

데이터구조설계

Project 2

제출일자: 2017년 11월 10일 (금)

학 과: 컴퓨터공학과

담당교수: 이기훈 교수님

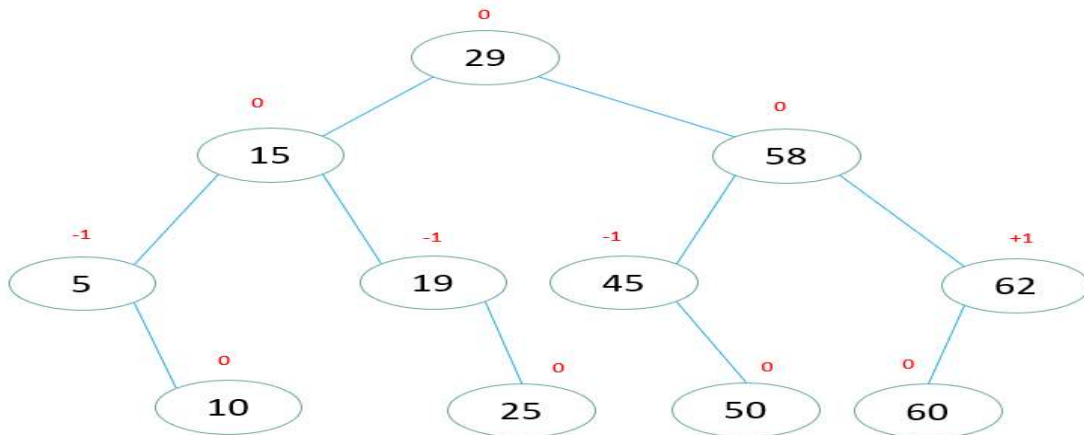
학 번: 2014722075

성 명: 이동준

O Introduction

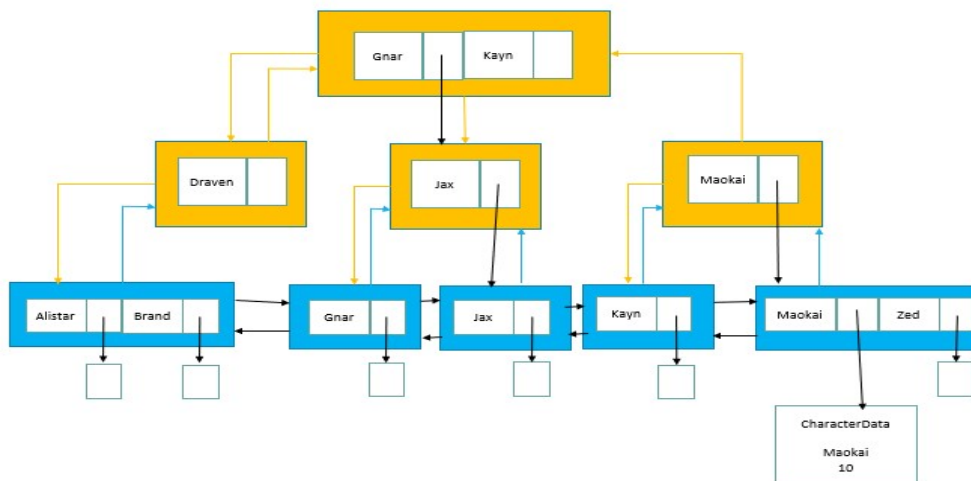
위 프로그램은 NoL 게임 정보 관리 시스템으로 각 사용자(id)들의 게임 정보를 저장하고, 효율적으로 관리할 수 있다. 게임 정보에는 Game ID, Duration, Winner, Characters of team A, team B 가 있다. Winner 가 0 이면 A 팀의 승리, 1 이면 B 팀의 승리이다. 각 사용자들의 게임 정보는 AVL Tree 를 이용하고, 각 게임에 사용된 캐릭터 정보는 B+-Tree 를 통해 관리한다. AVL Tree 는 Game ID 를 기준으로 설계하였고, B+-Tree 는 캐릭터의 이름을 기준으로 설계하였다.

* AVL Tree



- AVL Tree 는 BST 의 단점을 보충하기 위해 설계되었으며, 한 노드의 오른쪽 subtree 와 왼쪽 subtree 의 level 의 차이가 -1 이상 1 이하로 되도록 설정된 Tree 이다. 기본 조건은 BST 와 똑같이 노드의 data 에 의해 구성된다. 각 노드들은 balance factor 를 가지고 있으며 그 balance 가 2 이상 -2 이하가 되면 LL,LR 등의 방법으로 새롭게 balance 를 맞춘다. 위 프로그램에서는 한 node 안에 각 사용자가 play 한 하나의 게임 정보가 들어있다.

* B+ Tree



- B+ Tree 는 인덱스 노드와 데이터 노드로 구성되어 있으며, 각 노드 클래스는 B+ Tree 노드 클래스(BpTreeNode)를 상속받는다. 데이터 노드는 각 캐릭터들의 객체 정보를 map 컨테이너 형태로 저장되고, 인덱스 노드는 자식을 가리킬 포인터가 있는 map 형태로 저장된다. B+ Tree 는 order(한 노드에 들어갈 수 있는 맵의 개수를 구할 수 있음)를 가지고 있으며 그 수는 고정되어 있지 않다. B+ Tree 는 캐릭터 이름을 기준으로 캐릭터 정보를 저장한다. 캐릭터 data 를 입력하는 과정에서 처음 입력된 캐릭터는 새로운 객체를 생성하여 삽입하며, 이전에 참여했던 캐릭터가 다시 참여하는 경우는, 해당 객체의 Cpick 을 업데이트하고 상황에따라 Cwin 도 업데이트 한다.

* Heap

- Max Heap – 캐릭터 선택 횟수를 기준으로 캐릭터 정보를 삽입하고 출력한다.(내림차순)
- Min Heap – 캐릭터의 승률을 기준으로 캐릭터 정보를 삽입하고 출력한다.(오름차순)

* Command(command_list.txt) – 각 명령어가 입력되어 있다.

- LOAD

: NoL 게임 정보를 저장하는 명령어로, game_load.txt 를 읽어 각 Tree 를 구성한다.

Game ID Duration Winner Team A(5) Team B(5)순으로 입력되어 있으며 각 사용자들의 게임정보를 가지고 있다. ID 의 범위는 1~1000, Duration 의 범위는 1~10000 로 한정하며 winner 는 0 또는 1 뿐이다. 텍스트 파일이 존재하지 않거나 이미 LOAD 가 완료되어 있는 경우에는 에러 코드(100)을 출력한다.

- UPDATE

: 게임 정보를 수정하는 명령어로, game_update.txt.를 읽어 각 Tree 를 수정한다.

