

TIMETABLE GENERATOR USING HEURISTIC APPROACH

Anwesh Marwade 111303005
Pranay Deshmukh 111303010
Nakul Wenger 111303032

>>HOW TO RUN:

Compile: cc *.c -o run

Run: ./run

>>INPUT/ OUTPUT:

*/*Structure of input files f1.txt and f2.txt*/*

Teacher file format(f1.txt):

<No. of teachers n>

<teacher1 id>, <teacher1 name>

<teacher2 id>, <teacher2 name>

.

.

.

.

<teachern id>, <teachern name>

e.g.

6

1, Ghotkar

2, Jignesh M

3, A Sawant

4, Y Haribhakt

5, Jibi Abraham

6, M Khandekar

Class-Course file format(f2.txt):

<No. of calsses n>

<class1 name>, <class1 no. of courses m>

<class1_course1 id>, <course1 name>, <course1 credits>, <course1 type>, <course1 teacher id>

.

.

.

<class1_coursem id>, <coursem name>, <coursem credits>, <coursem type>, <coursem teacher id>

.
 .
 .
 <classn name>, <classn no. of courses m>
 <classn_course1 id>, <course1 name>, <course1 credits>, <course1 type>, <course1 teacher id>
 .
 .
 .
 <classn_coursem id>, <coursem name>, <coursem credits>, <coursem type>, <coursem teacher id>

e.g.

2

fy, 3

1, daa, 3, T, 1

3, jokes, 3, T, 6

2, sp, 3, L, 5

sy, 6

10, daa, 3, T, 2

20, cn, 3, T, 3

30, co, 3, T, 3

40, dbms, 3, T, 1

50, sp, 3, L, 3

60, sp, 3, T, 3

NOTE: Take care while giving proper no. of records in file input. Check on file if it is not provided e.g., if <no of teachers> = 5 and following teacher records are 6, no check is kept

//input

******* TIMETABLE GENERATOR *******

Enter the Dept name: Computer //enter dept name whose timetable to be generated

Enter no. of slots in a day(max 12): 8 //enter no. of slots as per requirement

Enter the teacher file: f1.txt //enter filename that contains teacher data

Enter the class and course file: f2.txt //enter filename that contains class-course data

//output

//teacher details will be displayed

//class-course data will be displayed

***** TIMETABLE GENERATOR *****

Enter the Dept name: Computer

Enter no. of slots in a day(max 12): 8

Enter the teacher file: f1.txt

Teacher details:

Id	Name
1	Ghotkar
2	Jignesh M
3	A Sawant
4	Y Haribhakt
5	Jibi Abraham
6	M Khandekar

Enter the class and course file: f2.txt

Class fy(1):

Course Id	Name	Credits	Teacher Id
1	daa	3	1
3	jokes	3	6
2	sp	3	5

Class sy(2):

Course Id	Name	Credits	Teacher Id
10	daa	3	2
20	cn	3	3
30	co	3	3
40	dbms	3	1
50	sp	3	3
60	sp	3	3

PREPARING TIMETABLE.....

//Timetable will be displayed :

Cx, Ty: course with id x and teacher with id y

ATTEMPT 1:

Timetable for class fy:

	Slot1	Slot2	Slot3	Slot4	Slot5	Slot6	Slot7	Slot8
Day 1	-	C2,T5	C2,T5	C2,T5	(Break)	-	-	-
Day 2	-	-	-	-	(Break)	-	-	-
Day 3	-	-	C3,T6	-	(Break)	C1,T1	-	-
Day 4	-	C1,T1	C3,T6	-	(Break)	-	-	-
Day 5	-	C1,T1	-	C3,T6	(Break)	-	-	-

Timetable for class sy:

	Slot1	Slot2	Slot3	Slot4	Slot5	Slot6	Slot7	Slot8
Day 1	C30,T3	C20,T3	C10,T2	-	(Break)	-	-	-
Day 2	-	C20,T3	-	C60,T3	(Break)	C30,T3	-	-
Day 3	C40,T1	C10,T2	-	-	(Break)	-	-	-
Day 4	C50,T3	C50,T3	C50,T3	-	(Break)	C60,T3	C40,T1	-
Day 5	C10,T2	C30,T3	C40,T1	C60,T3	(Break)	-	C20,T3	-

Rearrange timetable? Press 'Y' for Yes and 'N' for No: N

Teacher details:

Rearrange timetable? Press 'Y' for Yes and 'N' for No: N
tt type 'Y' or 'y' else press 'N' or 'n' to end

//If you want to rearrange the

//To display timetable for particular teacher:

