Algorithms Programming Project

1 Problem Definition

You are given n bags filled with devices. For each bag, you are also given $numWorking_i$ and $total_i$ that represent the number of working devices and the total number of devices for bag i, $0 \leq i < n$. You are also given k extra working devices. Place these k extra working devices in the bags in such a way that the average percentage of working devices in for all bags is maximized. You may place multiple devices in each bag.

2 Greedy Strategies

For each of the following greedy strategies, you must i) design a $\theta(n + k\log(n))$ algorithm for each strategy; ii) provide an example for the strategy that yields the optimal solution; iii) provide an example for each strategy that does not yield the optimal solution or prove that it is an optimal strategy.

STRAT1 For each extra device, place it in the bag, that currently has the minimum number of working devices. Pick the bag i with minimum $numWorking_i$.

STRAT2 For each extra device, place it in the bag, that currently has the minimum ratio number of working devices and total devices. Pick the bag i with minimum $\frac{numWorking_i}{total_i}$

STRAT3 For each extra device, place it in the bag, that currently has the minimum number of total devices. Pick the bag i with minimum $total_i$

STRAT4 For each extra device, place it in the bag, whose percentage of number of working devices will increase the most. Pick the bag i with maximum $\frac{numWorking_i+1}{total_i+1} - \frac{numWorking_i}{total_i}$

If there are multiple bags that can be picked using a particular strategy, pick the bag with a lesser index.

3 Programming Tasks

Once you complete the algorithm design tasks, you should have an implementation for each of the following programming procedures:

TASK1 Give an implementation of STRAT1.

TASK2 Give an implementation of STRAT2.

TASK3 Give an implementation of STRAT3.

TASK4 Give an implementation of STRAT4.

4 Language/Input/Output Specifications

You may use Java, Python, or C++. Your program must compile/run on the Thunder CISE server using gcc/g++, Python-3 interpreter, or standard JDK. You may access the server using SSH client on thunder.cise.ufl.edu. You must write a makefile document that executes first any

compilation tasks then will run each task. First the command **make** will be run. Then each task should have its own command in the format **make run#** where the number corresponds to each task e.g. when **make run3** is called from the terminal, your program needs to execute the implementation of Task3.

Input. Your program will read input from standard input (stdin) in the following order:

- Line 1 consists of two integers n and k separated by a single space.
- next n lines consists of 2 integers with each line being the number of working devices and the total number of devices for the bag separated by a single space.

Output. print a set of indices that correspond to the bag that you place a device in to achieve the maximum average percentage of working devices in the order you place them.

- Each bag index should be outputted on a single line separated by a single space.

For convenience assume that $1 \leq n, k < 10^5$, and $\forall i \;\; 0 \leq numWorking_i, total_i < 10^5$.

5 Experiments and Report

You should conduct an experimental study to test the optimality of all the strategies. Your report should include at least the following components: i) Team members; ii) Greedy Strategies; iii) Experimental Comparative Study; iv) Conclusion.

5.1 Team Members

You are allowed to work as teams of (at most) three students on this programming assignment. If you decide to work as a team, clearly state the names of team members and describe the main contributions of each member.

5.2 Greedy Strategies

For each of the four greedy strategies, you should clearly describe the algorithm and analysis. You must also provide examples that supports and contradicts each strategy. When a strategy is optimal, you must argue its correctness.

5.3 Experimental Comparative Study

You are expected to test your implementations for their optimality. For this purpose, you should create randomly generated input files of various sizes. The exact size of the experimental data sets that your program can handle depends on the quality of your implementation. For instance, you might want to choose $n = 1000, 2000, 3000, 4000, 5000$ with a fixed value for k. Then, you compare the average percentage for each task for each input size. You should plot a grouped bar graph with average percentage (y-axis) against input size (x-axis). You should group together all tasks for a given input size.

5.4 Conclusion

Summarize your learning experience on this project assignment. For each programming task, comment on ease of implementation and other potential technical challenges.

6 Submission

The following contents are required for submission:

1. **Makefile**: Your makefile must be directly under the zip folder. No nested directories. Do not locate the executable file in any directory either.
2. **Source code**: should include detailed comments next to each non-trivial block of code.
3. **Report**: The report must be in PDF format.
4. **Bundle**: Compress all your files together using a zip utility and submit through the Canvas system. Your submission should be identified with your group number, i.e., **Group##.zip**.

7 Grading Policy

Grades will be based on the correctness & efficiency of algorithms and the quality of the report:

- **Program 50%.** Correct/efficient design and implementation/execution. Also make sure to include comments with your code for clarity.
- **Report 50%.** Quality (clarity, details) of the write up on your design, analysis, programming experience, and experimental study.

8 Honesty Policy

All students admitted to the University of Florida have signed a statement of academic honesty committing themselves to be honest in all academic work and understanding that failure to comply with this commitment will result in disciplinary action. A student is responsible for all parts of their submission. It is expected that all submitted code will be written by you. Any external sources that are referred must be properly cited. Refer to https://policy.ufl.edu/wp-content/uploads/2020/12/UF-Regulation-4.040.pdf for more information.