

Final Assessment - Final Exam Period

COMP-251, Fall 2020

Due: December **22th**, 2020 (23:59 EST)

Please read the entire PDF before starting. You must do this longer form assessment individually.

It is very important that you follow the directions as closely as possible. They are designed to help you construct your solution optimally, which will help maximize your grade when your code is evaluated through automated tests.

To be eligible for full marks, you must:

- Follow all the directions below.
- Code your solution using the template provided with this handout. **Do not modify the template outside of the regions you are asked to modify..**
- Make sure that your code compiles and is correct.
 - Non-compiling code (code with syntax error) will receive a very low grade.
- Submit your code on codePost
 - Make sure that your code passes ALL the given open test cases.
 - Make sure your code is correct and efficient to pass ALL the blind test cases.
 - We cannot make any exceptions for submissions because the final deadline is established by McGill. Make sure you have a working codePost account and a first version submitted at least 24 hours before the deadline. If you try to upload last minute for the first time and it does not work, you will receive a grade of 0 for the project.
- Make sure you adhere to the code of conduct and integrity, and do not engage in any act of academic dishonesty, including but not limited to, using unauthorized aid and assistance, and committing plagiarism. In submitting this exam, you confirm that your conduct during this final assessment adheres to the Code of conduct and academic integrity (<https://www.mcgill.ca/students/srr/academicrights>)
 - This assessment is individual. You cannot share any code, test cases, strategies or other information about the assessment with other students or on the internet.
 - You may use all resources shared in the context of the course (textbooks, slides, recordings, old Piazza posts, etc), as well as Java Documentation, but nothing else.
 - You will not share or disseminate this final assessment on any platform or through personal communication.

The problem

It was November 2nd, 2020, and I was sitting in front of my computer thinking of an exercise to give to COMP251 students as a long term assessment. I had been procrastinating for almost 3 hours and I wasn't even close to having a decent idea for an exercise. While surfing the internet every page was talking about the United States Presidential elections. This year they were scheduled for Tuesday, November 3, 2020 (i.e., the day after writing this assessment outline). I did not know that US elections usually take place on the first Tuesday in November. This information is not useful at all for COMP251, but I thought it would be cool to share anyway. Thanks to the internet, I also discovered that the US electoral system is quite complicated! Particularly, I found the following interesting information:

1. The American electorate were to decide between incumbent President Donald Trump and his Democratic challenger, former Vice President Joe Biden.
2. The American people indirectly elect the president because even if each voter selects between Mr. Trump and Mr. Biden, they are actually voting for a representative of that candidate's party known as an *elector*. The totals are not calculated nationwide, but the president is elected by an Electoral College selected state-by-state. The electoral college meets a few weeks after election day, to carry out the task of choosing the president (to vote for the candidate who got the majority of the votes in their own state). This system is usually known as a 'winner-takes-all' because the candidate with the highest number of votes in a state claims all the electoral votes of that state.
3. For the 2020 elections, it was guaranteed that every state had at least one delegate (the number of electoral votes of a state) and at least one registered voter.
4. For the 2020 elections, there are 538 delegates, but **for our test cases we can have at most 2016 delegates.**
5. In 2016, Mr. Trump beat Mrs. Clinton in Florida by a margin of just 2.2%, but that meant he claimed all 29 of Florida's crucial electoral votes.
6. National polls were giving Mr. Biden an average lead of over 10%. However, is that enough, given the indirect way the United States elects its president?
7. There were over 100 million cast ballots in historic early voting, with millions more heading to the polls on Tuesday. Then, the electoral threshold (maximum number of allowed voters) was this year set to be 1 billion (10^9) cast ballots.
8. If there were a tie (either at the State or National level) between Mr. Trump and Mr. Biden, the US House of Representatives determines the president. As of November 2nd, 2020, Republicans were in the majority, with control of 26 state delegations, and as a consequence, Mr. Trump would be elected as the new president.
9. Based on the State Polls, Joe Biden was leading key states (the so-called battleground states including Michigan, Wisconsin, and Pennsylvania) that gave Mr. Trump his victory back in 2016.
10. Most State Polls were wrong about the 2016 presidential election. For example, the CBC's Presidential Poll Tracker gave Mrs. Clinton a 3.4-point lead in the national polls over Mr. Trump on the election day. She was projected to win North Carolina, Florida, Michigan, Pennsylvania, and Wisconsin; however, Mr. Trump won all of these states and secured more than the 270 electoral college votes needed to win the White House.

