

1 Wstęp

Algorytm silnie korzysta z faktu, że nie możemy umieścić 2 warsztatów w jednym pokoju (w przeciwnym wypadku mielibyśmy do czynienia z problemem NP-zupełnym). Na początku sortujemy pokoje po czasie, w którym nimi dysponujemy. Następnie przeglądamy je od najkrócej dostępnego do najdłużej dostępnego próbując każdorazowo dobrać do niego warsztat (który się w nim uda przeprowadzić - zarówno pod względem czasu trwania, jak i ilości uczestników oraz który nie został jeszcze przydzielony do żadnego pokoju) o jak największej ilości osób biorących w nim udział. Podczas działania algorytmu przy każdym przydzieleniu warsztatu do pokoju odpowiednio aktualizujemy (zmniejszamy) zapamiętaną liczbę warsztatów, które odbędą się na zewnątrz oraz liczbę osób biorących w nich udział.

Fakt

Zaprezentowany powyżej algorytm znajduje optymalne rozmieszczenie warsztatów w sensie naszego zadania.

Uzasadnienie Faktu

Przeładowujemy pokoje w opisany wyżej sposób, pamiętając, że są one odpowiednio posortowane: Jeżeli napotykamy na pokój P , dla którego moc zbioru W warsztatów, które mogą się w nim odbyć (spośród warsztatów jeszcze nie przydzielonych) jest większa od 0 zauważamy, że:

1. Nie przydzielając żadnego warsztatu z W do pokoju P nie uzyskamy lepszego rozwiązania, niż jakbyśmy go przydzielili - po prostu ograniczamy sobie w ten sposób liczbę pokoi, a ilość warsztatów pozostaje taka sama. Zatem któryś warsztat ze zbioru W powinien zostać ulokowany w pokoju P .

2. Ulokowanie w pokoju P warsztatu z W o największej liczbie uczestników jest optymalne - wszystkie warsztaty z W pasują pod względem czasu trwania do następnych pokoi po P (w kolejności przeglądania), zatem najlepiej jest ulokować w P warsztat z W o maksymalnej ilości zainteresowanych nim osób.

Jeżeli moc W dla rozpatrywanego pokoju P jest równa 0, przechodzimy do kolejnego pokoju - żaden z pozostałych warsztatów nie może zostać przeprowadzony w pokoju P , z kolei żadnemu przydzielonemu warsztatowi nie powinniśmy zmieniać pokoju na P co wynika z punktów 1 i 2 dla momentu, w którym ten warsztat ulokowywaliśmy.

Skoro przy każdym przydzieleniu warsztatu zmniejszamy liczbę pozostałych warsztatów (początkowo równą liczbie wszystkich warsztatów) i pomniejszamy liczbę osób zmuszonych do brania udziału w warsztatach na zewnątrz (początkowo równą liczbie wszystkich uczestników całej imprezy) otrzymujemy po zakończeniu działania algorytmu obie liczby, o które proszą nas w zadaniu.

2 Testy

Testy są przechowywane w następujących plikach:

- *test0* - test przykładowy
- *test1* - test poprawnościowy, jeden warsztat mieszczący się w jednym pokoju
- *test2* - test poprawnościowy, jeden warsztat zbyt długi aby zmieścić się w jednym pokoju
- *test3* - test poprawnościowy, cztery warsztaty, z których trzy mieszczą się w trzech pokojach
- *test4* - test poprawnościowy, dziewięć warsztatów mieszczących się w dziewięciu pokojach
- *test5* - test poprawnościowy, dwa warsztaty, z czego jeden mieści się w jednym z dwóch pokoi
- *test6* - test wydajnościowy, 10 prób, każda po 300 warsztatów i 300 pokoi
- *test7* - test wydajnościowy, 10 prób, każda po 1000 warsztatów i 300 pokoi
- *test8* - test wydajnościowy, 10 prób, każda po 300 warsztatów i 1000 pokoi
- *test9* - test wydajnościowy, 10 prób, każda po 1000 warsztatów i 1000 pokoi
- *test10* - test wydajnościowy, 10 prób, każda po 1000 warsztatów i 1000 pokoi

Każdy test poprawnościowy został napisany ręcznie, każdy test wydajnościowy został wygenerowany przez program *gentest.cpp* (dołączony)