian Mahinmi\mahinia01 | 56 | 488 | 13.7 | 2010-11 Dallas Mavericks | |
| 14 | Sasha Pavlović\pavloal01 | 10 | 163 | 7.8 | 2010-11 Dallas Mavericks | |
| 15 | Corey Brewer\breweco01 | 13 | 148 | 17.6 | 2010-11 Dallas Mavericks | |
| 16 | Dominique Jones\jonesdo02 | 18 | 135 | 10.6 | 2010-11 Dallas Mavericks | |

Team Elo:

|  |  |
| --- | --- |
| Team | Elo |
| 1997-98 Utah Jazz | 1766 |
| 1970-71 Los Angeles Lakers | 1753 |
| 2004-05 San Antonio Spurs | 1771 |
| 2016-17 Golden State Warriors | 1865 |
| 2012-13 Miami Heat | 1774 |
| 1995-96 Chicago Bulls | 1853 |
| 2000-01 Los Angeles Lakers | 1779 |
| 2007-08 Boston Celtics | 1760 |
| 1995-96 Seattle SuperSonics | 1731 |
| 1985-86 Boston Celtics | 1816 |
| 2006-07 Houston Rockets | 1697 |
| 1986-87 Los Angeles Lakers | 1760 |
| 2004-05 Phoenix Suns | 1743 |
| 2007-08 Denver Nuggets | 1691 |
| 2003-04 Detroit Pistons | 1715 |
| 2015-16 Cleveland Cavaliers | 1759 |
| 2010-11 Dallas Mavericks | 1736 |