Assignment

2021/2022

Luís Paquete
University of Coimbra

Assignment

There are several exams to be scheduled at the DEI and there is a number of free time slots. You have done a program that automatically schedules exams, that is, it assigns an exam to a time slot such that every student does not have to attend more than one exam simultaneously and it uses the least number of time slots available. Assume that the time slots do not overlap in time.

Given the success of your program, you are now planning to sell it to other universities. An important aspect to potential clients is to know how much time they need to wait to obtain a schedule. This should depend on the number of exams, number of time slots and the number of students that have exams. Your goal is to **characterize the performance** of your program for several scenarios.

Assignment

Support material

- Two programs (code1.c and code2.c) available that implement two randomized backtracking algorithms to solve the exam scheduling problem

- **code1.c**

```
Let n be the number of exams
find = True
While find = True:
    If it finds a schedule with n time slots:
      n = n – 1
     Else:
       find = False
Return n + 1
```

Assignment

Support material

- Two programs (code1.c and code2.c) available that implement two randomized backtracking algorithms to solve the exam scheduling problem

- **code2.c**

```
n = 1
find = False
While find = False:
    If it does not find a schedule with n time slots:
      n = n + 1
    Else:
      find = True
Return n
```

Assignment

Support material

- Two programs (code1.c and code2.c) available that implement two randomized backtracking algorithms to solve the exam scheduling problem

- There is a randomly input data generator in python (gen.py). You need to define the number of exams, the probability that each pair of exams will have a student in common, the random seed, and the name of the file that will store the input data.

Note: Test the programs as soon as possible.

Assignment

Example (under linux)

- Compile code1.c using gcc

```
$ gcc code1.c -o code1 -O3
```

- Generate input data with 10 exams, probability 0.2, random seed 3125, and store it in file data.in

```
$ python3 gen.py 10 0.2 3125 data.in
```

- Run the code on file data.in, with random seed 31235, and a cut-off time of 100 secs.

```
$ ./code1 31235 100 data.in
$ 2 0.000218
```

Assignment

Example (under linux)

- Compile code1.c using gcc

```
$ gcc code1.c -o code1 -O3
```

- Generate input data with 10 exams, probability 0.2, random seed 3125, and store it in file data.in

```
$ python3 gen.py 10 0.2 3125 data.in
```

- Run the code on file data.in, with random seed 31235, and a cut-off time of 100 secs.

```
$ ./code1 31235 100 data.in
$ 2 0.000218
```

Output: It found a schedule using 2 time slots in 0.000218 secs.

Assignment

Example (under linux)

- Compile code1.c using gcc

```
$ gcc code1.c -o code1 -O3
```

- Generate input data with 10 exams, probability 0.2, random seed 3125, and store it in file data.in

```
$ python3 gen.py 10 0.2 3125 data.in
```

- Run the code on file data.in, with random seed 31235, and a cut-off time of 100 secs.

```
$ ./code1 31235 100 data.in
$ 2 0.000218
```

An example of file data.in

```
10 7
1 2
2 3
3 5
4 8
5 9
6 10
8 10
```

10 exams and there are 7 pairs of exams with at least one student in common.

Assignment

Example (under linux)

- Compile code1.c using gcc

```
$ gcc code1.c -o code1 -O3
```

- Generate input data with 10 exams, probability 0.2, random seed 3125, and store it in file data.in

```
$ python3 gen.py 10 0.2 3125 data.in
```

- Run the code on file data.in, with random seed 31235, and a cut-off time of 100 secs.

```
$ ./code1 31235 100 data.in
$ 2 0.000218
```

An example of file data.in

```
10 7
1 2
2 3
3 5
4 8
5 9
6 10
8 10
```

Exam 1 and 2 have at least one student in common

Assignment

Assignment

1st Milestone - 9 November (20h, max. 8 pages):

Exploratory Data Analysis: Analyze and investigate the performance of your programs and summarize their main characteristics, using data visualization methods.

Linear Regression: Characterize the performance of your programs with respect to problem size using regression models

- **2nd Milestone** - 23 November (2h, 1 page):

  Pre-registration of hypotheses: Based on the results of the 1st Milestone, identify factors that may affect performance of your programs and formalize hypotheses to test in the 3rd milestone

- **3rd Milestone** - 20 December (20h, max. 8 pages):

  Hypothesis Testing: Test hypotheses under different scenarios

Assignment

Some notes

- In order to obtain results, you need to run many experiments. Take into account that you may need to redo experiments very often.
- Always draw conclusions from the your analysis in the reports for the 1st and the 3rd milestone.
- Submissions of the reports should be uploaded at Inforestudante. Submissions after the deadline (at 23h59) are not possible.

Note: Plagiarism means mandatory fail in the course and internal (UC) disciplinary procedure. Please, refer adequately all text and material you take from the Internet. All parts of the report must be written by the students and not copied & pasted & changed from the Internet.