

## Algorytmy lista 1

1. Napisz funkcję obliczającą  $x^n$  w czasie  $O(\log n)$  w wersjach z użyciem i bez użycia rekurencji.
2. Dana jest funkcja `double f(double)` ciągła, taka że  $f(0) < 0 < f(1)$ . Napisz program, który metodą bisekcji znajdzie pierwiastek funkcji `f`. Warunkiem zakończenia pętli uczyni wykrycie zapętlenia :).
3. Ile potrzeba mnożeń, aby wyliczyć wartość wielomianu stopnia  $n$ , o współczynnikach zawartych w tablicy `a`?  
Napisz funkcję `double oblicz(double a[], int n, double x)` realizującą twój algorytm.
4. Napisz funkcję, `P(int k, int n, double p)` która wyliczy szansę na to, że w  $n$  próbach będzie nie więcej niż  $k$  sukcesów, jeśli prawdopodobieństwo pojedynczego sukcesu wynosi  $p$ .

$$P = \sum_{i=0}^k \binom{n}{i} p^i q^{(n-i)}$$

Wykonanie funkcji nie może wymagać więcej niż  $3k + \log n$  mnożeń.

5. Napisz funkcję, `int nty(int n, lnode *l)` której wynikiem jest wartość  $n$ -tego elementu listy  $l$ , lub 0 jeśli długość listy jest mniejsza niż  $n$ . zakładamy że `lnode` jest zdefiniowana jako

```
struct lnode
{ int key;
  lnode *next;
}
```

6. Jak znaleźć minimum i maksimum  $n$  liczb nie używając więcej niż  $3n/2$  porównań?
7. W pliku "a.txt" znajdują się współczynniki wielomianu  $a(x)$  w następującej postaci:  $(a_0, a_2, a_3, \dots, a_n)$  a w pliku "b.txt" współczynniki wielomianu  $b(x)$   $(b_0, b_2, b_3, \dots, b_n)$ . Napisz program, który odczyta dane z tych plików, obliczy iloczyn wielomianów  $c(x) = a(x)b(x)$  a następnie współczynniki wielomianu  $c(x)$  wpisze do pliku "c.txt".
8. Stwórz klasę `ułamek` z polami `licznik` i `mianownik`. Powinien to być zawsze ułamek nieskracalny. Zaimplementuj 4 działania (jako operatory `+` `-` `*` `/`) jednoargumentowy operator `-` oraz czytanie ze strumienia

i pisanie do strumienia ułamków typu  $1/2$ ,  $4/5$ ,  $77/3$  itp. Napisz też konstruktor dwuargumentowy, gdzie drugi argument ma wartość domyślną 1.

9. Napisz program nie zawierający instrukcji `if` ani `switch`, który policzy ile razy występuje każdy znak ASCII w pliku podanym jako argument programu.
10. Przepisz klasy: `kolejka`, `stos`, `kolejka_priorytetowa` jako szablony.
11. Napisz procedurę `sito_erasstenesa` znajdującą wszystkie liczby pierwsze mniejsze od  $n$ . Wynikiem jest tablica `bool prime[n]`, taka że `prime[k]==true` wtedy i tylko wtedy gdy  $k$  jest liczbą pierwszą.