Autor:

DAWID PICHEN (132775)

PROJEKT Z TEORETYCZNYCH PODSTAW INFORMATYKI

Temat:

BINARNY PROBLEM PLECAKOWY

Wstęp

Celem niniejszego projektu była implementacja, analiza i wskazanie złożoności obliczeniowej binarnego problemu plecakowego. Problem plecakowy jest problemem optymalizacyjnym. Do dyspozycji mamy magazyn elementów, każdy element ma przypisany doń rozmiar oraz wagę. Problem znajduje taki podzbiór elementów z magazynu, dla których ich całkowita wartość jest maksymalna i zarazem elementy podzbioru mieszczą się w plecaku, którego rozmiar jest podany.

Opis programu

Badanie algorytmu polegało na mierzeniu czasu, jaki jest potrzebny na wyznaczenie podzbioru elementów, które mają trafić do plecaka. Badałem czas wyznaczenia podzbioru elementów dla następujących wielkości magazynów: 10, 50, 100, 500, 1000, 5000, 10000, 25000, 50000, 75000, 100000 elementów, oraz dla następujących pojemności plecaka: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. Każdy element w tablicy był reprezentowany w pamięci jako struktura zawierająca pola: rozmiar i waga. Każde z tych pól, było liczbą całkowitą bez znaku (*unsigned int*), liczba taka jest reprezentowana w pamięci w postaci 4 bajtów (kompilator GCC dla procesora i systemu 32-bitowego).

Postanowiłem napisać program testujący algorytm wyznaczania podzbioru elementów w języku ANSI C, gdyż zależało mi na możliwie szybkiej pracy programu. Aplikacja została napisana dla systemu operacyjnego Linux, jednakże nie powinno być większych problemów z jej skompilowaniem w innych systemach UNIXowych. Program składa się z 3 plików z kodami źródłowymi i 2 plików nagłówkowych. Plik *main.c* zawiera główną pętlę programu odpowiedzialną za uruchamiania algorytmu problemu plecakowego dla różnych rozmiarów magazynów, jak również dla różnych pojemności plecaka. Dla każdego rozmiaru magazynu i rozmiaru plecaka aplikacja wykonuje *L_PROB* prób wykonania algorytmu (domyślnie 10, definiowane w pliku *czas.h*). Dla każdego rozmiaru magazynu alokowana jest tablica struktur, zawierająca dane o elementach. Wagi i rozmiary elementów wybierane są losowo, ich maksymalne ograniczenie górne zdefiniowane jest poprzez definicje *OGR_WAGI* i *OGR_ROZM* w pliku *pp.h* (domyślne ograniczenia 20 i 20). Badanie algorytmu dla tych samych rozmiarów magazynów, ale różnych pojemności plecaka, wykonywane jest na takich samych elementach.

Plik *pp.c* zawiera implementację algorytmu plecakowego. Znajdują się w nim funkcje pozwalające zaalokować oraz zwolnić pamięć na tablicę roboczą algorytmu, obliczyć wartości elementów w tablicy roboczej, wyznaczyć elementy, który znajdą się w plecaku.

Wyznaczanie elementów plecaka, wykonywane jest rekurencyjnie. Tutaj znajduje się także funkcja wypełniająca magazyn wartościami losowymi, jak również funkcja wyświetlająca na ekranie zawartość magazynu, jak również zawartość tablicy roboczej. Domyślnie program nie wyświetla tablic roboczych, magazynów ani indeksów elementów, które znalazły się w plecaku. Jeśli jednak chcemy zobaczyć powyższe tablice, należy ustawić makrodefinicję *POKAZUJ_WSZYSTKO* w pliku *main.c*.

W pliku nagłówkowym *pp.h* oprócz nagłówków funkcji znajdują się także definicje typów strukturowych zawierających dane o elementach magazynu i elementach tablicy roboczej.

Z kolei plik *czas.c* zawiera funkcje związane z pomiarem czasu, jak również obliczaniem wartości średniej z *L_PROB* prób wykonywania algorytmu.
Zależało mi na jak najdokładniejszym pomiarze czasu. Nie mogłem użyć funkcji *time()*, gdyż funkcja ta zwracała czas tylko z dokładnością do sekundy. Postanowiłem wykorzystać w tym celu systemową funkcję *gettimeofday(struct timeval *tp, void *tzp)*. Jej pierwszym argumentem jest wskaźnik na strukturę *timeval* opisaną w pliku *sys/time.h*, struktura ta zawiera dwa składniki:
long tv_sec (liczba sekund, które upłynęły od początku epoki UNIXowej, czyli 01.01.1970)
long tv_usec (mikrosekundy).
Drugi argument to wskaźnik na strukturę opisującą strefę czasową. Dla pomiarów czasu jest on nieistotny (*NULL*). Widać, że funkcja systemowa pozwala na pomiar czasu z dokładnością do mikrosekund.
Program wyświetla wyniki czasów wykonania algorytmu na ekranie w sposób szczegółowy, tzn. dla każdej próby. Tworzy on także plik *wyniki.txt* w katalogu w którym został uruchomiony. Plik ten zawiera wyniki średnich czasów wykonań algorytmu dla każdego zadanego rozmiaru magazynu i pojemności plecaka.

Kompilacja programu

Dołączony do projektu dysk zawiera pliki źródłowe, jak również skompilowany plik wykonywalny programu. Uruchamiamy program poleceniem *projekt2* . Jeśli jednak chcemy skompilować program ze źródeł, to najpierw należy go wypakować z archiwum poleceniem:
tar -zxvf projekt2.tar.gz
następnie uruchamiamy kompilację poleceniem *make*.

