

Algorytmy i Struktury danych (2025)

Lista zadań 2

1. Ile dokładnie potrzeba mnożeń, by obliczyć:

(a) $\vec{a} \cdot \vec{b} = \sum_{i=1}^n a_i b_i$ – iloczyn skalarny dwóch wektorów rozmiaru n .

(b) $W(x) = \sum_{i=0}^n a_i x^i$ – wartość wielomianu stopnia n w punkcie x ,

(i) jeśli nie stosuje się schematu Hornera, (ii) stosując schemat Hornera.

(c) Współczynniki wielomianu będącego iloczynem dwóch wielomianów stopnia $n-1$.

(d) Iloczyn dwóch macierzy $n \times n$, czyli macierz o współczynnikach $c_{ij} = \sum_{k=0}^n a_{ik} b_{kj}$.

(e) Wyznacznik macierzy $n \times n$ metodą eliminacji Gaussa.

Dla każdego przypadku napisz jakiej klasy $\Theta(n^k)$, lub $\Theta(n^k \log n)$ są to funkcje.

2. Wyznacz ciąg wszystkich współczynników dwumianowych Newtona $\binom{n}{0}, \binom{n}{1}, \dots, \binom{n}{n}$ używając tylko $\lfloor \frac{n}{2} \rfloor$ mnożeń i $\lfloor \frac{n}{2} \rfloor$ dzielen liczb całkowitych. Wsk. $\binom{n}{k} = \binom{n}{k-1} \cdot (n-k+1) : k$.

3. Ile porównań na pewno wystarczy, by połączyć dwa rosnące ciągi długości n w jeden.

4. Sortowanie przez selekcję. Rozważ algorytm sortujący o następujących krokach:

(a) znajdź w tablicy n -elementowej element największy i zamień go z ostatnim.

(b) znajdź wśród $n-1$ początkowych elementów tablicy element największy i zamień go z elementem przedostatnim.

...

(c) znajdź wśród 2 początkowych elementów większy i zamień go z drugim.

Ile dokładnie porównań wykona ta procedura? Jakiej klasy $\Theta(n^k)$ jest to funkcja.

5. Dowiedz się na czym polega algorytm sortowanie przez wstawianie. Ile porównań, między elementami tablicy wykona algorytm, w najgorszym przypadku (gdy tablica jest uporządkowana malejąco), a ile w najlepszym (gdy tablica jest uporządkowana rosnąco), zakładając, że algorytm jest napisany w wersji z wartownikiem, czyli zakłada, że przed tablicą (w elemencie $t[-1]$) znajduje się $-\infty$, co pozwala nie sprawdzać zakresu, tylko napisać w pętli wewnętrznej `while(t[i-1]>t[i]) {std::swap(t[i],t[i-1]);i--}`?

Powiąz dokładnym wzorem liczbę porównań wykonanych przez algorytm z liczbą inwersji w wyjściowej tablicy. Inwersją nazywamy każdą parę indeksów i, j , taką że $i < j$ ale $t[i] > t[j]$. Wywnioskuj dlaczego sortowanie przez selekcję jest bardzo rzadko używane.

6. Bazując na implementacji listy jednokierunkowej, podanej na wykładzie napisz funkcje:

(a) `*node append(node *a, node* b)`; podpinającą listę b na koniec listy a , i zwracającą początek nowo powstałej listy. Dopilnuj by funkcja działała poprawnie również w przypadku, gdy któraś z list jest pusta.

(b) `void filter(node *&L, bool (*f)(int))`; usuwającą z listy L elementy nie spełniające warunku zadanego funkcją f .

(c) `void insert_after_smaller(node* &L, int x)`; która wstawia element x do posortowanej listy jednokierunkowej za wszystkimi elementami mniejszymi od x .

(e) `node* merge(node *a, node* b)`; scalającą dwie listy posortowane w jedną.

(d) `std::vector<node*> ascending_fragments(node* L)`; poprzecina listę L na rosnące podciągi i zwróci wektor wskaźników na początki tych list.