Duration 과 win, 게임에 참여한 캐릭터 이름만 수정하며, ID 는 수정되지 않는다. LOAD 와 마찬가지로 범위 밖의 입력은 없다고 가정한다. 만약 게임에 참여한 횟수가 0 이 되는 캐릭터가 있더라고 B+ Tree 에서 삭제하지 않는다. 텍스트 파일이 존재하지 않거나 입력되지 않은 ID 를 UPDATE 하는 경우, 에러코드(200)을 출력한다.

- SEARCH

: GAME 이나 CHARACTER 를 기반으로 정보를 검색하는 명령어.

Search_game 같은 경우에는 입력한 ID 를 기반으로 AVL Tree 에서 검색하여 반환한다.

Search_character 같은 경우에는 두 개의 캐릭터이름을 입력 받아 두 캐릭터 사이의 모든 캐릭터를 오름차순으로 출력한다.

만약 입력한 ID 정보가 존재하지 않는 경우 에러코드(300)을 출력한다.

입력한 캐릭터 중 하나라도 정보가 존재하지 않는 경우 에러코드(400)을 출력한다.

- PRINT

: game, character, pick, winratio 를 기준으로 정보들을 출력하는 명령어

print_game 은 AVL Tree 에 구성되어 있는 ID 기준의 모든 정보들을 오름차순(ID)으로 출력한다.

만약 AVL Tree 가 구성되어 있지 않은 경우, 에러코드(500)을 출력한다.

print_character 는 B+ Tree 에 구성되어 있는 character 기준의 모든 정보들을 오름차순(캐릭터이름)으로 출력한다.

만약 B+ Tree 가 구성되어 있지 않은 경우, 에러코드(600)을 출력한다.

print_pick 은 B+ Tree 를 탐색하여 캐릭터의 게임참여 횟수를 기준으로 내림차순(Max-heap)으로 출력한다.

만약 Heap 에 저장되어 있는 정보가 없을 경우, 에러코드(700)을 출력한다.

print_winaratio 는 B+ Tree 를 탐색하여 캐릭터의 승률을 기준으로 오름차순(Min-heap)으로 출력한다.

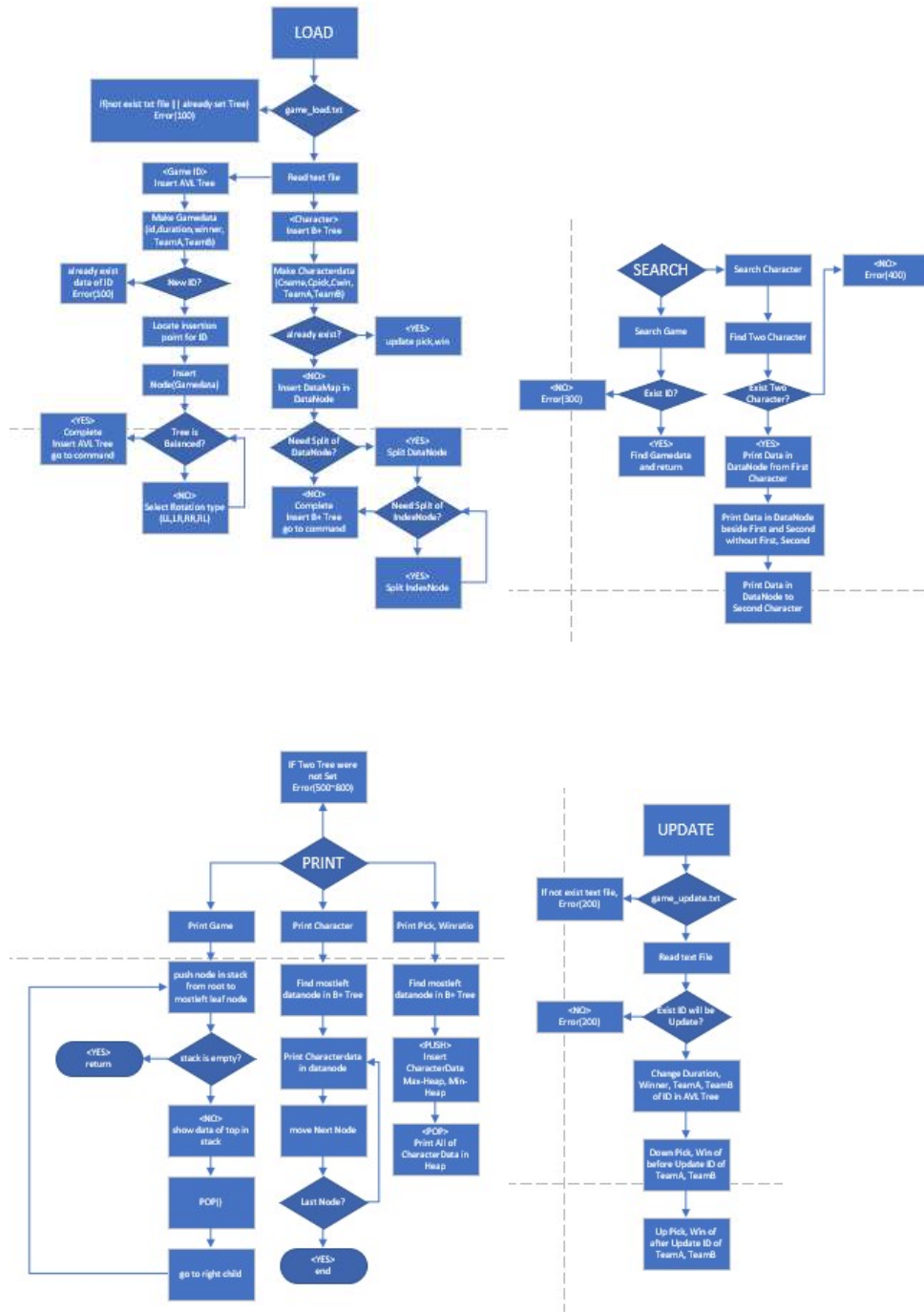
만약 Heap 에 저장되어 있는 정보가 없을 경우, 에러코드(800)을 출력한다.

- EXIT

: 프로그램을 종료하는 명령어

각 명령어들이 완료됨에 따라 결과(+에러)를 log.txt.에 출력한다.