Analiza złożoności obliczeniowej

W trakcie szukania rozwiązania binarnego problemu plecakowego, program wykonuje 3 ważne kroki. Pierwszym krokiem jest wypełnienie magazynu odpowiednimi wartościami. Dla każdego elementu generowana jest liczba odpowiadająca wadze i liczba odpowiadająca rozmiarowi. A zatem złożoność tego kroku zależna jest od rozmiaru magazynu (n) i wynosi $O(2n) = O(n)$. Jednakże przy pomiarze czasu nie biorę tego kroku pod uwagę, gdyż zakładam, że algorytm rozwiązania problemu plecakowego ma tylko znaleźć rozwiązanie dla wcześniej podanego magazynu i pojemności plecaka.

Drugim krokiem jest wypełnienie tablicy roboczej algorytmu. Tablica robacza jest zależna od pojemności plecaka oraz od rozmiaru magazynu. Dla każdego elementu tablicy wykonywanych jest maksymalnie pięć operacji przy ustalaniu wartości znacznika oraz wartości sumy elementów. Złożoność obliczeniowa tego kroku wynosi zatem $O(5\cdot(V+1)\cdot(n+1)) = O(n\cdot V)$.

Trzecim krokiem jest wyznaczenie elementów które trafią do plecaka. Jak widać, w programie wykonywana jest rekurencyjnie funkcja wypisująca elementy (*PokazElementyPlecaka()*), które mają zostać umieszczone w plecaku. Pierwsze wywołanie tej funkcji następuje dla ostatniego elementu w tabeli roboczej, tzn. elementu którego wiersz

równy jest wartości pojemności plecaka, a komórka równa się największemu indeksowi elementów w magazynie. Kolejnie rekurencyjne wywołania zależą od wartości znaczników, jak i aktualnych indeksów. Jednakże w najgorszym wypadku rekurencja ta wykona się n razy. Z taką sytuacją mielibyśmy do czynienia, gdyby wszystkie elementy były brane do plecaka. Zatem, złożoność obliczeniowa tego kroku, to O(n+1) = O(n).

Całkowita złożoność obliczeniowa binarnego problemu plecakowego wynosi: $O(n \cdot V + n) = O(n \cdot (V + 1)) = O(n \cdot V)$. Jest to złożoność pseudowielomianowa, gdyż zależy nie tylko od n, ale także o V.

Badanie czasów sortowań

Uruchomiłem program mierzący czasy algorytmów sortowania na następującym systemie komputerowym:

- procesor: Intel Pentium III 450 MHz,
- pamięć RAM: 160 MB SDRAM PC-100,
- system operacyjny: Slackware Linux 10,
- jądro systemu: 2.6.11.7.

Przed uruchomieniem programu zamknąłem wszystkie zbędne procesy, które potencjalnie mogłabym spowalniać pracę aplikacji, pozostawiłem tylko niezbędne procesy systemowe. Program został uruchomiony pod zwykłą konsolą tekstową (serwer X wyłączony). Po uruchomieniu programu uzyskałem następujące wyniki (załączone na dyskietce).

Otrzymane wyniki w tabeli:

Rozmiar magazynu [N]	**Średnie czasy dla pojemności plecaków V [s]**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
10	0,000018	0,000034	0,000050	0,000066	0,000081	0,000102	0,000112	0,000126	0,000143	0,000163
50	0,000074	0,000154	0,000238	0,000313	0,000399	0,000480	0,000561	0,000644	0,000733	0,000805
100	0,000151	0,000306	0,000474	0,000636	0,000796	0,000967	0,001135	0,001303	0,001470	0,001637
500	0,000733	0,001533	0,002371	0,003246	0,004086	0,004952	0,005831	0,006731	0,007633	0,008572
1000	0,001669	0,003301	0,005299	0,007243	0,009277	0,011248	0,013442	0,015635	0,017503	0,019755
5000	0,009862	0,022752	0,037190	0,051699	0,066684	0,081111	0,095619	0,109965	0,124427	0,138617
10000	0,022485	0,052350	0,087010	0,121600	0,156407	0,190797	0,225475	0,260100	0,294655	0,329293
25000	0,067212	0,156010	0,256603	0,357559	0,458453	0,559470	0,660221	0,761431	0,862517	0,963015
50000	0,146504	0,323121	0,520932	0,719148	0,916966	1,114402	1,312766	1,510651	1,708474	1,907439
75000	0,218058	0,482515	0,778150	1,074162	1,371823	1,670055	1,966643	2,296408	2,565431	2,865337
100000	0,289327	0,642419	1,040042	1,436687	1,834087	2,230910	2,629622	3,029897	3,429694	3,829811

Wnioski:

Otrzymane wyniki w pełni potwierdzają wyznaczoną wcześniej złożoność obliczeniową. Patrząc na wykresy obserwujemy, że zarówno wykres czasów średnich w zależności od pojemności plecaka przy ustalonej wielkości magazynu, jak i wykres czasów średnich w zależności od rozmiaru magazynu przy ustalonym rozmiarze placka, są wykresami liniowymi. Czasy średnie zależą od więc jednej zmiennej i od jednej stałej, co jest zgodne z wynikami badania złożoności obliczeniowej. Przypominam, że złożoność ta dla tego algorytmu wyniosła $O(n \cdot V)$.

Wybrane wykresy