Algorytmy i struktury danych

Algorytmy elementarne II

Prof. dr hab. Stanisław Gawiejnowicz

Wydział Biologii
Uniwersytet im. Adama Mickiewicza w Poznaniu

e-mail: stgawiej@amu.edu.pl

Wykład nr 3: Algorytmy elementarne II

- Algorytmy wykorzystujące tablice 2D
- Algorytmy obliczania wartości przybliżonych

Algorytmy wykorzystujące tablice 2D

- Rozpoczniemy od algorytmu dokonujacego inicjalizacji wszystkich elementów tablicy 2D

SUMY_INDEKSOW_TABLICY(m,n,A[1..m,1..n])

```
for i = 1 to m do
    for j =1 to n do
        A[i,j] = i + j
return
```

Algorytmy wykorzystujące tablice 2D

- Macierz to matematyczny obiekt, który można utożsamić z prostokątną tablicą liczb
- Każda macierz składa się z pewnej liczby wierszy oraz kolumn
- Jeżeli liczba wierszy jest równa liczbie kolumn, to macierz jest nazywana macierzą kwadratową
- Na macierzach można dokonywać różnego rodzaju operacji (dodawanie, mnożenie, mnożenie przez skalar itp.)
- Podamy teraz kilka przykładów algorytmów wykonywania ww. operacji na macierzach kwadratowych
- Algorytmy te wykorzystują strukturę tablicy 2D, którą można utożsamić z macierzą kwadratową

Algorytmy wykorzystujące tablice 2D

SUMA_ELEMENTOW_MACIERZY(n,A[1..n,1..n])

```
s = 0
for i = 1 to n do
    for j = 1 to n do
        s = s + A[i,j]
return s
```

Algorytmy wykorzystujące tablice 2D

SUMA_MACIERZY(n,A[1..n,1..n],B[1..n,1..n])

```
for i = 1 to n do
    for j = 1 to n do
        C[i,j] = A[i,j] + B[i,j]
return C
```

ILOCZYN_MACIERZY(n,A[1..n,1..n],B[1..n,1..n])

```
for i = 1 to n do
    for j = 1 to n do
        s = 0
        for k = 1 to n do
            s = s + A[i,k] * B[k,j]
            C[i,j] = s
return C
```

Algorytmy obliczania wartości przybliżonych

- W wielu przypadkach wartości obliczanej przez algorytm nie można obliczyć dokładnie
- W takich przypadkach algorytmy zwracają wartość przybliżoną wyniku, podaną z pewną dokładnością
- **Przykład:** liczba $\sqrt{2}$ jest liczbą niewymierną, co oznacza, że ma nieskończone rozwinięcie dziesiętne
- Powyższe oznacza, że wartość $\sqrt{2}$ można podać jedynie z pewną dokładnością
- **Przykład:** $\sqrt{2} \approx 1.41$ z dokładnością $\epsilon = 10^{-2}$

- Podamy teraz pseudokod algorytmu Herona obliczania $\sqrt{c}$ z liczby $c > 0$ z zadaną dokładnością

ALGORYTM_HERONA(c,c0,ϵ)

x1 = c0
x2 = 0.5*(c0+c/c0)
while (abs(x2-x1) >= ϵ) **do**
    tmp = 0.5*(x2+c/x2)
    x1 = x2
    x2 = tmp
return x2

Przykład zastosowania algorytmu Herona

- Niech $c = 2$, $c_0 = 1$ oraz $\epsilon = 10^{-2}$
- Wówczas w zerowej iteracji (przed wykonaniem pętli **while**) mamy $x_1 = 1$, $x_2 = \frac{1}{2}(1+2) = \frac{3}{2} = 1.5$
- $abs(x_2 - x_1) = 0.5 > \epsilon$, warunek w pętli **while** jest spełniony
- W pierwszej iteracji pętli **while** mamy: $tmp = \frac{1}{2}(\frac{3}{2} + \frac{2}{\frac{3}{2}}) = \frac{17}{12} = 1.41(6)$, $x_1 = \frac{3}{2}$, $x_2 = \frac{17}{12}$
- $abs(x_2 - x_1) = 0.01(6) > \epsilon$, warunek w pętli **while** jest spełniony
- W drugiej iteracji pętli **while** mamy: $tmp = \frac{1}{2}(\frac{17}{12} + \frac{2}{\frac{17}{12}}) = \frac{577}{408} = 1.4141\ldots$, $x_1 = \frac{17}{12}$, $x_2 = \frac{577}{408}$
- $abs(x_2 - x_1) = 0.0024\ldots < \epsilon$, warunek w pętli **while** nie jest spełniony
- Algorytm zatrzymuje się i zwraca $x_2 = \frac{577}{408}$

- Typowym przykładem wartości, które oblicza się z zadaną dokładnością są nieskończone sumy (pod warunkiem, iż istnieją)
- Nieskończone sumy oblicza się podobnie jak sumy skończone, biorąc jednak pod uwagę czy różnica kolejno obliczonych przybliżeń tych sum jest niewiększa niż zadana dokładność
- **Przykłady:** $S1 = 1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^n} + \ldots$
  $S2 = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} + \ldots$
- Suma S1 (szeregu geometrycznego) istnieje, natomiast suma S2 (szeregu harmonicznego) – nie istnieje

- Innym przykładem algorytmu obliczania przybliżonej wartości jest algorytm obliczania wartości liczby e z zadaną dokładnością
- Liczba e jest podstawą logarytmów naturalnych
- Wartość liczby e wynika z następującej definicji:

$$e = \lim_{n \to +\infty} \left(1 + \frac{1}{n}\right)^n$$

- Ww. algorytm korzysta z następującej równości:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \ldots$$

- By więc obliczyć przybliżoną wartość liczby e należy obliczyć wartość ww. nieskończonej sumy z zadaną dokładnością

Algorytmy obliczania wartości przybliżonych

- Co więcej, ponieważ zachodzi równość

$$\frac{x^{n+1}}{(n+1)!} = \frac{x^n}{n!} * \frac{x}{n+1},$$

  nie jest konieczne obliczanie kolejnych wartości $n!$
- Pseudokod algorytmu obliczania e^x jest następujący

EXP(x,ϵ)

```
i = 1
a = x
s = 1
while (a >= ε) do
    s = s + a
    i = i + 1
    a = a * (x/i)
return s
```