

# Arytmetyka Wielkich Liczb

Adrian Urbański

22 lutego 2019

## 1 Biblioteki i API

Wyłącznie biblioteka standardowa

## 2 Moduły projektu

- Plik nagłówkowy `bignum.h` zawierający strukturę `bignum` reprezentującą dużą liczbę
- Moduł obliczenia zawierający funkcje wykonujące operacje arytmetyczne
- Moduł konwersja zawierający funkcje odpowiedzialne za wczytywanie i wypisywanie liczb
- Moduł struktury zawierający implementację kolejki oraz stosu potrzebnych do parsera
- Moduł parser odpowiedzialny za obliczenie wyrażenia zgodnie z kolejnością wykonywania działań

## 3 Algorytmy i struktury danych

- Struktura reprezentująca liczbę - dynamicznie alokowana pamięć, zmienne `size`, `maxsize`, `sign`
- Kolejka oraz stos zaimplementowane przy pomocy listy
- Operacje arytmetyczne
- Algorytm Shunting-yard oraz kalkulator ONP wykorzystywane w parserze

## 4 Opis funkcji

### 1. Moduł obliczenia

- `bignum add(bignum a, bignum b)` - przyjmuje dwa obiekty typu `bignum` i zwraca nowy którego wartość jest sumą argumentów
- `bignum sub(bignum a, bignum b)` - przyjmuje dwa obiekty typu `bignum` i zwraca nowy którego wartość jest różnicą argumentów
- `bignum mult(bignum a, bignum b)` - przyjmuje dwa obiekty typu `bignum` i zwraca nowy którego wartość jest iloczynem argumentów
- `bignum divi(bignum a, bignum b)` - przyjmuje dwa obiekty typu `bignum` i zwraca nowy którego wartość jest ilorazem argumentów

### 2. Moduł konwersja

- `bignum fromInt(int a)` - przyjmuje zmienną `int` i zwraca obiekt `bignum` którego wartość jest równa wartości argumentu
- `int toInt(bignum a)` - przyjmuje obiekt `bignum` i zwraca zmienną `int` której wartość jest równa wartości argumentu
- `bignum fromArray(int *a, int size)` - przyjmuje tablicę zmiennych `int` o wartościach z przedziału 0-9 reprezentującą kolejne cyfry liczby oraz zmienną `int` oznaczającą jej wielkość i zwraca obiekt `bignum` którego wartość jest równa liczbie reprezentowanej przez tablicę
- `int *toArray(bignum a)` - przyjmuje obiekt `bignum` i zwraca tablicę zmiennych `int` o wartościach z przedziału 0-9 reprezentującą kolejne cyfry liczby reprezentowanej przez argument

- void print(bignum a) - przyjmuje obiekt bignum i wypisuje jego wartość na standardowe wyjście

### 3. Moduł struktury

- void push(token val, pnode\* node) - przyjmuje token i wskaźnik do wskaźnika na koniec listy, po czym dodaje token na koniec listy i aktualizuje koniec
- token get\_top(pnode \*node) - przyjmuje wskaźnik do wskaźnika na koniec listy, po czym usuwa ostatni węzeł, aktualizuje koniec i zwraca wartość ostatniego węzła
- token get\_front(pnode \*node) - przyjmuje wskaźnik do wskaźnika na początek listy, po czym usuwa pierwszy węzeł, aktualizuje początek i zwraca wartość pierwszego węzła
- token top(pnode node) - przyjmuje wskaźnik na koniec listy i zwraca wartość ostatniego węzła (bez usuwania go)
- pnode find\_front(pnode node) - przyjmuje wskaźnik do dowolnego węzła listy i zwraca wskaźnik na jej początek

### 4. Moduł parser

- pnode parseInput(char \*s) - przyjmuje tablicę charów reprezentującą wyrażenie, zamienia je na ONP i zwraca wskaźnik na początek kolejki reprezentującej wyrażenie w ONP
- bignum evalRPN(pnode Qfront) - przyjmuje wskaźnik na początek kolejki reprezentującej wyrażenie w ONP, oblicza wartość tego wyrażenia i zwraca ją jako obiekt bignum