# Opracowanie: 05i/2005i
# Workshops

**HISTORIA:**
- wersja 1.02: 2007.01.09, MK, spell-check, and description of test no 11
- wersja 1.01: 2007.01.09, MK, adding a paragraph about other solutions
- wersja 1.00: 2007.01.09, MK, reshaping the document to this state, making the proof more precise

dokument systemu SINOL 1.3.1

## 1 Introduction

The presented algorithm strongly relies on the assumption, that we are not allowed to accommodate two workshops in one room (otherwise the problem would be in NPC). But in present shape the problem may be solved in presented here greedy way.

## 2 Model solution

### 2.1 Algorithm description

Firstly, we sort our rooms by the length of time, at which they are at our disposal. Secondly, we iterate through all of our rooms starting from the one, that is the shortest available room and ending at the longest available one, trying to accommodate in each of them a suitable (that has not yet been allocated and is fitting both with time and the number of participants) workshop with the largest number of participants possible. Meanwhile, whenever we accommodate a workshop in a room, we decrease the count of open-air workshops by one and the number of people, who will take part in them by the number of participants of this particular workshop.

### 2.2 Correctness proof

**Lemma 1** *The algorithm described above finds an optimal (according to our problem) allocation of the workshops.*

Proof:
Let us assume, that solution given by our algorithm is not optimal. For some optimal solution let P be the first room (in the described before order) which assignment differs in this solution from assignment in our solution. We will consider such an optimal solution in which such room P is rented for maximal time possible from all rooms P in all optimal solutions. Consider three cases:

1. In our solution room P if free whilie in considered optimal solution it is assigned to an workshop. But room P is the first room which differs, and that means that even excluding the workshops rented to prior rooms there is a workshop, that can be assigned to room P. In this case our algorithm will assign a workshop to it. So it cannot be free. Contradiction.

2. In our solution we assigned room P to workshop A, but in considered optimal solution it is free. Then we can make a new optimal solution from considered optimal solution by moving workshop A to room P (and leaving the room assigned to A earlier, if there was any, empty). Therefore we obtain an optimal solution which agrees with our solution in all rooms up to room P. This contradicts the way in wich we have chosen considered optimal solution.

3. In our solution room P was assigned to workshop A, while in considered optimal solution it was assigned to workshop B. Instantly we get, that: $n_A \geq n_B$, where $n_W$ is the number of participants of workshop W. Let consider this two cases:

   - in the considered optimal solution workshop A has no room assigned. We can obtain a new optimal solution from the considered optimal solution by moving workshop B outdoors, and assigning workshop A to room P. As before this new optimal solution contradicts the way in which we have chosen the considered optimal solution.
   - in the considered optimal solution workshop A is assigned to room Q. Again we can obtain new optimal solution (which leads to contradiction) by exchanging workshops A and B.

The contradiction we have achieved ends the proof of the lemma.

Since each time we allocate a workshop in a room, we decrease appropriately the number of unallocated workshops (initially equal the number of all workshops) and the number of people, that will take part in open-air workshops (initially the number of all people participating in all workshops) - at the end of the algorithm we receive both values, that are asked for in our task.

# 3 Other solutions

Looking through the workshops in decreasing order by the number of participant and assigning them and assigning them rooms rented for shortest possible time gives the same result.

Other rational solutions have not been observed.

# 4 Tests

The tests are stored in the following files:

- *05i0.in* - example test
- *05i1.in* - correctness test, one workshop fitting in one room
- *05i2.in* - correctness test, one workshop too long to fit in one room
- *05i3.in* - correctness test, four workshops, three of them fitting in three rooms
- *05i4.in* - correctness test, nine workshops fitting in nine rooms
- *05i5.in* - correctness test, two workshops, one of them fitting in one of two rooms
- *05i6.in* - performance test, 10 trials, each containing 300 workshops and 300 rooms
- *05i7.in* - performance test, 10 trials, each containing 1000 workshops and 300 rooms
- *05i8.in* - performance test, 10 trials, each containing 300 workshops and 1000 rooms
- *05i9.in* - performance test, 10 trials, each containing 1000 workshops and 1000 rooms
- *05i10.in* - performance test, 10 trials, each containing 1000 workshops and 1000 rooms
- *05i11.in* – correctness test, 2 trials, the same number of people or the same time.

Every correctness test has been hand-written, every performance test has been randomly generated by the (included) gentest.cpp program.