

Assignment 1 – Task allocation

About the problem:

The goal of this assignment is to implement a genetic algorithm (GA) for allocating tasks between two processing cores. Each task has a specified execution time, and the objective is to minimize the total execution time while ensuring that neither core exceeds a specified execution time limit.

.

What you are required to do:

Write a canonical genetic algorithm to solve the Task allocation problem.

What the input looks like:

You'll be given an input file with the following format:

- *First line:* Number of test cases (must be at least 1)

For each test case:

- Max time limit
- Number of tasks to be allocated

For each task:

- task execution time

For example:

```
3
100
5
10
20
30
```

40
50
150
4
60
70
80
90
200
3
20
30
40

Important remarks to help you solve the problem:

1. Use a binary, one-dimensional chromosome.
2. Try the following for the population size. You can try different population sizes [50,100, 250] to see how this will affect your results. The maximum number of generations is also up to you.
3. To evaluate the chromosome, assign genes with 1 to core 1 and genes with 0 to core 2 where time taken will be the max between the two.
4. Think about how you will handle **infeasible solutions**. Infeasible solutions are solutions that violate the constraints of the problem; therefore, they are not allowed.
5. Use roulette wheel **selection**
6. Use **one-point crossover**.
7. Use flip bit mutation
8. Elitism replacement

9. The output should consist of the **test case index**, The **best solution** , it **evaluation score**, its **chromosome representation**, the **tasks assigned to each core** and their **total time**.

Assignment submission notes:

- The maximum number of students in a team is 3 and the minimum is 2 **from the same lab**.
- You could use any programming language.
- The deadline will be on **"Saturday 9 Nov 2024 at 11:59 pm"**, and no late submission is allowed.
- Please submit one compressed folder. The folder name should follow this structure: ID1_ID2_ID3_MAJOR_GROUP
- Cheating students will take negative grades and no excuses will be accepted. If you have any problems during the submission, contact your TA but don't, under any circumstances, give your code to or take the code from your friends.

Grading Criteria: (100 marks)

Representation, structure and initialization	15
Fitness function	10
Handling infeasible solutions	5
Selection, crossover, mutation and replacement	60
Output	10

Good luck