Opracowanie: 05i/2005i
Warsztaty

HISTORIA:
- wersja 1.02: 2007.01.09, MK, sprawdzenie pisowni i dodanie opisy testu nr 11
- wersja 1.01: 2007.01.09, MK, dodanie akapitu o innych rozwiązaniach
- wersja 1.00: 2007.01.09, MK, przerobienie dokumentu do obecnej formy, uściślenie dowodu

dokument systemu SINOL 1.3.1

1 Wprowadzenie

Algorytm silnie korzysta z faktu, że nie możemy umieścić 2 warsztatów w jednym pokoju (w przeciwnym wypadku mielibyśmy do czynienia z problemem NP-zupełnym). A tak mamy do czynienia z problemem rozwiązywalnym zachłannie.

2 Rozwiązanie wzorcowe

2.1 Opis algotytmu

Na początku sortujemy pokoje po czasie, w którym nimi dysponujemy. Następnie przeglądamy je od najkrócej dostępnego do najdłużej dostępnego próbując każdorazowo dobrać do niego jedne warsztaty (które się w nim uda przeprowadzić – zarówno pod względem czasu trwania, jak i liczby uczestników oraz które nie zostały jeszcze przydzielone do żadnego pokoju) o jak największej liczbie osób biorących w nich udział. Podczas działania algorytmu przy każdym przydzieleniu warsztatów do pokoju odpowiednio aktualizujemy (zmniejszamy) zapamiętaną liczbę warsztatów, które odbędą się na zewnątrz oraz liczbę osób biorących w nich udział.

2.2 Dowód poprawności

Fakt 1 *Zaprezentowany powyżej algorytm znajduje optymalne rozmieszczenie warsztatów w sensie naszego zadania.*

Dowód:
Załóżmy nie wprost, że rozwiązanie znalezione przez nasz algorytm nie jest rozwiązaniem optymalnym. Niech dla pewnego optymalnego rozwiązania więc P będzie pierwszym pokojem (w porządku od najkrócej dostępnego do najdłużej dostępnego), co do którego w rozwiązaniu optymalnym podjęto inną decyzję niż w naszym rozwiązaniu. Rozważmy takie rozwiązanie optymalne, w którym taki pokój różniący się P jest jak najdłużej wynajętym pokojem. Rozważmy 3 następujące przypadki:

1. W naszym rozwiązaniu pokój pozostał wolny, ale w rozwiązaniu optymalnym temu pokojowi przyporządkowano pewne warsztaty. Zauważmy jednak, że pokój P jest pierwszym pokojem z różnicą i to, że w rozwiązaniu optymalnym przyporządkowano mu jakieś warsztaty, oznacza, że nawet wykluczywszy warsztaty już przyporządkowane można mu coś przyporządkować. Nasz algorytm zawsze w takiej sytuacji coś przyporządkowyje, więc pokój ten nie może być w naszym rozwiązaniu pusty. Sprzeczność.

2. W naszym rozwiązaniu pokojowi P przyporządkowano warsztaty A, zaś w rozwiązaniu optymalnym ten pokój pozostaje wolny. Tworzymy wtedy inne niegorsze (a więc też optymalne)

rozwiązanie, poprzez przesunięcie w naszym optymalnym rozwiązaniu warsztatów A do pokoju P (jeśli zajmowały jakiś inny pokój, ten pokój pozostaje pusty). Stworzyliśmy więc rozwiązanie optymalne, w którym wszystkie pokoje aż do P włącznie zgadzają się z naszym rozwiązaniem. Jest to sprzeczne z dokonanym wyborem rozwiązania optymalnego, które rozważamy.

3. W naszym rozwiązaniu pokojowi P przyporządkowano warsztaty A, zaś w rozważanym rozwiązaniu optymalnym przyporządkowano mu warsztaty B. W związku z naturą naszego algorytmu i tym, że poprzednie pokoje się zgadzają, mamy, że $n_A \geq n_B$, gdzie n_W to liczba uczestników warsztatów W. Mogą zajść dwa przypadki:

   - w rozważanym rozwiązaniu optymalnym warsztaty A nie mają przyporządkowanego pokoju. Tworzymy wtedy niegorsze rozwiązanie (a więc też optymalne) wyrzuciając warsztaty B na dwór, a pokojowi P przyporządkowujemy warsztaty A (wiemy już przecież, że się tam mieszczą). Powstałe rozwiązanie optymalne jest znów sprzeczne z wyborem rozwiązania optymalnego, które rozważamy.
   - w rozważanym rozwiązaniu optymalnym warsztaty A zostały przyporządkowane pokojowi Q. Od razu wiemy, że pokój Q jest wynajmowany na czas dłuższy niż pokój P. W związku z tym możemy zamienić w rozwiązaniu optymalnym miejscami warsztaty A i B i dostać inne rozwiązanie optymalne, w którym wszystkie pokoje, aż do P włącznie są zgodne z naszym rozwiązaniem, co jest sprzeczne z sposobem doboru rozpatrywanego rozwiązania optymalnego.

Osiągnięta sprzeczność kończy dowód faktu.

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.

3 Inne rozwiązania

Identyczne rozwiązanie daje przeglądanie po kolei warsztatów od najbardziej liczebnych do najmniej liczebnych i przyznawanie im pokoi o możliwie krótkim czasie. Nie wydaje się, aby istniały inne racjonalne rozwiązania.

4 Testy

Testy są przechowywane w następujących plikach:

- *05i0.in* - test przykładowy
- *05i1.in* - test poprawnościowy, jeden warsztat mieszczący się w jednym pokoju
- *05i2.in* - test poprawnościowy, jeden warsztat zbyt długi aby zmieścić się w jednym pokoju
- *05i3.in* - test poprawnościowy, cztery warsztaty, z których trzy mieszczą się w trzech pokojach
- *05i4.in* - test poprawnościowy, dziewięć warsztatów mieszczących się w dziewięciu pokojach
- *05i5.in* - test poprawnościowy, dwa warsztaty, z czego jeden mieści się w jednym z dwóch pokojów

- *05i6.in* - test wydajnościowy, 10 prób, każda po 300 warsztatów i 300 pokojów
- *05i7.in* - test wydajnościowy, 10 prób, każda po 1000 warsztatów i 300 pokojów
- *05i8.in* - test wydajnościowy, 10 prób, każda po 300 warsztatów i 1000 pokojów
- *05i9.in* - test wydajnościowy, 10 prób, każda po 1000 warsztatów i 1000 pokojów
- *05i10.in* - test wydajnościowy, 10 prób, każda po 1000 warsztatów i 1000 pokojów
- *05i11.in* - test poprawnościowy, 2 próby, takie same liczby chętnych lub takie same czasy

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