BLG336E
Homework-1

24.03.2020

Res. Asst. Yusuf Huseyin Sahin
sahinyu@itu.edu.tr

Prepare for trouble, and make it double
To protect the world from devastation,
To unite all people within our nation,
To denounce the evil of truth and love,
To extend our reach to the stars above
Jesse! James!
Team Rocket blasts off at the speed of light,
Surrender now, or prepare to fight

Team Rocket

- You should write all your code in C++ language.
- Your code should be able to be compiled with default g++ compiler and run under Ubuntu OS.
- Disciplinary actions of the faculty will be applied for plagiarism.
- For every part of the homework, same program should be run with different command line arguments. The codes not using these arguments will not be graded.

1 - Overview

In this homework you will simulate a Pokémon battle where a Pikachu is fighting against a Blastoise. For that problem I simplified the Pokémon fighting procedures given in [1]. In the simplified procedure, there are two attributes of each Pokémon: HP and PP.

- **HP (Health Points)**: Total health of a Pokémon. May be decreased after attack of the opponent. No defense mechanism is included.
- **PP (Power Points)**: Total ability to do an attack. Each Pokémon starts with a total of 100 PP and it decreases according to each attack's PP values. If Skip attack is used, +100 PP is obtained.

[1] https://www.math.miami.edu/~jam/azure/pokedex

Also for an attack there are four different attributes:

- **PP**: To do an attack, PP value of the selected attack should be decreased from Pokémon's own PP. If this value is greater than the Pokémon's PP, attack cannot be used.
- **Accuracy**: Accuracy of an attack. If it is not an attack with 100% accuracy, multiple nodes should be created in the graph.
- **Damage**: Damage of the attack.
- **FirstUsage**: The first level of the graph where the attack may be used.

To simulate the graph you should create a graph. Some important rules for creating the graph are given below.

- The HPs of Pikachu and Blastoise are 273 and 361 respectively. Attack properties of them are given in text files.
- A node is a leaf node, if one of the competitors are knocked-out or the level limit of the tree is reached. No children of a leaf node is allowed.
- In the simulation, it is taught that Pokémons are not using a decision mechanism to select an attack. Thus, for a list of attacks, it is equally likely to select one of them.

The graph's first three levels are given in the figure below.

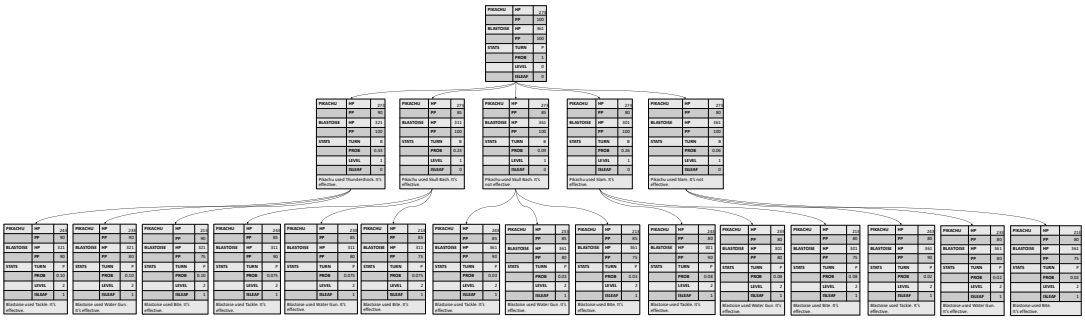

Figure 1: First three layers of the match

2 Graph Implementation (40 pts.)

Implement the code which creates a graph according to the rules given in the Overview section. You should create the graph according to the max-level value given in the graph. Your code here should output the last layer's node information. An example run is given below:

```
g++ main.cpp -o project1
./project1 part1 2

P_HP:243 P_PP:90 B_HP:321 B_PP:90 PROB:0.1111
P_HP:233 P_PP:90 B_HP:321 B_PP:80 PROB:0.1111
P_HP:213 P_PP:90 B_HP:321 B_PP:75 PROB:0.1111
P_HP:243 P_PP:85 B_HP:311 B_PP:90 PROB:0.0762
P_HP:233 P_PP:85 B_HP:311 B_PP:80 PROB:0.0762
P_HP:213 P_PP:85 B_HP:311 B_PP:75 PROB:0.0762
P_HP:243 P_PP:85 B_HP:361 B_PP:90 PROB:0.0326
P_HP:233 P_PP:85 B_HP:361 B_PP:80 PROB:0.0326
P_HP:213 P_PP:85 B_HP:361 B_PP:75 PROB:0.0326
P_HP:243 P_PP:80 B_HP:301 B_PP:90 PROB:0.0792
P_HP:233 P_PP:80 B_HP:301 B_PP:80 PROB:0.0792
P_HP:213 P_PP:80 B_HP:301 B_PP:75 PROB:0.0792
P_HP:243 P_PP:80 B_HP:361 B_PP:90 PROB:0.0198
P_HP:233 P_PP:80 B_HP:361 B_PP:80 PROB:0.0198
P_HP:213 P_PP:80 B_HP:361 B_PP:75 PROB:0.0198
```

3 BFS-DFS Implementation (40 pts.)

Using the graph generated by the functions in the previous part, implement BFS and DFS algorithms to traverse the graph. Run both BFS and DFS algorithms and print node count and running time. Analyze the results. Your code should be run by **"./project1 part2 <max-level> dfs"** or **"./project1 part2 <max-level> bfs"** commands.

4 Probability of the Easiest Path (20 pts.)

For both Pikachu and Blastoise, find out the probability of the easiest action sequence (containing minimum number of levels) to win the battle. Select one of the algorithms in the previous part to solve the problem. An example output is given below.

```
./project1 part3 pikachu

Pikachu used x. It's <effective/noneffective>.
Blastoise used y. It's <effective/noneffective>.
...

Level count : -
Probability : -
```