Algorytmy i Struktury danych (2022)

Lista zadań 1

1. Napisz funkcję obliczającą x^n za pomocą dokładnie $\lceil \log_2 n \rceil$ 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 najbliżej zera. Warunkiem zakończenia pętli uczyń wykrycie zapętlenia (czyli że końce przedziału przestały się zbliżać do siebie).

3. Shemat Hoernera: (a) Przed odpowiednie wyłączenia x przed nawias, pokaż, że wystarczy **dokładnie** n mnożeń, aby wyliczyć wartość wielomianu stopnia n?

$$W(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

   Wskazówka: zadanie wykonaj kolejno dla n=0, 1, 2, 3, 4 a potem uogólnij pisząc odpowiedni algorytm.
   (b) Napisz funkcję `double oblicz(std::vector<double> a, double x)` realizującą twój algorytm, zakładając, że wektor `a` zawiera n+1 współczynników $[a_0, a_1, ..., a_n]$ czyli tak, że `a[i]`$= a_i$ jest współczynnikiem przy przy x^i. Oczywiście nie możesz korzystać z żadnej funkcji obliczającej potęgę. Aby wyliczyć $3x^2 + 5x + 4$ dla $x = 5$ napisz: `cout << oblicz({4,5,3}, 5) << endl;`

4. Jak znaleźć minimum i maksimum n liczb nie używając więcej niż $3n/2$ porównań? Wskazówka: każde dwie kolejne liczby porównuj najpierw między sobą. Znajdź wzory na optymalną liczbę porównań dla n parzystych i dla n nieparzystych.

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`. Jeśli do zapamiętania współczynników użyjesz `std::vector`, to twoja funkcja `main` może składać się z jednej linii:

```
zapisz("c.txt",iloczyn(wczytaj("a.txt"),wczytaj("b.txt")));
```

   Jak ilość mnożeń wykonywanych przez twój program zależy od stopni wielomianów $A(x)$ i $B(x)$?

6. Korzystając z tablicy liczników `int ile[256]={0};` 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]`) i wyniki wypisze na standardowym wyjściu. Wskazówka w języku C++ `unsigned char` jest liczbą z przedziału `[0,255)`. Aby wczytać jeden znak z pliku bez pomijania spacji użyj `istream::get()` lub `istream::get(char&)`.

7. Dana jest struktura węzła listy jednokierunkowej, z dodanym konstruktorem.

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

Napisz funkcję (0.5pkt za każdą):

(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.