☐ Flowchart



O Algorithm

#Manager(run)

file.open(command text file)

if(LOAD)

file.open(load text file)

if(not exist text file)

error(100)

else

AVL Tree = GameData

if(already exist GameData)

error(100)

B+ Tree = CharacterData

load complete

else if(SEARCH)

if(GAME)

find data of ID

if(not exist ID)

error(300)

else

fout<<GameData of ID<<;

else if(CHARACTER)

find data of Characters

if(not exist first || not exist second)

error(400)

else

```

                                fout<<CharacterData from first character to second<<;
else if(PRINT)
    if(GAME)
        if(not set AVL Tree)
            error(500)
        else
            fout<<all of GameData<<;
    else if(PICK)
        if(not set Heap)
            error(700)
        else
            fout<<Pick Rank of CharacterData<<;
    else if(CCHARACTER)
        if(not set B+ Tree)
            error(600)
        else
            fout<<all of CharacterData<<;
    else if(WINRATIO)
        if(not set Heap)
            error(800)
        else
            fout<<Winratio Rank of CharacterData<<;
else if(UPDATE)
    file.open(update text file
    if(not exist update text file)

```

```
error(200)
```

```
else
```

```
if(ID will be updated is not exist)
```

```
error(200)
```

```
else
```

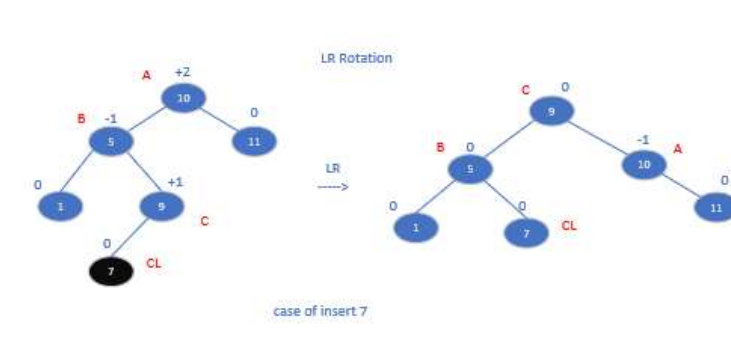
```
update GameData of ID in AVL Tree
```

```
update Character of GameData in B+ Tree
```

```
else if(EXIT)
```

```
end Program
```

```
# AVL Tree
```



```
- Insert(pGame)
```

```
if(root == NULL)
```

```
root = pGame
```

```
else
```

```
find insertion point for ID of pGame
```

```
make new AVLNode(pGame)
```

```
insert node in insertion point for ID
```

```
reset balance factor from root
```

```
while(Tree is unbalanced)
```

```
find position not unbalanced
```



```
    decide rotation type  
    if(type is 'LL')  
        do LL rotation  
    if(type is 'LR')  
        do LR rotation  
    if(type is 'RR')  
        do RR rotation  
    if(type is 'RL')  
        do RL rotation  
    if(Tree is balanced)  
        break;
```

insert complete

- Print

```
d = root;
```

```
while(d != NULL)
```

```
    push d in stack
```

```
    d = Leftchild of d
```

```
if(stack == empty)
```

```
    return;
```

```
d = top data of stack
```

```
delete top data of stack(pop)
```

```
print Gamedata of d
```

```
d = Rightchild of d
```

```

- Search(ID)

if(root == gamedata of ID)
    return gamedata of root

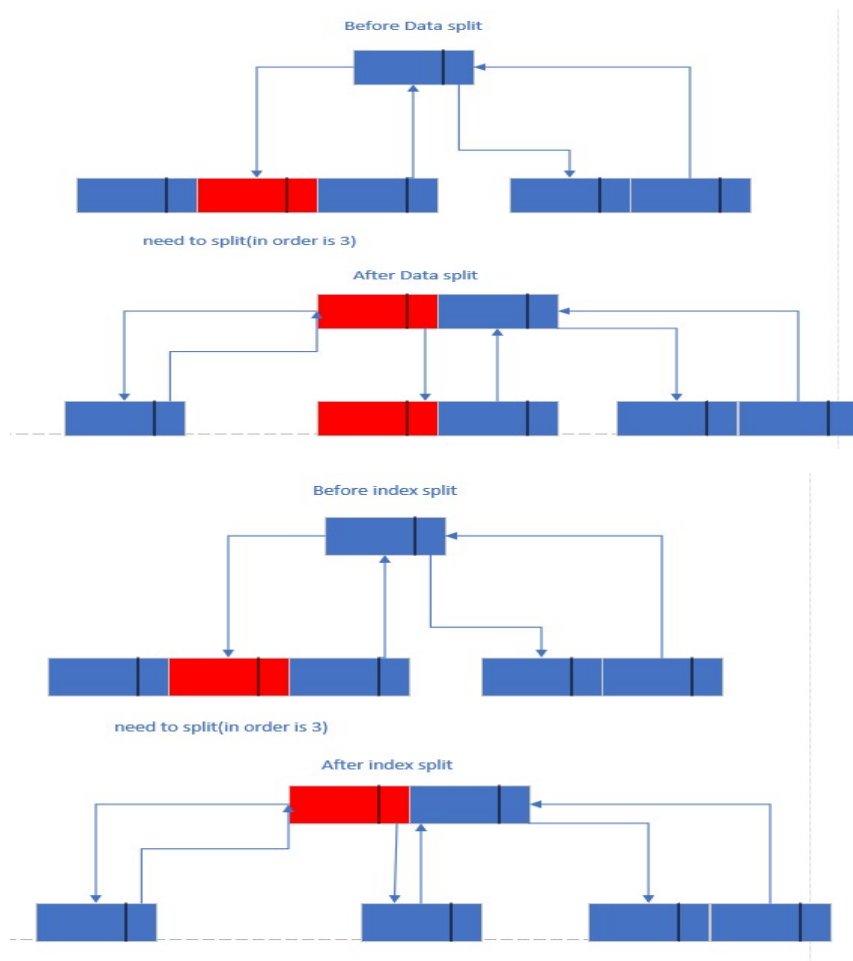
find node want to search

if (find)
    return gamedata of ID found

if(cant find)
    return NULL;

```

B+ Tree



```

- Insert(CharacterData)

```

```

if(exist CharacterData)
    update pick,win of character
    return false;

if(root == NULL)
    make new BpTreeNode
    insert data map of Character in new Node
    root = new Node
    return true;

find node can insert data map of character
after found insert data map
if(found node need to split)
    splitDataNode(found node)
else
    return true;

- SplitDataNode(pNode)
find map of pNode will be split
make new node(after split, it will be left node)
if(no parent, only root)
    make index node
    insert map
    set new link
    root = new index node
if(have parent)
    if(pNode is mostleft of parent)

```

```
        insert map in parent node

        set new link(must be change mostleft)

    else(the others case    )

        insert map in parent node

        set new link

    if(parent node need to split)

        splitIndexNode(parent)
```

```
- SplitIndexNode(pNode)

    find index map will be split

    make new node

    set new link for new node

    if(parent == NULL)

        make new parent node

        insert index map in parent

        set new link

        root = parent

    else if(pNode is mostleft of parent)

        set new link(mostleft child will be changed)

        insert map in parent

        delete map in pNode

    else if(pNode is not mostleft of parent)

        set new link

        insert map in parent

        delete map in pNode
```

if(parent node need to split)

splitIndexNode(parent)