Enter the teacher id whose tt to be displayed(-1 to exit): 1

//Enter valid teacher id or it
will display error msg; to exit enter
id as -1

Rearrange timetable? Press 'Y' for Yes and 'N' for No: N

Teacher details:

Id	Name
1	Ghotkar
2	Jignesh M
3	A Sawant
4	Y Haribhakt
5	Jibi Abraham
6	M Khandekar

Enter the teacher id whose tt to be displayed(-1 to exit): 1

Timetable for Ghotkar:

	Slot1	Slot2	Slot3	Slot4	Slot5	Slot6	Slot7	Slot8
Day 1	-	-	-	-	(Break)	-	-	-
Day 2	-	-	-	-	(Break)	-	-	-
Day 3	C40,Y2	-	-	-	(Break)	C1,Y1	-	-
Day 4	-	C1,Y1	-	-	(Break)	-	C40,Y2	-
Day 5	-	C1,Y1	C40,Y2	-	(Break)	-	-	-

Enter the teacher id whose tt to be displayed(-1 to exit): -1

(Check Timetable.txt, Details.txt, Teachertt.txt for results)

pranay@pranay-HP-Pavilion-15-Notebook-PC:~/daa_mp\$
pranay@pranay-HP-Pavilion-15-Notebook-PC:~/daa_mp\$ █

Output stored in files(auto-generated):

1. **Details.txt**: Stores teacher details and all class-course details of the dept.
2. **Timetable.txt**: Stores timetables for all classes of the dept.
3. **Teachertt.txt**: Stores timetables for all teachers of the dept.

>>INTRO & ABSTRACT:

Abstract:

This project proposes a general solution for the timetabling problem. Most heuristic proposed earlier approaches the problem from the students' point of view. This solution, however, works from the teachers' point of view i.e. teacher availability for a given time slot. While all the hard constraints (e.g. the availability of teachers, etc.) are resolved rigorously, the scheduling solution presented in this paper is an adaptive one, with a primary aim to solve the issue of clashes of lectures and subjects, pertaining to teachers.

Introduction:

The class timetabling problem is a scheduling algorithm with great interest and implications in the

fields of operational research and artificial intelligence. The problem was first studied by Gotlieb, who formulated a class-teacher timetabling problem by considering that each lecture contained one group of students and one teacher, such that the combination of teacher and students can be chosen freely. Because of the size of real problem, almost all effective solutions are heuristic in nature, and do not guarantee optimality.

Here, an attempt is made to use heuristic approach to its best to generate timetables for all the classes of a particular department.

Clash avoidance and proper allocation of available resources is taken care of.

This solution generates timetable for all classes as well as all teachers of a department.

All these results are stored in files.

>>DESIGN MODEL:

Timetabling is known to be a non-polynomial complete problem i.e. there is no known efficient way to locate a solution.

Also, the most striking characteristic of NP-complete problems is that, no best solution to them is known. Hence, in order to find a solution to a timetabling problem, a heuristic approach is chosen.

This heuristic approach, therein, leads to a set of good solutions (but not necessarily the best solution).

Heuristic optimization methods are explicitly aimed at good feasible solutions that may not be optimal where complexity of problem or limited time available does not allow exact solution.

Generally, two questions arise (i) How fast the solution is computed? and (ii) How close the solution is to the optimal one?

Trade-off is often required between time and quality which is taken care of by running simpler algorithms more than once, comparing results obtained with more complicated ones and effectiveness in comparing different heuristics. The empirical evaluation of heuristic method is based on analytical difficulty involved in the problem's worst case result. In its simplest form the scheduling task consists of mapping class, teacher and room combinations (which have already been pre- allocated) onto time slots.

>>DESIGN TECHNIQUES USED:

Heuristic Approach is achieved by:

- Restricted Brute-Force,
- Backtracking
- Greedy Approach

Starting with first class of the department, courses are assigned in the empty slots depending on teachers' availability and following required constraints.

First class timetable can be easily generated because all slots are available.

Random selection of day and slot in that day is done and then availability of that slot is checked.

Then on the basis of timetable of first class, second class timetable is generated.

To solve the problems of clashes, backtracking is done to check teacher and slot clashes.

Similarly for third class on the basis of first and second class timetable and so on.

So, timetable of first class is produced more efficiently, decreasing the efficiency as we go to further classes.

In cases, if it is impossible to generate timetable due to infinite loop, after 50000 looping the error is flagged and program is stopped.

Heuristic approach does not guarantee best solution. Hence, option of regenerating timetable is provided to make choice.

