

LUCRARE DE LABORATOR NR. 1

Tema: Analiza algoritmilor (Timpul de execuție al algoritmilor).

Scopul lucrării:

1. Analiza empirică a algoritmilor

Note de curs:

1. Timpul de execuție al algoritmilor.

De multe ori, pentru rezolvarea unei probleme, trebuie ales un algoritm dintre mai mulți posibili, două criterii principale de alegere fiind contradictorii:

- 1) algoritmul să fie simplu de înțeles, de codificat și de depanat;
- 2) algoritmul să folosească eficient resursele calculatorului, să aibă un timp de execuție redus.

Dacă programul care se scrie trebuie rulat de un număr mic de ori, prima cerință este mai importantă; în această situație, timpul de punere la punct a programului e mai important decât timpul lui de rulare, deci trebuie aleasă varianta cea mai simplă a programului.

Dacă programul urmează a fi rulat de un număr mare de ori, având și un număr mare de date de prelucrat, trebuie ales algoritmul care duce la o execuție mai rapidă. Chiar în această situație, ar trebui implementat mai înainte algoritmul mai simplu și calculată reducerea de timp de execuție pe care ar aduce-o implementarea algoritmului complex.

Timpul de rulare al unui program depinde de următorii factori:

- datele de intrare;
- calitatea codului generat de compilator;
- natura și viteza de execuție a instrucțiunilor programului;
- complexitatea algoritmului care stă la baza programului.

Deci timpul de rulare e o funcție de intrarea sa, de cele mai multe ori, nedepinzând de valorile de la intrare, ci de numărul de date.

În continuare vom nota cu $T(n)$ timpul de execuție al unui algoritm destinat rezolvării unei probleme de dimensiune n . Pentru a estima timpul de execuție trebuie stabilit un *model de calcul* și o unitate de măsură. Vom considera un model de calcul (numit și mașină de calcul cu acces aleator) caracterizat prin:

- Prelucrările se efectuează în mod secvențial.
- Operațiile *elementare* sunt efectuate în timp constant *indiferent* de valoarea operanzilor.
- Timpul de acces la informație nu depinde de poziția acesteia (nu sunt diferențe între prelucrarea primului element și cea a ultimului element al unui tablou).

A stabili o unitate de măsură înseamnă a stabili care sunt operațiile elementare și a considera ca unitate de măsură timpul de execuție a acestora. În acest fel timpul de execuție va fi exprimat prin numărul de operații elementare executate. Operațiile elementare sunt cele aritmetice (adunare, scădere, înmulțire, împărțire), comparațiile și cele logice (negație, conjuncție și disjuncție). Cum scopul calculului timpului de execuție este de a permite compararea algoritmilor, uneori este suficient să se contorizeze doar anumite tipuri de operații elementare, numite *operații de bază* (de exemplu în cazul unui algoritm de căutare sau de sortare se pot contoriza doar operațiile de comparare) și/sau să se considere că timpul de execuție a acestora este unitar (deși operațiile de înmulțire și împărțire sunt mai costisitoare decât cele de adunare și scădere în analiza se poate considera că ele au același cost).

Timpul de execuție al întregului algoritm se obține însumând timpii de execuție a prelucrărilor componente.

Importanța celui mai defavorabil caz. În aprecierea și compararea algoritmilor interesează în special cel mai defavorabil caz deoarece furnizează cel mai mare timp de execuție relativ la *orice* date de intrare de dimensiune fixă. Pe de altă parte pentru anumiți algoritmi cazul cel mai defavorabil este relativ frecvent.

În ceea ce privește analiza celui mai favorabil caz, aceasta furnizează o margine inferioară a timpului de execuție și poate fi utilă pentru a identifica algoritmi ineficienți (dacă un algoritm are un cost mare în cel mai favorabil caz, atunci el nu poate fi considerat o soluție acceptabilă).

Timp mediu de execuție. Uneori, cazurile extreme (cel mai defavorabil și cel mai favorabil) se întâlnesc rar, astfel ca analiza acestor cazuri nu furnizează suficientă informație despre algoritm.

În aceste situații este utilă o altă măsură a complexității algoritmilor și anume *timpul mediu de execuție*. Acesta reprezintă o valoare medie a timpilor de execuție calculată în raport cu distribuția de probabilitate corespunzătoare spațiului datelor de intrare.