- SearchRange(start, end)

find dataNode1 had map of start

find dataNode2 had map of end

fout<<CharacterData in dataNode 1 from start

fout<<CharacterData in dataNode between start and end

fout<<CharacterData in dataNode2 to end

- Print()

Cur = find mostleft DataNode in B+ Tree

fout<<CharacterData in B+ Tree from Cur to mostright DataNode in B+ Tree

- printPickRank() – using max-Heap

while(visit all of dataNode)

push data in heap (data will be sorted according to Pick)

while(heap == empty)

fout<<top data of heap<<;

pop();

- printRatioRank() – using min-Heap

while(visit all of dataNode)

push data in heap(data will be sorted according to winratio)

while(heap == empty)

fout<<top data of heap<<;

pop()

O Result Screen

* LOAD

ERROR														
100														
LOAD														
game_load.txt														

LOAD 실행 후 결과화면이다. 만약 텍스트 파일이 존재하지 않거나 이미 트리가 구성되어 있는 경우 위의 그림처럼 ERROR 를 출력하게 되고 정상적으로 LOAD 가 완료되면 LOAD 를 실행한 text 파일 이름과 같이 LOAD 를 출력하게 된다.

LOAD 명령어는 text 파일에 있는 게임 정보들을 받아와 AVL Tree 와 B+ Tree 를 구성한다.

* UPDATE

ERROR															
200															
UPDATE															
Gameid	Duration	Winner		TeamA_c1	Team_c2	Team_c3	Team_c4	Team_c5			TeamB_c1	TeamB_c2	TeamB_c3	TeamB_c4	TeamB_c5
82	1934	1	Jarvan IV	Nidalee	Lux	Alistar	Jhin	Syndra	Maokai	Master Yi	Tristana	Tryndamere			
27	1638	0	Zed	Maokai	Kha'Zix	Zilean	Twitch	Caitlyn	Sona	Urgot	Lux	Ivern			

UPDATE 실행 후 결과화면이다. 만약 텍스트 파일이 존재하지 않거나 입력한 게임 ID 가 존재하지 않을 경우 위와 같이 ERROR 를 출력하게 되고 정상적으로 UPDATE 가 완료된 게임데이터들을 출력한다. 만약 UPDATE 과정에서 하나라도 잘못된 ID 가 입력되었을 경우에는 UPDATE 를 종료한다.

SEARCH 27														
Gameid : 27														
Duration : 2472														
Winner : 1														
Team A : Vayne/Viktor/Janna/Fiddlesticks/Akali														
Team B : Singed/Kalista/Orianna/Thresh/Jax														
SEARCH 27														
Gameid : 27														
Duration : 1638														
Winner : 0														
Team A : Zed/Maokai/Kha'Zix/Zilean/Twitch														
Team B : Caitlyn/Sona/Urgot/Lux/Ivern														

위 두 그림은 ID 27 에 대한 업데이트 전 후의 그림이다 다음과 같이 Duration, Winner, TeamA, TeamB 에 있어서 Update 를 진행한다.

* SEARCH_GAME

ERROR	
300	
SEARCH 50	
Gameid : 50	
Duration : 2237	
Winner : 1	
Team A : Ashe/Thresh/Maokai/Syndra/Yorick	
Team B : Renekton/Orianna/Warwick/Blitzcrank/Caitlyn	

ID 를 기반으로 AVL Tree 에서 탐색하여 그 ID 의 GameData 를 출력한다. 만약 입력한 게임 ID 가 존재하지 않을 경우에는 ERROR 를 출력하게 된다.

* SEARCH_CHARACTER

SEARCH Nami/Pantheon			
Name	Pick	Win	WinRatio
Nami	7	4	57.1429
Nasus	11	6	54.5455
Nautilus		6	2 33.3333
Nidalee	5	1	20
Nocturne		2	1 50
Nunu	2	2	100
Olaf	2	0	0
Orianna	13	7	53.8462
Ornn	1	0	0
Pantheon		2	0 0

두 개의 캐릭터를 입력 받아 B+ Tree 를 탐색하여 Start 캐릭터와 end 캐릭터사이에 characterData 를 출력한다. 만약 입력한 두 캐릭터 중 하나라도 data 가 존재하지 않는 경우, ERROR 를 출력한다.

