

Algorytmy i struktury danych

Wyszukiwanie wzorca w tekście

15 grudnia 2020

1 Implementacja algorytmów wyszukiwania wzorca

Napisz trzy funkcje, które implementują następujące algorytmy wyszukiwania wzorca w tekście:

- algorytm N (tzw. naiwny),
- algorytm KMP (Knutha-Morrisa-Pratta),
- algorytm KR (Karpa-Rabina)

w postaci funkcji typu:

```
def find (string, text)
    """
    Parameters:
    string (str): szukany napis
    text (str): przeszukiwany tekst
    Returns:
    (list): lista pozycji w 'text' (w kolejności rosnącej), od których zaczyna się 'string'
    """
```

2 Sprawdzenie poprawności implementacji

Przetestuj wszystkie funkcje dla przypadków brzegowych:

- pusty jeden lub oba napisy wejściowe,
- napis string równy napisowi text,
- napis string dłuższy od napisu text,
- napis string nie występuje w text.

Przetestuj implementację algorytmu naiwnego (dobierz kilka zestawów danych testowych oraz sprawdź poprawność wyników). Następnie przetestuj implementację pozostałych algorytmów w ten sposób, że dla generowanych losowo tekstów i wzorców (alfabet ogranicz do dwóch liter), sprawdź czy wszystkie implementacje zwracają ten sam wynik.

3 Porównanie algorytmów wyszukiwania wzorca

Jako tekst przeszukiwany wykorzystaj plik pan-tadeusz.txt. Dla każdej z funkcji:

- zmierz czas wyszukiwania w całym pliku n pierwszych słów wczytanych z pliku (np. $n = 100, 200, \dots, 1000$).

Narysuj zbiorczy wykres (jeden wykres dla trzech funkcji) pokazujący zależność czasu wyszukiwania od liczby szukanych słów.

4 Wyniki

Rezultatem powinny być:

- kod źródłowy z zaimplementowanymi funkcjami,
- kod źródłowy przeprowadzający komplet testów (punkt 2) i wyświetlający wyniki testów na ekranie,
- zrzut ekranu z wyświetlonymi wynikami testów,
- kod źródłowy przeprowadzający komplet pomiarów wydajności (punkt 3) i generujący plik z wykresem,
- wygenerowany plik z wykresem.

5 Ocena

Zadanie oceniane jest w skali 0-6 pkt.