

Algorytmy i Struktury danych (2021)

Lista zadań 1

1. Napisz funkcję obliczającą x^n za pomocą $O(\log n)$ mnożeń: (a) z użyciem rekurencji i (b) bez użycia rekurencji. Uwaga: nie możesz używać funkcji `pow`, `log`, `exp` itp. a jedynie operator mnożenia. Wskazówka: skorzystaj z faktu, że dla parzystych n zachodzi: $x^n = (x^2)^{n/2}$ lub $x^n = (x^{n/2})^2$.
2. Dana jest funkcja `double f(double)` ciągła, taka że $f(0) < 0 < f(1)$. Napisz program, który metodą bisekcji znajdzie pierwiastek funkcji f (taki x , że $f(x) = 0$). Uwaga: może się zdarzyć, że taki x nie istnieje, więc algorytm powinien znajdować taki x , dla którego $f(x)$ jest najbliższemu zera. Warunkiem zakończenia pętli uczyni wykrycie zapętlenia (czyli że końce przedziału przestały się zbliżać do siebie).
3. Schemat Hoernera: (a) Pokaż, że wystarczy dokładnie n mnożeń, aby wyliczyć wartość wielomianu stopnia n , o współczynnikach zawartych w tablicy `a`?

$$W(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

w ten sposób, że `a[0]=a0`, `a[1]=a1` itd.. Zakładamy, że nie ma dostępu do funkcji `pow(x,n)` obliczającej x^n i potęgowanie należy wykonać za pomocą mnożenia. Nie używamy funkcji z zadania 1.

(b) Napisz funkcję `double oblicz(double a[], int n, double x)` realizującą twój algorytm. Pamiętaj, że do przechowania wielomian stopnia n ma $n + 1$ współczynników więc rozmiar tablicy `a[]` jest o jeden większy od stopnia wielomianu $W(x)$.

4. (a) Jak znaleźć minimum i maksimum n liczb nie używając więcej niż $3n/2$ porównań?
(b) Napisz funkcję, `void max_min(int a[], int n)`, która zrealizuje ten algorytm i wypisze wartości `max` i `min` na standardowym wyjściu.
5. W jednej linii pliku `a.txt` znajdują się oddzielone spacjami stopień i współczynniki wielomianu $A(x)$ w kolejności: $n \ a_0 \ a_1 \ a_2 \ a_n$ a w pliku `b.txt` stopień i współczynniki wielomianu $B(x)$. Napisz w C++ program, który odczyta dane z tych plików, obliczy stopień i współczynniki wielomianu $C(x) = A(x)B(x)$, a następnie zapisze je w pliku `c.txt`. Jak ilość mnożeń wykonywanych przez twój program zależy od stopni wielomianów $A(x)$ i $B(x)$.
6. Napisz program nie zawierający instrukcji `if` ani `switch`, który policzy ile razy występuje każdy znak ASCII w pliku o nazwie podanej jako argument programu (`argv[1]`).

7.

```
struct lnode
{
    int key;
    lnode *next;
    lnode(int key0, lnode* next0=nullptr):key(key0),next(next0){}
};
```

to struktura węzła listy jednokierunkowej, z dodanym konstruktorem. Napisz funkcję:

- (a) `int wypisz(lnode* L)`, która wypisze elementy listy `L` oddzielone spacjami.
- (b) `int suma(lnode* L)`, która obliczy sumę elementów listy `L`.
- (c) `int nty(int n, lnode *L)` której wynikiem jest wartość n -tego elementu listy `L` lub 0, jeśli długość listy jest mniejsza niż n .
- (d) `void insert(lnode *&L, int x)`, która wstawi `x` na początek listy `L`.
- (e) `void insert_after_smaller(lnode *&L, int x)`, która wstawi `x` do listy `L` za wszystkimi elementami od niego mniejszymi.

- (f) `void remove(lnode* &L, int x)`, która usunie z listy L elementy o wartości `x`.
- (g) `void filter(lnode* &L, bool(*cond)(int))`, która usunie z listy L klucze `x` dla których `cond(x)==false`.
- (h) `void destroy(lnode* &L)`, która usunie wszystkie elementy z listy L.
- (i) `void reverse(lnode* &L)`, która odwróci kolejność elementów listy L zmieniając jedynie zmienną L oraz wskaźniki `next` w węzłach.