* PRINT_GAME

PRINT GAME												
Gameid	Duration	Winner	TeamA_c1	Team_c2	Team_c3	Team_c4	Team_c5	TeamB_c1	TeamB_c2	TeamB_c3	TeamB_c4	TeamB_c5
1	1949	0	Vladimir	Bard	Kog'Maw	Master Yi	Viktor	Graves	Xayah	Darius	Zed	Thresh
2	1851	0	Draven	Irelia	Nidalee	Kayle	Shaco	Malphite	Morgana	Hecarim	Yasuo	Riven
3	1493	0	Tristana	Kayn	Nami	Rumble	Kassadin	Cassiopeia	Thresh	Jayce	Jax	Ashe
4	1758	0	Maokai	Brand	Twitch	Orianna	Dr. Mundo	Malzahar	Warwick	Thresh	Riven	Ashe
5	2094	0	Warwick	Twitch	Janna	Draven	Syndra	Sona	Jarvan IV	Kayn	Kassadin	Caitlyn
6	2059	0	Janna	Kayn	Jax	Veigar	Vayne	Riven	Sivir	EKKO	Olaf	Alistar
7	1993	0	Heimerdinger	Teemo	Thresh	Tristana	Maokai	Blitzcrank	Miss Fortune	Fiora	Vel'Koz	Warwick
8	1334	0	Gnar	Xayah	Soraka	Malzahar	Gragas	Viktor	Lee Sin	Rakan	Vayne	Maokai
9	1387	1	Nautilus	Maokai	Tristana	Nami	Vel'Koz	Twitch	Jarvan IV	Lux	Kayn	Brand
10	2681	1	Ivern	Master Yi	Nasus	Kalista	Diana	Shaco	Varus	Heimerdinger	Lux	Garen
11	1391	1	Lux	Lucian	Zyra	Maokai	Nautilus	LeBlanc	Kayle	Fizz	Tristana	Leona
12	1671	0	Ashe	Twisted Fate	Rammus	Yasuo	Blitzcrank	Bard	Fiora	Amumu	Xerath	Miss Fortune
13	2071	0	Gragas	Fizz	Garen	Sona	Ashe	Sivir	Janna	Zed	Master Yi	Xin Zhao
14	1942	0	Zyra	LeBlanc	Rek'Sai	Jax	Caitlyn	Annie	Lee Sin	Jhin	Ezreal	Akali
15	2374	1	Janna	Ahri	Xayah	Riven	Kayn	Lux	Lee Sin	Syndra	Jax	Twitch
16	1717	0	Tryndamere	Katarina	Tristana	Amumu	Soraka	Janna	Lee Sin	Sivir	Ahri	Gnar
17	2390	1	Vayne	Janna	Azir	Xin Zhao	Riven	Fizz	Sejuani	Jhin	Soraka	Mordekaiser
18	1994	0	Fiora	Ezreal	Janna	Draven	EKKO	Cho'Gath	Sona	Tryndamere	Orianna	Jinx
19	1513	1	Jinx	Lee Sin	Taric	EKKO	Irelia	Swain	Teemo	Brand	Jarvan IV	Lucian
20	1898	1	Rek'Sai	Corki	Jarvan IV	Tristana	Rakan	Draven	Nami	Gangplank	Kha'Zix	Twisted Fate
21	2026	1	Syndra	Dr. Mundo	Amumu	Leona	Draven	Lulu	Vayne	Graves	Zed	Teemo
22	1962	1	Zed	Janna	Gragas	Twitch	Maokai	Diana	Garen	Jhin	Taric	Heimerdinger
23	1511	1	Kog'Maw	Lulu	LeBlanc	Graves	Urgot	Tryndamere	Cho'Gath	Jinx	Janna	Twisted Fate
24	1756	0	Darius	Kalista	Thresh	Twisted Fate	Fiddlesticks	Xayah	Janna	Gnar	Orianna	Nocturne
25	2173	1	Xayah	Evelynn	Fizz	Tryndamere	Veigar	Xerath	Gnar	Vayne	Kassadin	Warwick
26	2211	1	Gangplank	Tryndamere	Jinx	Elise	Thresh	Taliyah	Darius	Twitch	Vayne	Blitzcrank
27	1638	0	Zed	Maokai	Kha'Zix	Zilean	Twitch	Caitlyn	Sona	Urgot	Lux	Ivern
28	1342	1	Warwick	Cho'Gath	Ashe	Leona	Veigar	Lulu	Caitlyn	Gnar	Ahri	Twitch
29	2288	1	Poppy	Yasuo	Sona	Caitlyn	Nidalee	Camille	Bard	Udyr	Swain	Kog'Maw
30	1899	0	Thresh	Miss Fortune	Gnar	Malzahar	Shaco	Gragas	Kalista	Kayn	Leona	Garen
31	1597	1	Vladimir	Leona	Twitch	Graves	Yasuo	Talon	Jarvan IV	Lulu	Ashe	LeBlanc
32	1649	0	Rakan	Akali	Warwick	Tristana	Malzahar	Xerath	Garen	Jarvan IV	Kog'Maw	Lulu
33	1894	0	Jax	Tristana	Yasuo	Soraka	Rammus	Janna	Riven	Malzahar	Lee Sin	Kalista
34	2473	0	Vayne	Thresh	Maokai	Vel'Koz	Sion	Galio	Xin Zhao	Braum	Varus	Darius
35	2452	1	Thresh	Ezreal	Wukong	Brand	Dr. Mundo	Nasus	Zed	Tristana	Gragas	Fiddlesticks
36	1478	1	Yasuo	Maokai	Miss Fortune	Syndra	Alistar	Twisted Fate	Taric	Jhin	Nasus	Rammus
37	1505	1	Morgana	Olaf	Kayle	Caitlyn	Ryze	Xin Zhao	Veigar	Yorick	Varus	Leona
38	2112	0	Shaco	Sona	Twitch	Brand	Akali	Vayne	Fiora	Udyr	Taric	Lux
39	2054	0	Kled	Braum	Orianna	Xayah	Vi	Miss Fortune	Kayn	Tahm Kench	Veigar	Lee Sin
40	1349	0	Jax	Kog'Maw	Xin Zhao	Thresh	Ahri	Trundle	Zac	Zyra	Tristana	EKKO
41	2309	1	Karma	Lucian	Anivia	Sion	Elise	Gragas	Sona	Varus	Veigar	Gnar
42	1329	1	Riven	Draven	Sona	Cho'Gath	Rengar	Morgana	Urgot	Ezreal	Twitch	EKKO
43	1800	0	Kennen	Gragas	Zed	Jhin	Rengar	Zyra	Talon	Cassiopeia	Nidalee	Lucian
44	932	1	Jinx	Thresh	Cassiopeia	Amumu	Gangplank	Soraka	Lucian	Elise	Sivir	Tryndamere
45	2166	1	Shaco	Teemo	Annie	Lulu	Jinx	Rakan	Twitch	Swain	Orianna	Vi
46	1994	0	Gnar	Rakan	Lee Sin	Orianna	Jinx	Jax	Lucian	Ryze	Zyra	Lissandra
47	2048	0	Jax	EKKO	Alistar	Jinx	Teemo	Yasuo	Tristana	Fiora	Jarvan IV	Janna
48	1461	1	Gangplank	Malzahar	Soraka	Xayah	Fiddlesticks		Taric	Vayne	Lulu	Darius
49	2302	1	Jayce	Tristana	Bard	Elise	Gangplank	Fiora	Jarvan IV	EKKO	Annie	Miss Fortune