Algo makett(dept)

```
{  
  
    initialize day_limit of all subjects to 0                //no. of classes * no. of subj * 5  
  
    initialize day_limit of all teachers to 0                //no. of teachers * 5  
  
    initialize all teachers timetable                        //no. of teachers * 5 * no. of slots  
  
    for each class in dept                                  //no. of classes times  
    {  
        initialize timetable of that class                  //5 * no_of_slots times  
  
        i = 0;  
        m = yr_opt;  
        for each subject in particular class                //no. of subjects in a class  
        {  
            j = 0;  
            k = 0;  
  
            loop = 0;  
            for all credits of particular subject           //tot credits of that subj  
            {  
  
                //max loop can run for 50000 times, if exceeded shows error and returns  
from func  
  
                if(loop > 50000)                             //restricted brute-force  
                {  
                    print("ERROR!");  
                    return 1;  
                }  
  
                generate random day k  
                if(day_limit of k == 0)  
                {  
                    generate random slot l of day k  
  
                    index = returnIndexOnId(d1, d1->years[m].subjects[i].t_id);  
  
                    //if lec is of 'T' type, no need of continuous slots  
                    if(type of subject Theory)  
                    {  
  
                        //backtracking to check slot availability  
                        check if for slot l of day k same teacher has occupied slot  
                        if yes, f = 0 else f = 1  
  
                        //check status of slot and flag f and day limit < 3  
                        if(slot is free && f && day limit of teacher <= 3)
```

```

        {
            assign all details to the slot l of day k
        }
    }

    //if course is of 'L' type, practicals need continuous slots
    else
    {
        f1 = 1;

        checks for free continuous credits no. of slots
        if yes, set f1 = 1
        else f1 = 0

        check if teacher is available at that slots
        if yes, f2 = 1
        else f2 = 0

        if(f1 && f2)
        {
            for all credits of practical
                assign details to that slot
        } //end of if(f1 && f2)
    } //end of else

    } //end of if
    loop++;
} //end of for

} //end of for

}

```

Constraints/Assumptions:

1. 5 working days a week.
2. Each teacher can take maximum 3 lectures per day.
3. One lecture can happen only once a day.
4. Labs happen in all continuous slots.
5. Labs are assumed to be performed by all students together.
6. No two teachers can have same id.
7. No two courses in a class can have same id.

Errors displayed if:

1. If total no. of credits more than total no. of slots in a week.
2. If teacher ids are duplicated.
3. If course ids of a class are duplicated.
4. If loop runs for along time due to improper mapping of teacher and course (loop > 50000)
5. If improper input filenames entered.
6. No. of slots more than 12

>>TEST CASES:

1. Normal Functioning:

As shown in Input/Output.

2. With errors:

```
***** TIMETABLE GENERATOR *****

Enter the Dept name: Computer

Enter no. of slots in a day(max 12): 5

Enter the teacher file: f1.txt

Teacher details:

    Id      Name
    1      Ghotkar
    2      Jignesh M
    3      A Sawant
    4      Y Haribhakt
    5      Jibi Abraham
    6      M Khandekar

Enter the class and course file: f2.txt

Class fy(1):
    Course Id  Name      Credits  Teacher Id
    1          daa        3          1
    3          jokes      3          6
    2          sp         3          5

Class sy(2):
    Course Id  Name      Credits  Teacher Id
    10         daa        3          2
    20         cn         3          3
    30         co         3          3
    40         dbms       3          1
    50         sp         3          3
    60         sp         3          3

                        PREPARING TIMETABLE.....
```



```
ERROR! Improper mapping of course & teacher data for class fy  
Timetable cannot be generated further
```

```
pranay@pranay-HP-Pavilion-15-Notebook-PC:~/daa_mp$ █
```

If total no. of slots in a week is less than total no. of credits for particular class

```
ERROR! Teacher id duplicated
```

```
pranay@pranay-HP-Pavilion-15-Notebook-PC:~/daa_mp$ █
```

*If teacher ids in file f1.txt is duplicated
(similarly for duplicate course ids in f2.txt)*

>>**APPLICATIONS:**

This solution helps to generate timetables for all courses of a particular dept and ease the hectic task of the teachers to map the courses in a week.

It can also be used to generate various other timetables like exam timetable, movie timetable, sports timetable, etc.

Many scheduling problems can be solved using this approach.

>>**LIMITATIONS OF MODEL:**

Heuristic approach is used for NP complete problem of Timetable generator. Therefore, there is no efficient algorithm for solution as such.

Thus, the solution obtained may not be the best solution but will follow all the constraints as mentioned.

The limitation is that we have to choose from the set of solutions a particular solution according to one's requirement.

Also the time complexity analysis says that it is not as expected efficient algorithm.

In case of infinite loop, we restrict loop activity till 50,000 and then break the loop. In such case, no solution is obtained as loop goes on continuously.
