

Raport pracy projektowej nr 1

Zadanie 1.1

1. Dokumentacja

Funkcja `calka(f, a, b, n=100)` wyznacza całkę funkcji f na przedziale $[a, b]$ przy użyciu metody trapezów.

$$\int_a^b f(x)dx \approx h \cdot \left(\frac{y_0 + y_1}{2} + \frac{y_1 + y_2}{2} + \dots + \frac{y_{n-1} + y_n}{2} \right)$$

2. Argumentacja

f funkcja

n wartość liczbową większą od 0

a i b takie, że $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$ takie, że $x_i = a + ih$ gdzie $h = \frac{b-a}{n}$

3. Zwracana wartość

Funkcja zwraca wartość liczbową w postaci `double`.

4. Przykłady

```
calka(dnorm, -3, 3, 1000)
```

```
## [1] 0.9973001
```

```
calka(function(x) -x^2+2, -1, 1, 100)
```

```
## [1] 3.3332
```

Zadanie 1.2

1. Dokumentacja

Funkcja `sklej(x, sep)` skleja konkretne elementy listy tzn. bierze pierwsze elementy z elementów listy i skleja tworząc pierwszy napis, potem bierze drugie elementy każdego z elementów listy i skleja itd. Poszczególne elementy sklejanego napisu są oddzielane przy użyciu znaku określonego przez parametr `sep` (domyślnie jest to pusty napis.)

2. Argumentacja

x to k -elementowa lista zawierająca wektory napisów, wszystkie o tej samej długości n

`sep` odziela poszczególne elementy sklejanego napisu

3. Zwracana wartość

Funkcja zwraca n elementowy wektor napisów, w którym i -ty napis powstaje przez sklejanie i -tych elementów ze wszystkich wektorów z x .

4. Przykłady

```
sklej(list(letters[1:4], LETTERS[1:4], sep = "*"))
```

```
## [1] "a*A" "b*B" "c*C" "d*D"
```

```
w <- list(c("Ala", "Kasia", "Zosia"),
          c("ma", "lubi", "zawsze"),
          c("kota", "truskawki", "wygrywa"),
          c(".", ";", "!"))
sklej(w, sep=" ")
```

```
## [1] "Ala ma kota ." "Kasia lubi truskawki ;" "Zosia zawsze wygrywa !"
```

```
sklej(list(c("a", "b", "c"),
           as.character(1:3),
           rep("%", 3)),
       sep = "~")
```

```
## [1] "a 1 %" "b 2 %" "c 3 %"
```

```
sklej(c("a", "b", "c"), sep = " ")
```

```
## Error in sklej(c("a", "b", "c"), sep = " "): is.list(x) is not TRUE
```

Zadanie 1.3

1. Dokumentacja

Funkcja `repr_macierz(x, eps)` daną macierz liczbową przekształca do specjalnej postaci macierzy dostosowanej do oszczędnego reprezentowania w pamięci komputera macierzy rzadkich czyli takich dla których większość komórek zawiera wartości zerowe.

2. Argumentacja

x macierz liczbowa eps najmniejsza liczba uznawana za niezerową co do wartości bezwzględnej. Domyślnie to $eps = 1e^{-16}$

3. Zwracana wartość

Macierz składająca się z trzech kolumn i liczby wierszy równej liczbie niezerowych elementów macierzy wejściowej. W pierwszej i drugiej kolumnie znajdują się indeksy pól w których te wartości były natomiast w trzeciej kolumnie jest wartość.

4. Przykłady

```
set.seed(1)
values <- ifelse(sample(c(T,F),35, replace=TRUE, prob= c(0.1,0.9)), rnorm(20),0)
x <- matrix(values, ncol=7)
x
```

```
##           [,1]      [,2] [,3]      [,4]      [,5] [,6] [,7]
## [1,] 0.0000000 0.0000000 0 0.000000 0.4356833 0 0
## [2,] 0.0000000 -0.05710677 0 0.000000 0.0000000 0 0
## [3,] 0.0000000 0.0000000 0 1.151912 0.0000000 0 0
## [4,] 0.3773956 0.0000000 0 0.000000 0.0000000 0 0
## [5,] 0.0000000 0.0000000 0 0.000000 0.0000000 0 0
```

```
repr_macierz(x)
```

```
##      row col      val
## [1,]  4   1 0.37739565
## [2,]  2   2 -0.05710677
## [3,]  3   4 1.15191175
## [4,]  1   5 0.43568330
```

```
y <- matrix(0, ncol = 5, nrow = 7)
y[c(1,2,5,6),1] <- rnorm(4)
y
```

```
##           [,1] [,2] [,3] [,4] [,5]
## [1,] 1.2383041 0 0 0 0
## [2,] -0.2793463 0 0 0 0
## [3,] 0.0000000 0 0 0 0
## [4,] 0.0000000 0 0 0 0
## [5,] 1.7579031 0 0 0 0
## [6,] 0.5607461 0 0 0 0
## [7,] 0.0000000 0 0 0 0
```

```
repr_macierz(y)
```

```
##      row col      val
## [1,]  1   1 1.2383041
## [2,]  2   1 -0.2793463
## [3,]  5   1 1.7579031
## [4,]  6   1 0.5607461
```

```
y <- matrix(0, ncol = 5, nrow = 7)
y[1, c(1,3,4)] <- rnorm(3)
y[6,2] <- runif(1)
y
```

```
##           [,1]      [,2]      [,3]      [,4] [,5]
## [1,] -0.452784 0.0000000 -0.8320433 -1.166571 0
## [2,] 0.000000 0.0000000 0.0000000 0.000000 0
```

```
## [3,] 0.000000 0.0000000 0.0000000 0.000000 0.000000 0
## [4,] 0.000000 0.0000000 0.0000000 0.000000 0.000000 0
## [5,] 0.000000 0.0000000 0.0000000 0.000000 0.000000 0
## [6,] 0.000000 0.1433044 0.0000000 0.000000 0.000000 0
## [7,] 0.000000 0.0000000 0.0000000 0.000000 0.000000 0
```

```
repr_macierz(y)
```

```
##      row col      val
## [1,]   1   1 -0.4527840
## [2,]   6   2  0.1433044
## [3,]   1   3 -0.8320433
## [4,]   1   4 -1.1665705
```

Zadanie 1.4

1. Dokumentacja

Funkcja `logiderle(i, j, n)`, generuje n -elementowy wektor logiczny długości n wtedy i tylko wtedy gdy $(\exists p) l \in [i_p, j_p]$

2. Argumentacja

i wektor całkowitoliczbowy
 j wektor całkowitoliczbowy
 n długość zwracanego wektora

3. Zwracana wartość

Wektor logiczny długości n

4. Przykłady

```
logiderle(i=c(1,4), j=c(1, 6), n=7)
```

```
## [1] TRUE FALSE FALSE TRUE TRUE TRUE FALSE
```

```
logiderle(i=c(1,4,7), j = c(1,6,8), n=10)
```

```
## [1] TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE FALSE FALSE
```