50	2237	1	Ashe	Thresh	Maokai	Syndra	Yorick	Renekton	Orianna	Warwick	Blitzcrank	Caitlyn
51	2555	1	Brand	Trundle	Lee Sin	Ashe	Lux	Ahri	Morgana	Jax	Lulu	Twitch
52	1026	0	Tristana	Akali	Talon	Maokai	Morgana	Yasuo	Jhin	Thresh	Sejuani	Taliyah
53	1303	1	Cassiopeia	Thresh	Zac	Camille	Kalista	Tristana	Syndra	Gnar	Nunu	Alistar
54	1843	0	Tryndamere	Blitzcrank	Jinx	Zed	Jax	Twitch	Riven	Morgana	Lee Sin	Diana
55	1691	1	Rakan	Zed	Kalista	Riven	Dr. Mundo	Alistar	Ezreal	Heimerdinger	Kayn	Veigar
56	1875	0	Zed	Twitch	Veigar	Lee Sin	Camille	Ashe	Rakan	Kayn	Shaco	Lissandra
57	1413	1	Jax	Janna	Katarina	Sivir	Hecarim	Warwick	Xin	Thresh	Nasus	EKKO
58	1682	0	Kayle	Master Yi	Jhin	Zyra	Aurelion Sol	Shen	Taliyah	Thresh	Nautilus	Lucian
59	1640	1	Master Yi	Ashe	Trundle	Lux	Kayle	Caitlyn	Kayn	EKKO	Nautilus	Lulu
60	1925	0	Gnar	Jarvan IV	Jinx	Fizz	Lux	Ornn	Alistar	Heimerdinger	Ahri	Ezreal
61	1816	0	Jax	Leona	Katarina	Tristana	Cassiopeia	Vi	Sona	Gragas	Jhin	Vel'Koz
62	1070	0	Draven	Kled	Gragas	Kassadin	Thresh	Orianna	Alistar	Caitlyn	Xin Zhao	Jax
63	1633	1	Heimerdinger	Lucian	Fizz	Yasuo	Rakan	Kayn	Orianna	Vi	Soraka	Tristana
64	2169	1	Zed	Rammus	Maokai	Ashe	Leona	Anivia	Nautilus	Nocturne	Kled	Tristana
65	197	1	Nautilus	Warwick	Caitlyn	Pantheon	EKKO	Ashe	Gragas	Zyra	Kled	Katarina
66	1261	0	Jax	Kennen	Ezreal	Cho'Gath	Thresh	Heimerdinger	Shen	Diana	Lucian	Orianna
67	2493	0	Miss Fortune	Elise	Galio	Thresh	Irelia	Renekton	Nami	Cho'Gath	EKKO	Kog'Maw
68	2259	0	Illaoi	Fizz	Thresh	Rammus	Xayah	Lucian	Jax	Riven	Braum	Cassiopeia
69	1897	1	Orianna	Gragas	Volibear	Shen	Jhin	Master Yi	Annie	Caitlyn	Nasus	Rakan
70	1723	0	Vayne	Nasus	LeBlanc	Kayn	Lulu	Urgot	Morgana	Jax	Twitch	Syndra
71	1643	1	Nasus	Fiddlesticks	Leona	Viktor	Ashe	Nunu	Ezreal	Maokai	Xayah	Lulu
72	2187	1	Vladimir	Ezreal	Vel'Koz	Nami	Lee Sin	Riven	Jarvan IV	Gangplank	Jhin	Taliyah
73	1590	0	Gragas	Taric	Gnar	Taliyah	Vayne	Jinx	Rumble	Jax	Thresh	Rek'Sai
74	1638	0	Zed	Maokai	Kha'Zix	Zilean	Twitch	Caitlyn	Sona	Urgot	Lux	Ivern
75	2147	1	Kayn	Lucian	Braum	Nasus	Ryze	Warwick	Nami	Morgana	Draven	Gnar
76	1296	0	Rakan	Shen	Xayah	Dr. Mundo	Taliyah	Lulu	Tristana	Quinn	Zyra	Kayn
77	2634	0	Vladimir	Tristana	Braum	Ahri	Vi	Jinx	Malphite	Yasuo	Wukong	Garen
78	963	0	EKKO	Ezreal	Zac	Alistar	Riven	Cho'Gath	Vayne	Brand	Cassiopeia	Viktor
79	2929	1	Ahri	Warwick	Zac	Thresh	Jhin	Tristana	Veigar	Gnar	Sejuani	Malphite
80	2097	0	Fizz	Nami	Jax	Miss Fortune	Dr. Mundo	Lux	Sona	Vi	Mordekaiser	Twitch
81	2051	0	Cassiopeia	Kayle	Hecarim	Vayne	Janna	Lulu	Miss Fortune	Xin Zhao	Talon	Ivern
82	1934	1	Jarvan IV	Nidalee	Lux	Alistar	Jhin	Syndra	Maokai	Master Yi	Tristana	Tryndamere
83	193	0	Brand	Darius	Kha'Zix	Tristana	Lux	Xin Zhao	Volibear	Twitch	Viktor	Leona
84	1934	1	Jarvan IV	Nidalee	Lux	Alistar	Jhin	Syndra	Maokai	Master Yi	Tristana	Tryndamere
85	1451	0	EKKO	Twitch	Fiora	Brand	Gragas	Kog'Maw	Syndra	Karma	Kha'Zix	Pantheon
86	1257	1	Vi	Lulu	Twitch	Rammus	Swain	Jarvan IV	Miss Fortune	Thresh	Syndra	Nasus
87	1789	0	Xin Zhao	Jarvan IV	Vayne	Leona	Cassiopeia	LeBlanc	Malphite	Gnar	Kha'Zix	Jhin
88	1311	0	Sivir	Rakan	Ahri	Darius	Sion	Taric	Kalista	LeBlanc	Garen	Jarvan IV
89	2218	1	Veigar	Fizz	Shen	Maokai	Ashe	Katarina	Warwick	Sona	Yasuo	Caitlyn
90	1469	0	Volibear	Twisted Fate	Blitzcrank	Morgana	Jinx	Vayne	Leona	Sejuani	Vladimir	Gragas
91	1543	1	Ezreal	Karma	Katarina	Yasuo	Udyr	Heimerdinger	Renekton	Blitzcrank	Xin Zhao	Tristana
92	2149	0	Kayn	Vi	Ziggs	Zyra	Ashe	Ahri	Xayah	Maokai	Kayle	
93	2050	0	Vi	Kalista	Thresh	Kled	Kassadin	Xerath	Talon	Blitzcrank	Udyr	Jinx
94	2131	1	Diana	Braum	Gnar	Kassadin	Jinx	Draven	Gragas	Renekton	Katarina	Thresh
95	2039	1	Lucian	Jarvan IV	Mordekaiser	Xerath	Ahri	Kayn	Vayne	Rakan	Cassiopeia	Camille
96	2600	0	Braum	Rumble	Kayn	Varus	Morgana	Lee Sin	Yasuo	Tristana	Nasus	Lulu
97	1761	0	Trundle	Varus	Lulu	Yasuo	Kled	Akali	Jinx	Leona	Nasus	Vi
98	1534	0	Fiddlesticks	Kled	Ahri	Udyr	Varus	Orianna	Lucian	Jarvan IV	Leona	Jax
99	1226	1	Morgana	Zed	Kennen	Vi	Twitch	Rengar	Orianna	Darius	Leona	Draven

