Algorytmy i Struktury danych (2023)

Lista zadań 1

1. Wyraź w notacji O złożoność następujących procedur:
   (a) Sprawdzenie czy liczba n ma dzielnik większy od 1 ale mniejszy od n.
   (b) Sprawdzenie czy liczba n ma dzielnik większy od 1 ale mniejszy lub równy $\sqrt{n}$.
   (c) Wyznaczenie wszystkich liczb pierwszych z przedziału $1..n$ algorytmem Erastotenesa (podziel wynik przez n). Skorzystaj z faktu, że

$$\sum_{k=1}^{n} \frac{1}{k} < \ln n + 1$$

   (d) Wyznacz numeryczne proporcje otrzymanych wyników (a)/(b) i (b)/(c) dla $n = 10^6$.

2. Wyraź w notacji O ile dodawań wykonasz wyznaczając n początkowych liczb Fibonacciego:
   (a) Za pomocą procedury typowej rekurencyjnej.
   (b) Za pomocą procedury iteracyjnej wywoływanej dla każdej liczby osobno.
   (c) Za pomocą procedury iteracyjnej wyznaczającej wszystkie liczby w jednej pętli.

3. 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 $x^n = (x^{n/2})^2$.

4. Dana jest funkcja `double f(double)` ciągła, taka że $f(0) < 0 < f(1)$. Napisz program, który metodą bisekcji stopniowo zawężając granice przedziału, znajdzie miejsce zerowe funkcji f (czyli taki x, że $f(x) = 0$). Uwaga: może się zdarzyć, że taki x nie istnieje, więc algorytm powinien znajdować taki x, w pobliżu którego $f(x)$ zmienia znak. Warunkiem zakończenia pętli uczyń wykrycie sytuacji, że środek przedziału pokrywa się z jednym z końców.

5. Schemat 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 $18x^5 + 3x^2 + 5x + 4$ dla $x = 10$ napisz: `cout << oblicz({4,5,3,0,0,18}, 10) << endl;`

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

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

   Napisz funkcje (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, zakładając, że liczby na liście ustawione są rosnąco.

(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`.

7. Napisz funkcję `void reverse(lnode* &L)`, która odwróci kolejność elementów listy `L` zmieniając jedynie zmienną `L` oraz wskaźniki `next` w węzłach. Funkcja **nie** korzysta z przydziału i zwalniania pamięci na stercie, czyli operatorów z `new` oraz `delete` ani ich odpowiedników.

8. (2pkt) Napisz procedurę `lnode* merge(lnode* L1, lnode* L2)`, która złączy posortowane listy `L1` i `L2` w jedną posortowaną listę, zmieniając jedynie wartości pól `next` i używając tylko dwóch dodatkowych wskaźników. Ilość porównań nie powinna przekroczyć długości wynikowej listy. Po wykonaniu algorytmu listy L1 i L2 są zniszczone, gdyż ich węzły należą już do listy wynikowej.