

## Żądanie projektowe nr 3

### Implementacja i analiza efektywności algorytmów optymalnych o pseudowielomianowej złożoności obliczeniowej dla wybranych problemów kombinatorycznych

Należy zaimplementować oraz dokonać pomiaru czasu działania algorytmów dla następujących problemów kombinatorycznych w wersji optymalizacyjnej:

- ❖ dyskretnego problemu plecakowego,
- ❖ asymetrycznego problemu komiwojażera.

#### 1. Dyskretny problem plecakowy

Parametrami zadania są: skończony zbiór elementów  $A = \{a_1, a_2, \dots, a_n\}$ , z których każdy ma określony rozmiar  $s(a_i) > 0$  i wartość  $w(a_i) > 0$  oraz pojemność plecaka  $b > 0$ . Rozwiązaniem jest taki podzbiór elementów  $A' \subset A$ , który maksymalizuje łączną wartość wybranych elementów ( $\sum_{a_i \in A'} w(a_i)$ ) przy warunku nie przekroczenia dopuszczalnej pojemności plecaka ( $\sum_{a_i \in A'} s(a_i) \leq b$ ). Należy przyjąć, że wszystkie parametry zadania są liczbami naturalnymi.

Dla tak zdefiniowanego problemu plecakowego należy opracować następujące algorytmy:

- przeгляд zupełny,
- algorytm zachłanny (jako kryterium wyboru elementu przyjąć jego wartość  $w(a_i)$  oraz stosunek wartości i rozmiaru  $w(a_i) / s(a_i)$ ),
- algorytm oparty na programowaniu dynamicznym.

#### 1.1. Założenia

- ❖ używane struktury danych powinny być alokowane dynamicznie (w zależności od aktualnego rozmiaru problemu),
- ❖ koszt krawędzi jest liczbą całkowitą,
- ❖ program powinien umożliwić weryfikację poprawności działania poszczególnych algorytmów. W tym celu powinna istnieć możliwość wczytania danych z pliku tekstowego (należy przyjąć następującą strukturę pliku: w pierwszej linii podajemy pojemność plecaka oraz liczbę przedmiotów, które chcemy w nim umieścić, w kolejnych liniach podajemy rozmiar i wartość dla poszczególnych przedmiotów – w jednej linii dla jednego przedmiotu)
- ❖ po zaimplementowaniu i sprawdzeniu poprawności działania każdego z algorytmów należy dokonać pomiaru czasu działania algorytmów w zależności od liczby przedmiotów  $N$  oraz rozmiaru plecaka  $B$ . Badania należy wykonać dla 5 różnych (reprezentatywnych) liczb przedmiotów  $N$ . Dla każdej wartości  $N$  badania trzeba wykonać dla 3 różnych pojemności plecaka  $B$  (jest to szczególnie ważne dla algorytmu wykorzystującego programowanie dynamiczne),

- ❖ dla każdego zestawu: algorytm, liczba przedmiotów  $N$  i pojemność plecaka  $B$  należy wygenerować po 100 losowych instancji (w sprawozdaniu należy umieścić tylko wyniki uśrednione),
- ❖ przy generowaniu danych testowych należy zapewnić warunek, aby sumaryczny rozmiar przedmiotów był większy niż pojemność plecaka (min. 25%),
- ❖ przy badaniach algorytmu wykonującego przegląd zupełny należy określić wartość  $N$ , dla którego algorytm wykonuje się w „rozsądnym” czasie (przyjąć  $t_{max}$  np. 10, 30 lub 60 minut),
- ❖ dokładnego pomiaru czasu w systemie Windows dokonujemy z wykorzystaniem funkcji *QueryPerformanceCounter* lub `std::chrono::high_resolution_clock`. Opis na stronie: <http://cpp0x.pl/forum/temat/?id=21331>,
- ❖ dopuszczalnymi językami programowania są języki kompilowane do kodu natywnego (np. C, C++), a nie interpretowane lub uruchamiane na maszynach wirtualnych (np. JAVA, .NET, Python),
- ❖ używanie okienek nie jest konieczne i nie wpływa na ocenę (wystarczy wersja konsolowa),
- ❖ z gotowych bibliotek np. STL, Boost lub innych można korzystać jedynie w uzasadnionych przypadkach; uzasadnienie musi zostać zaakceptowane przez prowadzącego,
- ❖ implementacja projektu powinna być wykonana w formie jednego programu,
- ❖ kod źródłowy powinien być komentowany.

## 2. Asymetryczny problem komiwojażera

Parametrami zadania są: skończony zbiór miast  $C = \{c_1, c_2, \dots, c_n\}$  oraz odległości  $d_{ij}$  z miasta  $c_i$  do miasta  $c_j$  ( $d_{ij} \neq d_{ji}$ ). Należy określić kolejność odwiedzania wszystkich miast (ich permutację)  $\langle c_{i[1]}, c_{i[2]}, \dots, c_{i[n]} \rangle$ , aby sumaryczna trasa była jak najkrótsza ( $\sum_{j=1}^{n-1} d_{i[j],i[j+1]} + d_{i[n],i[1]}$ ) przy założeniu, że każde miasto zostało odwiedzone dokładnie jeden raz. Należy przyjąć, że wszystkie parametry zadania są liczbami naturalnymi.