AVL Tree 에 있는 모든 GameData 를 ID 의 오름차순으로 출력을 한다. 만약 AVL Tree 가 구성이 되어있지 않은 경우 ERROR 를 출력한다.

* PRINT_CHARACTER

PRINT CHARACTER			
Name	Pick	Win	WinRatio
Ahri	12	6	50
Akali	5	3	60
Alistar	10	4	40
Amumu	4	1	25
Anivia	2	1	50
Annie	4	2	50
Ashe	15	5	33.3333
Aurelion Sol		1	100
Azir	1	0	0
Bard	4	2	50
Blitzcrank		8	75
Brand	9	6	66.6667
Braum	8	4	50
Caitlyn	13	6	46.1538
Camille	4	3	75
Cassiopeia		10	40
Cho'Gath		7	28.5714
Corki	1	0	0
Darius	8	6	75
Diana	5	1	20
Dr. Mundo		6	50
Draven	10	8	80
Ekko	13	8	61.5385
Elise	5	2	40
Evelynn	1	0	0
Ezreal	11	6	54.5455
Fiddlesticks		5	60
Fiora	7	3	42.8571
Fizz	9	6	66.6667
Galio	2	1	50
Gangplank		6	33.3333
Garen	7	3	42.8571
Gnar	15	11	73.3333
Gragas	15	10	66.6667
Graves	4	1	25
Hecarim	3	1	33.3333
Heimerdinger		8	62.5
Illaoi	1	1	100
Irelia	3	2	66.6667
Ivern	4	0	0
Janna	14	5	35.7143
Jarvan IV		17	47.0588
Jax	19	11	57.8947
Jayce	2	0	0
Jhin	14	6	42.8571
Jinx	17	7	41.1765
Kalista	8	2	25
Karma	3	0	0
Kassadin		6	66.6667
Katarina		7	71.4286
Kayle	7	4	57.1429

B+ Tree 에 있는 모든 CharacterData 를 캐릭터 이름의 오름차순으로 출력한다. 만약 B+ Tree 가 구성되어 있지 않은 경우 ERROR 를 출력한다.

Kayn	17	10	58.8235	
Kennen	3	2	66.6667	
Kha'Zix	6	4	66.6667	
Kled	7	7	100	
Kog'Maw	7	3	42.8571	
LeBlanc	7	4	57.1429	
Lee Sin	13	3	23.0769	
Leona	15	5	33.3333	
Lissandra		2	0	0
Lucian	13	2	15.3846	
Lulu	16	9	56.25	
Lux	14	5	35.7143	
Malphite		4	1	25
Malzahar		6	3	50
Maokai	18	9	50	
Master Yi		8	5	62.5
Miss Fortune		10	5	50
Mordekaiser		3	1	33.3333
Morgana	11	6	54.5455	
Nami	7	4	57.1429	
Nasus	11	6	54.5455	
Nautilus		6	2	33.3333
Nidalee	5	1	20	
Nocturne		2	1	50
Nunu	2	2	100	
Olaf	2	0	0	
Orianna	13	7	53.8462	
Ornn	1	0	0	
Pantheon		2	0	0
Poppy	1	0	0	
Quinn	1	0	0	
Rakan	13	7	53.8462	
Rammus	6	4	66.6667	
Rek'Sai	3	1	33.3333	
Renekton		4	3	75
Rengar	2	1	50	
Riven	12	2	16.6667	
Rumble	3	2	66.6667	
Ryze	3	0	0	
Sejuani	5	3	60	
Shaco	6	4	66.6667	
Shen	5	1	20	
Singed	0	0	0	
Sion	3	2	66.6667	
Sivir	6	2	33.3333	
Sona	12	4	33.3333	
Soraka	7	6	85.7143	
Swain	4	3	75	
Syndra	11	6	54.5455	
Tahm Kench		1	0	0
Taliyah	6	4	66.6667	
Talon	5	2	40	
Taric	7	4	57.1429	
Teemo	5	4	80	
Thresh	25	13	52	
Tristana		25	18	72
Trundle	4	1	25	
Tryndamere		9	6	66.6667
Twisted Fate		6	6	100
Twitch	22	14	63.6364	
Udyr	5	2	40	
Urgot	5	1	20	
Varus	7	6	85.7143	
Vayne	16	11	68.75	
Veigar	10	6	60	
Vel'Koz	5	1	20	
Vi	11	6	54.5455	
Viktor	5	1	20	
Vladimir		5	2	40
Volibear		3	1	33.3333
Warwick	12	7	58.3333	
Wukong	2	0	0	
Xayah	11	5	45.4545	
Xerath	5	1	20	
Xin Zhao		10	4	40
Yasuo	14	4	28.5714	
Yorick	2	1	50	
Zac	4	1	25	
Zed	13	7	53.8462	
Ziggs	1	1	100	
Zilean	2	2	100	
Zyra	9	4	44.4444	

* PRINT_PICK

PRINT PICK			
Name	Pick	WinRatio	
Tristana	25	52	72
Thresh	22	63.6364	
Twitch	19	57.8947	
Jax	18	50	
Maokai	17	58.8235	
Kayn	17	41.1765	
Jinx	17		47.0588
Jarvan IV	16	68.75	
Vayne	16	56.25	
Lulu	15	33.3333	
Leona	15	66.6667	
Gragas	15	73.3333	
Gnar	15	33.3333	
Ashe	14	28.5714	
Yasuo	14	35.7143	
Lux	14	42.8571	
Jhin	14	35.7143	
Janna	13	53.8462	
Zed	13	53.8462	
Rakan	13	53.8462	
Orianna	13	15.3846	
Lucian	13	23.0769	
Lee Sin	13	61.5385	
Ekko	13	46.1538	
Caitlyn	12	58.3333	
Warwick	12	33.3333	
Sona	12	16.6667	
Riven	12	50	
Ahri	11	45.4545	
Xayah	11	54.5455	
Vi	11	54.5455	
Syndra	11	54.5455	
Nasus	11	54.5455	
Morgana	11	54.5455	
Ezreal	10	60	40
Xin Zhao	10		50
Veigar	10	80	
Miss Fortune	10		40
Draven	10	40	
Cassiopeia	9	44.4444	
Alistar	9	66.6667	
Zyra	9	66.6667	
Tryndamere	9	66.6667	
Fizz	8	25	62.5
Brand	8		62.5
Master Yi	8	75	
Kalista	8	50	
Heimerdinger	8		
Darius	8		
Braum	8		

B+ Tree 에 있는 모든 CharacterData 를 Pick 의 데이터로 max-heap 에 push 하여 가장 높은 pick 의 data 부터 pop 을 하여 출력한다. 만약 heap 이 구성되어 있지 않은 경우 ERROR 를 출력한다.

