Project – High-School Schedule/Time Tables

Introduction

Solve the problem in Python by using an **Informed Search Algorithm** (the algorithm implemented needs to be commented in the beginning of the code).

The problem to solve is a problem of optimization, as there is a clear goal.

Problem

. The executable receives as an argument a file with an instance (attached goes examples of instances and there is an example instance, with a solution, in page 2).

. School Schedules considering only classes and teachers (we assume that there are unlimited classrooms of any type).

. Classes are defined by the letter T, Teachers by the letter P and Groups of Lessons by the letter A.
Each lesson is defined by the triple: (Class, Teacher, Number of Lessons).
The number of lessons is the weekly needs/responsibilities of the teacher to the class.

. Weekly schedule: a week has 5 days, 6 periods of lessons per day (we assume a bigger break for lunch).
In each day, one teacher or class can have 4 periods (with or without "holes").
The same pair Class/Teacher can only have one lesson per day.

. "Holes" – Penalty unity.
It is a period without lesson (either for the teacher or the class).
If in one day there is only one lesson, we also consider that one "hole".

. Cost – Sum of all the penalty unities ("Holes").

. **Goal:** Reduce the number of "holes" for the teachers and for the classes.

. **Results:** For each solved instance (stopping criteria 15 seconds of CPU), refer the algorithm/configuration, solution (as in the example in page 2), number of expansions and generations.

Example of an Instance:

he6t6p2a10s.txt

6, 6	1, 1, 2	2, 3, 2	3, 0, 2	4, 2, 2	5, 0, 2
0, 4, 2	1, 0, 2	2, 2, 2	3, 4, 2	4, 1, 2	5, 2, 2
0, 3, 2	1, 2, 2	2, 1, 2	3, 2, 2	4, 3, 2	5, 5, 2
0, 5, 2	1, 4, 2	2, 4, 2	3, 5, 2	4, 5, 2	5, 3, 2
0, 2, 2	1, 5, 2	2, 5, 2	3, 1, 2	4, 0, 2	5, 1, 2
0, 1, 2	1, 3, 2	2, 0, 2	3, 3, 2	4, 4, 2	5, 4, 2
0, 0, 2					

. In the first line, the instance has: Number of Classes, Number of Teachers.

. In the following lines it has: Number of the Class, Number of the Teacher, Number of weekly Lessons (This number can vary, but in mostly all the instances this value is always the same. To test the implementation with different weekly lessons, run the file *he9t6p2a10s_diffwlessons.txt* – it does not have, necessarily, a solution).

(*Note*: The instance is in a table with 5 columns in order to save space. If it can induce to error, take in consideration the instance in the file attached with the name of the instance.)

Example of a solution for the instance above:

```
Classes:
TA | FAEC.. | AFBE.. | DBC... | D..... | ......
TB | CBDE.. | CEDA.. | A.FB.. | F..... | ......
TC | DCFA.. | DBFC.. | BAE... | E..... | ......
TD | EFAB.. | ED.B.. | FCAD.. | C..... | ......
TE | ADC... | FCAD.. | EFB... | BE.... | ......
TF | BE.F.. | BACF.. | CD.E.. | AD.... | ......
Teachers:
PA | EADC.. | AFEB.. | BCD... | F..... | ......
PB | FB.D.. | FCAD.. | CAEB.. | E..... | ......
PC | BCEA.. | BEFC.. | FDA... | D..... | ......
PD | CEB... | CDBE.. | AF.D.. | AF.... | ......
PE | DFAB.. | DB.A.. | E.CF.. | CE.... | ......
PF | ADCF.. | EACF.. | DEB... | B..... | ......
Cost: 16
```

To note:

. The class A (TA) has the first period with the teacher F (PF), and vice versa, that's why the line of the TA starts with F, and the line of the PF starts with A.

. This solution has a cost of 16, because the class A has an isolated lesson (in the 4th day of the week) and so on (the cost is the sum of the "holes")
This is a valid solution as it satisfied the constraints. Ideally the solution should have the least cost as possible.
(*Note*: In the solution, each column represents a day of the week.)