

Algorytmy i struktury danych - laboratorium

Lista nr 1

Janusz Szwabiński

Zad. 1 Niech $p(x)$ będzie wielomianem stopnia n , tzn.

$$p(x) = \sum_{i=0}^n a_i x^i$$

1. Zaproponuj prosty algorytm klasy $O(n^2)$ obliczenia wartości wielomianu w punkcie.
2. Zaproponuj algorytm klasy $O(n \log n)$ obliczenia wartości tego wielomianu w oparciu o bardziej wydajne obliczenie x^i .
3. Wielomian $p(x)$ można zapisać w postaci

$$p(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + x(a_{n-1} + xa_n) \dots))),$$

znanej jako **schemat Hornera**. Korzystając z notacji O , podaj liczbę operacji wymaganych do wyliczenia wielomianu tą metodą.

Zad. 2 Rozważmy następujący fragment kodu:

```
def example1(S):
    """Return the sum of the elements in sequence S."""
    n = len(S)
    total = 0
    for j in range(n):
        total += S[j]
    return total

def example2(S):
    """Return the sum of the elements with even index in sequence S.
    """
    n = len(S)
    total = 0
    for j in range(0, n, 2):
        total += S[j]
    return total

def example3(S):
    """Return the sum of the prex sums of sequence S."""
    n = len(S)
    total = 0
    for j in range(n):
        for k in range(1+j):
            total += S[k]
    return total

def example4(A, B): # assume that A and B have equal length
    """Return the number of elements in B equal to the sum of prex
    sums in A."""
    n = len(A)
```

```
count = 0
for i in range(n):
    total = 0
    for j in range(n):
        for k in range(1+j):
            total += A[k]
    if B[i] == total:
        count += 1
return count
```

Przeprowadź analizę eksperymentalną złożoności powyższych funkcji.

Zad. 3 Sprawdź eksperymentalnie, czy funkcja `sorted` w Pythonie rzeczywiście ma średnią złożoność $O(n \log n)$.