Dla tak zdefiniowanego problemu plecakowego należy opracować następujące algorytmy:

- D. przegląd zupełny,
- E. algorytm zachłanny (jako kryterium wyboru kolejnego miasta należy przyjąć minimalną odległość od ostatnio odwiedzonego miasta,
- F. algorytm przeszukiwania lokalnego (w najprostszej wersji można zastosować algorytm 2-opt. Algorytm zaczyna od losowej permutacji miast i w kolejnych krokach próbuje ją ulepszyć. Dla danej permutacji należy rozpatrzyć wszystkie permutacje uzyskane na jej podstawie przez zamianę dwóch niesąsiadujących ze sobą krawędzi tzw. zamiana dwu-krawędziowa. Spośród wszystkich uzyskanych w ten sposób permutacji wybieramy permutację dającą najlepszą poprawę. Permutacja ta staje się aktualnym rozwiązaniem. Algorytm jest kontynuowany do momentu, gdy możliwe jest uzyskanie lepszych rozwiązań. Szczegółowy opis algorytmu można znaleźć w literaturze, np. w [2]).

### 2.1. Założenia

- ❖ używane struktury danych powinny być alokowane dynamicznie (w zależności od aktualnego rozmiaru problemu),
- ❖ do reprezentacji odległości między miastami należy użyć macierzy sąsiedztwa,
- ❖ program powinien umożliwić weryfikację poprawności działania poszczególnych algorytmów. W tym celu powinna istnieć możliwość wczytania danych z pliku tekstowego (należy przyjąć następującą strukturę pliku: w pierwszej linii podajemy liczbę miast, w kolejnych liniach podajemy odległości z danego miasta do wszystkich pozostałych oddzielone spacją – w jednej linii z jednego miasta do wszystkich pozostałych),
- ❖ po zaimplementowaniu i sprawdzeniu poprawności działania każdego z algorytmów należy dokonać pomiaru czasu działania algorytmów w zależności od liczby miast  $N$ . Badania należy wykonać dla 5 różnych (reprezentatywnych) liczb miast  $N$ ,
- ❖ dla każdego zestawu: algorytm i liczba miast  $N$  należy wygenerować po 100 losowych instancji (w sprawozdaniu należy umieścić tylko wyniki uśrednione),
- ❖ przy badaniach algorytmu wykonującego przegląd zupełny przyjąć wartości  $N$  równe 10, 20 i 30. Następnie określić średni czas badania 1 permutacji. Na tej podstawie określić przewidywany czas działania algorytmu dla  $N$  równego 50, 100 oraz 1000,
- ❖ pozostałe założenia są identyczne jak dla problemu plecakowego.

### 3. Sprawozdanie

Sprawozdanie powinno zawierać:

- ❖ krótki wstęp zawierający oszacowanie złożoności obliczeniowej poszczególnych problemów na podstawie literatury,
- ❖ plan eksperymentu czyli założenia co do wielkości struktur, sposobu generowania ich elementów, sposobu pomiaru czasu, itp.
- ❖ opis metody losowego generowania grafu. Sposób powinien zapewnić spójność grafu,
- ❖ wyniki,
- ❖ wnioski dotyczące efektywności poszczególnych struktur. Wskazać (jeśli są) przyczyny rozbieżności pomiędzy złożonościami teoretycznymi a uzyskanymi eksperymentalnie,
- ❖ załączony kod źródłowy w formie elektronicznej (cały projekt wraz z wersją skompilowaną programu) oraz wydrukowane sprawozdanie.

### 4. Zadania i oceny

3,0 : 1A i 1B lub 2D i 2E

4,0 : 1A, 1B i 1C lub 2D, 2E i 2F

5,0 : 1A, 1B, 1C, 2D, 2E, 2F.

## 5. Termin złożenia projektu

Zadanie projektowe nr 1 należy złożyć do dnia **13 czerwca 2017r.** Każdy rozpoczęty tydzień spóźnienia oznacza obniżenie oceny uzyskanej z zadania projektowego o jeden stopień. Należy również pamiętać, że ocena końcowa z projektu (jako formy zajęć) liczona będzie jako średnia z sumy ocen z poszczególnych zadań projektowych z uwzględnieniem spóźnień. Oznacza to, że aby uzyskać ocenę dostateczną z projektu (formy zajęć) suma z trzech zadań projektowych z uwzględnieniem ewentualnych spóźnień, musi wynieść co najmniej 9,0. W przeciwnym przypadku projekt (forma zajęć) zostanie oceniony na ocenę niedostateczną.

## 6. Forma złożenia projektu

Patrz [zjo.iiar.pwr.wroc.pl/sdizo.html](http://zjo.iiar.pwr.wroc.pl/sdizo.html) **składanie projektów** (pdf)

## 7. Źródła

- [1] T. H. Cormen (i inni) , Wprowadzenie do algorytmów, WNT, Warszawa, 1997
- [2] M. Sysło, N. Deo, J.S. Kowalik, Algorytmy optymalizacji dyskretnej z programami w języku Pascal, WNT, Warszawa 1999
- [3] A. Drozdek, C++ Algorytmy i struktury danych, Helion, Gliwice, 2001
- [4] [http://eduinf.waw.pl/inf/alg/001\\_search/index.php](http://eduinf.waw.pl/inf/alg/001_search/index.php)

**Plagiaty będą surowo karane !!!**