

Zadania projektowe z przedmiotu „Struktury danych i złożoność obliczeniowa”

Prowadzący: prof. Adam Janiak, dr inż. Tomasz Krysiak, dr inż. Maciej Lichtenstein

I. Wprowadzenie

W ramach zajęć należy zrealizować niżej przedstawione zadania projektowe. Obowiązkowe są zadania od 1 do 4. Zadanie 5 jest dla tych, którzy chcieliby otrzymać ocenę celującą. Zadania realizowane są w grupach dwuosobowych (ewentualnie samodzielnie). Z wykonania każdego zadania należy sporządzić sprawozdanie. Zawartość sprawozdania jest opisana w ramach każdego zadania. Na pierwszej stronie sprawozdania MUSZA być podane nazwiska i numery indeksów autorów oraz przynależność do grup wraz z nazwiskiem prowadzącego (nazwiskami prowadzących).

Sprawozdanie należy oddać PROWADZĄCEMU WYKŁAD (w wersji papierowej, nie elektronicznej). Terminy oddawania sprawozdań są następujące (każdy tydzień spóźnienia oznacza obniżenie oceny o 1.0). :

1. 25 III 2010
2. 22 IV 2010
3. 6 V 2010
4. 1 VI 2010
5. 1 VI 2010

Ocena końcowa z projektu stanowi średnią arytmetyczną ocen z poszczególnych zadań (zaokrąglanie wg metody Round-to-Even). Nieoddanie jakiegoś sprawozdania (nawet po bardzo przekroczonym terminie) skutkuje oceną niedostateczną z całości projektu.

II. Zadania projektowe

1. Badanie efektywności algorytmów sortowania w zależności od liczby sortowanych elementów.

Należy zaimplementować oraz przeprowadzić pomiary czasu działania algorytmów sortowania liczb całkowitych (int) umieszczonych w tablicy. Należy zaimplementować następujące algorytmy: sortowanie bąbelkowe, sortowanie Shella, sortowanie przez scalanie, sortowanie przez kopcowanie, QuickSort, sortowanie kubełkowe, sortowanie introspektywne (introspektywne na ocenę celującą). Należy zmierzyć czasy działania powyższych algorytmów w zależności od liczby n sortowanych elementów. Pomiarów należy dokonywać wielokrotnie dla ustalonego n i wyznaczyć wartości średnie. Sortowanie Shella zbadać dla różnych wartości parametrów odległości (parametry *h-sortingu*). Sortowanie QuickSort zbadać dla różnych metod doboru elementu podziału i różnego typu danych wejściowych (elementy posortowane, odwrotnie posortowane itp.).

Sprawozdanie powinno zawierać informacje o sposobie pomiaru czasu, planie eksperymentu (rozważanych wartościach n), tabele/wykresy z pomiarami i wnioski dotyczące efektywności poszczególnych algorytmów oraz załączony kod programu w formie elektronicznej.

2. Badanie efektywności algorytmów grafowych w zależności od rozmiaru instancji oraz sposobu reprezentacji grafu w pamięci komputera.

Należy zaimplementować oraz dokonać pomiaru czasu działania następujących algorytmów grafowych:

- a. Algorytmy Prima oraz algorytm Kruskala wyznaczający Minimalne Drzewo Rozpinające
- b. Algorytm Dijkstry oraz algorytm Forda-Bellmana wyznaczający najkrótszą ścieżkę w grafie.

Algorytmy te należy zaimplementować dla obu poniższych reprezentacji grafu w pamięci komputera:

- Reprezentacja macierzowa (macierz incydencji)
- Reprezentacja listowa (lista następników/poprzedników)

Po zaimplementowaniu każdego z algorytmów dla obu reprezentacji należy dokonać pomiaru czasu działania algorytmów w zależności od rozmiaru grafu oraz jego gęstości (liczba krawędzi w stosunku do liczby wierzchołków) lub struktury (graf równoległy, szeregowy, równoległo-szeregowy, itp.).

Sprawozdanie powinno zawierać informacje o szczegółach implementacji algorytmów (np. zastosowanych metod detekcji cykli, sposobach przeszukiwania macierzy/list, itp.), planie eksperymentu (rodzajach i wielkościach grafów, dla których dokonywano pomiaru), tabele/wykresy z pomiarami oraz wnioski dotyczące efektywności algorytmów oraz kod programu w formie elektronicznej. Podobnie jak w punkcie 1, pomiarów należy dokonywać wielokrotnie dla ustalonego rozmiaru i typu grafu i wyznaczać wartości średnie.

3. Maszyna Turinga

Przy użyciu programu symulującego działanie DTM

(<ftp://sith.ict.pwr.wroc.pl/Informatyka/SDIZO/Turing>) należy napisać następujące trzy programy:

- a. Program wyznaczający sumę algebraiczną dwóch liczb binarnych zapisanych na taśmie i oddzielonych pojedynczym separatorem. Wynik powinien być zapisany po prawej stronie tych liczb i również oddzielony od nich pojedynczym separatorem. Liczby mogą być dowolnej (i różnej) długości (o różnej liczbie bitów). Dopuszczalny alfabet: {0,1,#}.
- b. Program sprawdzający, czy dany łańcuch jest palindromem. Łańcuch składa się z pojedynczego słowa dowolnej długości. Alfabet: {a,...,z,#}
- c. Program sprawdzający, czy dany łańcuch symboli zawiera w sobie inny łańcuch. Przy starcie na taśmie zapisany jest łańcuch szukany S i łańcuch przeszukiwany T (szukamy S w T) oddzielone pojedynczym separatorem (...#S#T#...). Łańcuchy mogą zawierać symbole alfabetu {a,...,z} i być dowolnej długości. Dopuszczalny alfabet: {a,...,z,#}.

Sprawozdanie powinno zawierać opis zastosowanych algorytmów, istotne fragmenty grafów stanów/przejęć, napotkane trudności podczas realizacji oraz wnioski. Należy również załączyć programy w wersji elektronicznej.

4. Badanie efektywności algorytmów pseudowielomianowych.

Należy zaimplementować i zmierzyć czas działania optymalnego algorytmu pseudowielomianowego dla wybranego problemu NP-trudnego. Do wyboru są trzy następujące problemy:

- a. Binarny problem plecakowy (max ocena 3.5)
- b. Problem szeregowania na 3 identycznych procesorach (max ocena 4.0)
- c. Problem szeregowania na m identycznych procesorach (m dowolne)

Sprawozdanie powinno być opracowane w zakresie podobnym do zadania 1 i 2.

5. * Opracowanie i analiza eksperymentalna algorytmów konstrukcyjnych dla wybranych problemów optymalizacji kombinatorycznej.

Zadanie polega na opracowaniu (wymyśleniu) kilku (3-4) algorytmów heurystycznych dla wybranego NP-trudnego problemu optymalizacji kombinatorycznej oraz przeprowadzenie analizy eksperymentalnej tych algorytmów. Szczegółowy zakres należy skonsultować prowadzącym projekt (lub z prowadzącym wykład).

III. Literatura

1. Cormen Thomas H., Leiserson Charles E., Rivest Ronald L., Stein Clifford, *Wprowadzenie do algorytmów*, WNT
2. Wirth N., *Algorytmy + struktury danych = programy*, WNT
3. Janiak A., *Wybrane problemy i algorytmy szeregowania zadań i rozdziału zasobów*, PLJ, 1999
4. R.J. Wilson, *Wprowadzenie do teorii grafów*, PWN, 2004
5. Janiak A, *Scheduling in Computer and Manufacturing Systems*, WKŁ, 2006