Blitzcrank		8	75	
Varus	7		85.7143	
Taric	7		57.1429	
Soraka	7		85.7143	
Nami	7		57.1429	
LeBlanc	7		57.1429	
Kog'Maw	7		42.8571	
Kled	7		100	
Kayle	7		57.1429	
Katarina		7		71.4286
Garen	7		42.8571	
Fiora	7		42.8571	
Cho'Gath		7		28.5714
Twisted Fate		6		100
Taliyah	6		66.6667	
Sivir	6		33.3333	
Shaco	6		66.6667	
Rammus	6		66.6667	
Nautilus		6		33.3333
Malzahar		6		50
Kha'Zix	6		66.6667	
Kassadin		6		66.6667
Gangplank		6		33.3333
Dr. Mundo		6		50
Xerath	5		20	
Vladimir		5		40
Viktor	5		20	
Vel'Koz	5		20	
Urgot	5		20	
Udyr	5		40	
Teemo	5		80	
Talon	5		40	
Shen	5		20	
Sejuani	5		60	
Nidalee	5		20	
Fiddlesticks		5		60
Elise	5		40	
Diana	5		20	
Akali	5		60	
Zac	4		25	
Trundle	4		25	
Swain	4		75	
Renekton		4		75
Malphite		4		25
Ivern	4		0	
Graves	4		25	
Camille	4		75	
Bard	4		50	
Annie	4		50	
Amumu	4		25	
Volibear		3		33.3333
Sion	3		66.6667	
Ryze	3		0	
Rumble	3		66.6667	
Rek'Sai	3		33.3333	
Mordekaiser		3		33.3333
Kennen	3		66.6667	
Karma	3		0	
Irelia	3		66.6667	
Hecarim	3		33.3333	
Zilean	2		100	
Yorick	2		50	
Wukong	2		0	
Rengar	2		50	
Pantheon		2		0
Olaf	2		0	
Nunu	2		100	
Nocturne		2		50
Lissandra		2		0
Jayce	2		0	
Galio	2		50	
Anivia	2		50	
Ziggs	1		100	
Tahm Kench		1		0
Quinn	1		0	
Poppy	1		0	
Ornn	1		0	
Illaoi	1		100	
Evelynn	1		0	
Corki	1		0	
Azir	1		0	
Aurelion Sol		1		100
Singed	0		0	

* PRINT_WINRATIO

PRINT WINRATIO				
Name	Pick	WinRatio		
Azir	1	0		
Corki	1	0		
Evelynn	1	0		
Ivern	4	0		
Jayce	2	0		
Karma	3	0		
Lissandra		2	0	
Olaf	2	0		
Ornn	1	0		
Pantheon		2	0	
Poppy	1	0		
Quinn	1	0		
Ryze	3	0		
Singed	0	0		
Tahm Kench		1	0	
Wukong	2	0		
Lucian	13	15.3846		
Riven	12	16.6667		
Diana	5	20		
Nidalee	5	20		
Shen	5	20		
Urgot	5	20		
Vel'Koz	5	20		
Viktor	5	20		
Xerath	5	20		
Lee Sin	13	23.0769		
Amumu	4	25		
Graves	4	25		
Kalista	8	25		
Malphite		4	25	
Trundle	4	25		
Zac	4	25		
Cho'Gath		7	28.5714	
Yasuo	14	28.5714		
Ashe	15	33.3333		
Gangplank		6	33.3333	
Hecarim	3	33.3333		
Leona	15	33.3333		
Mordekaiser		3	33.3333	
Nautilus		6	33.3333	
Rek'Sai	3	33.3333		
Sivir	6	33.3333		
Sona	12	33.3333		
Volibear		3	33.3333	
Janna	14	35.7143		
Lux	14	35.7143		
Alistar	10	40		
Cassiopeia		10	40	
Elise	5	40		
Talon	5	40		
Udyr	5	40		

B+ Tree 에 있는 모든 CharacterData 를 winratio 의 데이터로 min-heap 에 push 하여 가장 낮은 승률의 data 부터 pop 을 하여 출력한다. 만약 heap 이 구성되어 있지 않은 경우 ERROR 를 출력한다.

Vladimir		5		40
Xin Zhao		10		40
Jinx	17		41.1765	
Fiora	7		42.8571	
Garen	7		42.8571	
Jhin	14		42.8571	
Kog'Maw	7		42.8571	
Zyra	9		44.4444	
Xayah	11		45.4545	
Caitlyn	13		46.1538	
Jarvan IV		17		47.0588
Ahri	12		50	
Anivia	2		50	
Annie	4		50	
Bard	4		50	
Braum	8		50	
Dr. Mundo		6		50
Galio	2		50	
Malzahar		6		50
Maokai	18		50	
Miss Fortune		10		50
Nocturne		2		50
Rengar	2		50	
Yorick	2		50	
Thresh	25		52	
Orianna	13		53.8462	
Rakan	13		53.8462	
Zed	13		53.8462	
Ezreal	11		54.5455	
Morgana	11		54.5455	
Nasus	11		54.5455	
Syndra	11		54.5455	
Vi	11		54.5455	
Lulu	16		56.25	
Kayle	7		57.1429	
LeBlanc	7		57.1429	
Nami	7		57.1429	
Taric	7		57.1429	
Jax	19		57.8947	
Warwick	12		58.3333	
Kayn	17		58.8235	
Akali	5		60	
Fiddlesticks		5		60
Sejuani	5		60	
Veigar	10		60	
Ekko	13		61.5385	
Heimerdinger		8		62.5
Master Yi		8		62.5
Twitch	22		63.6364	
Brand	9		66.6667	
Fizz	9		66.6667	
Gragas	15		66.6667	
Irelia	3		66.6667	
Kassadin		6		66.6667
Kennen	3		66.6667	
Kha'Zix	6		66.6667	
Rammus	6		66.6667	
Rumble	3		66.6667	
Shaco	6		66.6667	
Sion	3		66.6667	
Taliyah	6		66.6667	
Tryndamere		9		66.6667
Vayne	16		68.75	
Katarina		7		71.4286
Tristana		25		72
Gnar	15		73.3333	
Blitzcrank		8		75
Camille	4		75	
Darius	8		75	
Renekton		4		75
Swain	4		75	
Draven	10		80	
Teemo	5		80	
Soraka	7		85.7143	
Varus	7		85.7143	
Aurelion Sol		1		100
Illaoi	1		100	
Kled	7		100	
Nunu	2		100	
Twisted Fate		6		100
Ziggs	1		100	
Zilean	2		100	

O Consideration

이번 프로젝트에서는 새로운 Tree 를 이용하여 설계를 하였다. AVL Tree 와 B+ Tree 를 이용하였는데, 처음에는 AVL Tree 의 rotation 방법과 어떤 경우에 어떤 rotation 을 해야 하는지 이해가 가지 않았다. 마찬가지로 B+ Tree 에서 data split 과 index split 에 대한 이해도도 떨어졌다. 프로젝트를 준비하면서 공책에 rotation 과 split 을 직접 그리면서 다양한 케이스를 따져보니 명확한 해법을 얻어냈고, 그 결과 프로젝트를 완성할 수 있었다. B+ Tree 에서 map 이라는 새로운 개념을 이용하였는데 처음에는 왜 굳이 이런 걸 쓰나 싶었지만, 설계를 하면 할수록 효율성이 뛰어나다는 것을 알게 되었다. 이번 프로젝트를 진행 하면서 상속에 대해서도 알게 되었다.

다양한 시도 끝에 프로젝트를 완성할 수 있어서 뿌듯하고 3 차 프로젝트로 성공적으로 완성 해야 겠다.