At this point in my procrastination, I got interested in the polls that compiled information about the voting preferences per state. Particularly, for each State, I compiled the information regarding *i*) The expected number of people who are projected to vote for Mr. Biden, *ii*) The expected number of people who are projected to vote for Mr. Trump and *iii*) The number of people who have not made a voting decision yet. The technical description of the polls makes it clear that the numbers of the first two categories are not likely to change because they are votes from people with long-time roots in a particular political party. On the other hand, people in the third category are the ones who expressed no preference and did not lean towards either of the major parties. At this point (also of my procrastination), I realized that it would be great to

have a very efficient algorithm to know the minimum number of people that Mr. Biden needs to convince in order to secure the presidency of the United States of America. I was about to start coding my solution when I realized that this could be a nice and fun exercise for COMP251 students. The idea is that I would provide you the files (some of them will be open cases and others will be blind cases) containing the following information:

- The first line of my file contains a single integer (i.e., *num_states*) that represents the number of states taken into account by a poll.
- Following *num_states* lines each contain four integers (separated by spaces) with the following information.
 1. The number of delegates for a specific state.
 2. The number of people who will vote for Mr. Biden in that state.
 3. The number of people who will vote for Mr. Trump in that state.
 4. The number of people who have not made a voting decision in that state yet.

For each provided file you must calculate the minimum number of people that Mr. Biden would have to convince to earn the US presidency. If it is not possible for Mr. Biden to win the election, you must output the integer -1 .

Let see some examples of my files with the expected answers:

Sample Input 1:

```
3
7 2401 3299 0
6 2401 2399 0
2 750 750 99
```

Sample Output 1: 50

Sample Input 2:

```
3
7 100 200 200
8 100 300 200
9 100 400 200
```

Sample Output 2: -1

Sample Input 3:

```
3
32 0 0 20
32 0 0 20
64 0 0 41
```

Sample Output 3: 32

Given the different electoral fraud claims and legal challenges launched after election day and the long process to give a verdict, it would be fundamental that your algorithm would be **correct** and **efficient**. In other words, your algorithm must take all possible instances of the described problem to the right result (you do not want your algorithm to be used as a proof of electoral fraud and get a very low mark) and your algorithm must do that in a reasonable amount of time as a function of the size of the input (you do not want your algorithm to take days to produce results and get a very low mark). In order to guarantee that, we recommend you to: i) review the algorithms covered in class looking for the optimal technique to be applied to the problem and, ii) generate a good number of quality test cases (or develop a proof of correctness) to be sure that your algorithm is correct.

Your task: completing the solution method

For this part of the assessment, you will create a function called `solution` which gets the following parameters:

- An `int` variable called `num_states` that represents the number of states considered by the Poll.
- An `int[]` array of integers called `delegates` with the number of delegates for the `num_states` states.
- An `int[]` array of integers called `votes_Biden` with the number of votes for Mr Biden for the `num_states` states.
- An `int[]` array of integers called `votes_Trump` with the number of votes for Mr Trump for the `num_states` states.
- An `int[]` array of integers called `votes_Undecided` with the number of Undecided votes for the `num_states` states.

The function `solution` must return an `int` representing the minimum number of people that Mr Biden needs to convince in order to secure the presidency of the United States of America, or `-1` if it is impossible and Mr Trump has already secured the re-election.

The signature of the function `solution` in the java file `US_elections` is as follows.

```
public static int solution(int num_states, int[] delegates, int[] votes_Biden, int[]
    votes_Trump, int[] votes_Undecided){

}
```

Please complete the body of the function `solution` and please do not change the methods or constructors that are already given to you, do not import extra code and do not touch the method headers.

Note: main function

We have already implemented a `main` function to read the data from the files, to create the variables and arrays that are passed as arguments to the function `solution` and finally to call the function `solution`. Please note that this function will not be graded, and it is there only to make sure that all of the Comp251 students understand the input of the function `solution` and to test your own code.

The `main` function is defined in the java file `US_elections`. Please do not change or alter the function `main` in any way.

What To Submit?

Attached to this assignment is a file called `US_elections.java`. PLEASE note that you only need to complete the body of the function `solution`. Then, please do not change or alter the name of the file (`US_elections`) you must submit, or the method headers in this file. Files with the wrong name will not be

graded. Make sure you are not changing file names by duplicating them. You are allowed to create helper functions, but those functions must be located inside the (`US_elections`) file. Do not modify the code in any other way and in particular, do not change the methods or constructors that are already given to you, do not import extra code and do not touch the method headers.

You have to submit only the `US_elections.java`. file. Please DO NOT zip (or rar) your files, and do not submit any other files.

Where To Submit?

You need to submit your assignment on CodePost. Please note that you do not need to submit anything to myCourses.

When To Submit?

Please do not wait until the last minute to submit your final assessment. You never know what could go wrong during the last moment. Please also remember that you are allowed to have multiple submission. So, submit your partial work early and you will be able to upload updated versions later (as long as they are submitted before the deadline).

How will this assessment be graded?

Each student will receive an overall score for this assessment. This score evaluates the correctness and efficiency of your solution.

- **Correctness and time score:** This score is between 0% and 100% depending on the number of test cases your code has passed. We have hundreds of test cases (3 open cases and several blind cases). The open cases will be run within your submissions and you will receive automated test results (i.e., the autograder output) for them. You **MUST** guarantee that your code passes these cases. The blind test cases are inputs that you have not seen and they will test the correctness and speed of your algorithm on those inputs once the deadline of the assignment is over.