2. Analiza empirică a complexității algoritmilor

O alternativă la analiza matematică a complexității o reprezintă *analiza empirică*.

Aceasta poate fi utilă pentru: (i) a obține informații preliminare privind clasa de complexitate a unui algoritm; (ii) pentru a compara eficiența a doi (sau mai mulți) algoritmi destinați rezolvării aceleiași probleme; (iii) pentru a compara eficiența mai multor implementări ale aceluiași algoritm; (iv) pentru a obține informații privind eficiența implementării unui algoritm pe un anumit calculator.

Etapale analizei empirice. În analiza empirică a unui algoritm se parcurg de regulă următoarele etape:

1. Se stabilește scopul analizei.
2. Se alege metrica de eficiență ce va fi utilizată (număr de execuții ale unei/unor operații sau timp de execuție a întregului algoritm sau a unei porțiuni din algoritm).
3. Se stabilesc proprietățile datelor de intrare în raport cu care se face analiza (dimensiunea datelor sau proprietăți specifice).
4. Se implementează algoritmul într-un limbaj de programare.
5. Se generează mai multe seturi de date de intrare.
6. Se execută programul pentru fiecare set de date de intrare.
7. Se analizează datele obținute.

Alegerea măsurii de eficiență depinde de scopul analizei. Dacă, de exemplu, se urmărește obținerea unor informații privind clasa de complexitate sau chiar verificarea acurateții unei estimări teoretice atunci este adecvată utilizarea numărului de operații efectuate. Dacă însă scopul este evaluarea comportării implementării unui algoritm atunci este potrivit timpul de execuție.

Pentru a efectua o analiză empirică nu este suficient un singur set de date de intrare ci mai multe, care să pună în evidență diferitele caracteristici ale algoritmului. În general este bine să se aleagă date de diferite dimensiuni astfel încât să fie acoperită plaja tuturor dimensiunilor care vor apare în practică. Pe de altă parte are importanță și analiza diferitelor valori sau configurații ale datelor de intrare. Dacă se analizează un algoritm care verifică dacă un număr este prim sau nu și testarea se face doar pentru numere ce nu sunt prime sau doar pentru numere care sunt prime atunci nu se va obține un rezultat relevant. Același lucru se poate întâmpla pentru un algoritm a cărui comportare depinde de gradul de sortare a unui tablou (dacă se alege fie doar tabloul aproape sortate după criteriul dorit fie tablouri ordonate în sens invers analiza nu va fi relevantă).

În vederea analizei empirice la implementarea algoritmului într-un limbaj de programare vor trebui introduse secvențe al căror scop este monitorizarea execuției. Dacă metrica de eficiență este numărul de execuții ale unei operații atunci se utilizează un contor care se incrementează după fiecare execuție a operației respective. Dacă metrica este timpul de execuție atunci trebuie înregistrat momentul intrării în secvența analizată și momentul ieșirii. Majoritatea limbajelor de programare oferă funcții de măsurare a timpului scurs între două momente. Este important, în special în cazul în care pe calculator sunt active mai multe taskuri, să se contorizeze doar timpul afectat execuției programului analizat. În special dacă este vorba de măsurarea timpului este indicat să se ruleze programul de test de mai multe ori și să se calculeze valoarea medie a timpilor.

După execuția programului pentru datele de test se înregistrează rezultatele iar în scopul analizei fie se calculează mărimi sintetice (media, abaterea standard etc.) fie se reprezintă grafic perechi de puncte de forma (dimensiune problema, măsură de eficiență).

SARCINA DE BAZĂ:

1. Implementați algoritmi propuși într-un limbaj de programare
2. Stabiliți proprietățile datelor de intrare în raport cu care se face analiza

3. Alegeți metrica pentru compararea algoritmilor
4. Efectuați analiza empirică a algoritmilor propuși
5. Faceți o prezentare grafică a datelor obținute
6. Faceți o concluzie asupra lucrării efectuate.

Întrebări de control:

1. Enumerați factorii ce influențează timpul de execuție al algoritmului.
2. Care sunt etapele analizei empirice?
3. În care cazuri se face analiza empirică a algoritmilor?