Cviko z Počítačové algebry

Verze ze dne 7. října 2024

1 Asymptotická složitost

Složitost obecně

Úloha 1.1. Uvažujme reálné matice o rozměru $n \times n$. Jakou časovou složitost má

(a) nalezení největšího prvku matice,

(b) výpočet stopy matice,

(c) vynásobení dvou matic školským algoritmem,

(d) výpočet determinantu matice přímo z definice,

(e) výpočet determinantu matice rekurzivním rozvíjením podle prvního řádku,

(f) výpočet determinantu matice pomocí LU rozkladu matice,

(g) výpočet inverzní matice školským algoritmem,

(h) výpočet inverzní matice pomocí adjungované matice, kde všechny minory počítáme pomocí LU rozkladu.

Úloha 1.2. Pro určení charakteristického polynomu $p_A(\lambda) = \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0$ reálné $n \times n$ matice A je možné použít tzv. *Faddejev-Leverrierův algoritmus*, který funguje následovně:

1. Vytvoříme posloupnost matic M_k, kde $M_0 = 0$ a další matice se počítají rekurentně: $M_k = AM_{k-1} - c_{n-k+1}I$
2. Koeficienty c_k se vypočítávají jako $c_{n-k} = -\frac{1}{k}\operatorname{tr}(AM_k)$ (přičemž $c_n = 1$).

Určete časovou složitost tohoto algoritmu; předpokládejte kubickou složitost násobení matic. (Jako bonus si můžete rozmyslet, že algoritmus opravdu funguje.)

Master theorem

Úloha 1.3. Odhadněte pomocí Master theoremu časovou složitost binárního vyhledávání v setříděném poli (ano, je to kanón na vrabce). Jak se výsledek změní, pokud bychom uvažovali *ternární* vyhledávání, tj. krájeli pole na třetiny?

Úloha 1.4. Předpokládejme (pro jednoduchost), že n je mocnina dvojky. Emil chce násobit dvě $n \times n$ matice A, B tak, že se na ně podívá jako na blokové matice a tak je i vynásobí:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad AB = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix}$$

(zde bloky mají vždy poloviční velikost oproti původní matici). Postupuje navíc rekurzivně, takže každý ze součinů bloků opět spočte rozdělením na poloviční bloky. Jakou časovou složitost bude toto násobení matic mít, jestliže

(a) vždy spočte přesně ty součiny bloků, které potřebuje,

(b) bude kapánek natvrdlý a spočte vždy pro jistotu všech 16 možných součinů bloků z A a B?

Úloha 1.5. Navrhněte algoritmus typu „rozděl a panuj“, který v reálné matici $n \times n$ najde (nějaké) lokální minimum, tj. prvek, který bude menší než všichni jeho sousedé (pro jednoduchost předpokládáme všechny prvky různé). Jakou bude mít časovou složitost? Jakou časovou složitost by měl algoritmus „kutálení z kopce“, kdy začneme v jednom z rohů a v každém kroku skočíme na menšího souseda, pokud existuje?

Výsledky:

1.1. ***(a)*** $O(n^2)$ ***(b)*** $O(n)$ ***(c)*** $O(n^3)$ ***(d)*** $O(n!)$ ***(e)*** $O(n!)$ ***(f)*** $O(n^3)$ ***(g)*** $O(n^3)$ ***(h)*** $O(n^5)$

1.2. $O(n^4)$

1.3. $O(\log n)$, nijak se nezmění

1.4. ***(a)*** $O(n^3)$ ***(b)*** $O(n^4)$

1.5. složitost $O(n)$; kutálení má složitost $O(n^2)$