Jan Danowski

Politechnika Rzeszowska

Algorytmy i struktury danych

Sprawozdanie

Projekt 2

Opis problemu

Zaimplementuj sortowanie Quicksort oraz sortowanie Gnome sort

Opis podstaw teoretycznych

1. Quicksort - Algorytm wykorzystuje technikę "dziel i zwyciężaj". Według ustalonego schematu wybierany jest jeden element w sortowanej tablicy, który będziemy nazywać pivot. Pivot może być elementem środkowym, pierwszym, ostatnim, losowym lub wybranym według jakiegoś innego schematu dostosowanego do zbioru danych. Następnie ustawiamy elementy nie większe na lewo tej wartości, natomiast nie mniejsze na prawo. W ten sposób powstaną nam dwie części tablicy (niekoniecznie równe), gdzie w pierwszej części znajdują się elementy nie większe od drugiej. Następnie każdą z tych podtablic sortujemy osobno według tego samego schematu.

Zalety:

• działa w miejscu, czyli ”in situ” (używa tylko niewielkiego stosu pomocniczego)

• do posortowania n elementów wymaga średnio czasu proporcjonalnego do n ∗ logn

• ma wyjątkowo skromną pętlę wewnętrzą

Wady:

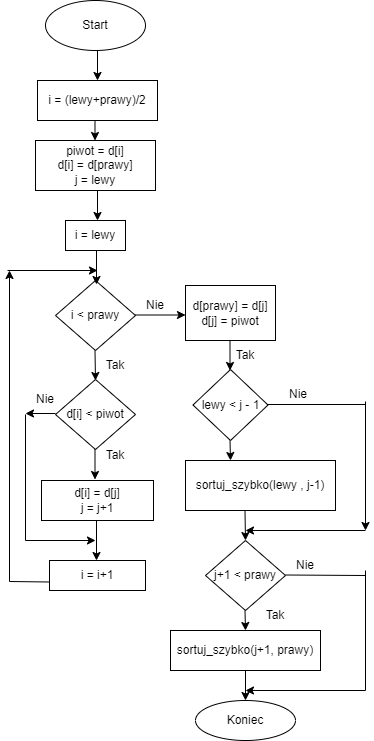
• jest niestabilny

• zabiera około n^2 operacji w najgorszym przypadku

• jest wrażliwy (prosty niezauważony błąd w implementacji może powodować niewłaściwe działanie w przypadku niektórych danych)

1. Gnome sort - Sortowanie Gnoma to metoda sortowania danych, która wymaga jedynie jednej pętli. Implementacja częściowo opiera się na pomyśle z sortowania bąbelkowego. Algorytm wymaga tylko zmiennej do zapamiętania aktualnej pozycji sortowania. Sortowanie rozpoczyna się od pierwszego elementu na liście. Jeśli aktualnie rozpatrywany element jest pierwszym elementem na liście, albo spełnia warunki posortowania to należy zwiększyć numer indeksu. W przeciwnym wypadku należy zamienić aktualny element z poprzednim i zmniejszyć indeks o 1.

Schemat blokowy quicksort



Pseudokod quicksort

K01: i ← (lewy + prawy)/2

K02: piwot ← d[i]; d[i] ← d[prawy]; j ← lewy

K03: Dla i = lewy, lewy + 1, ..., prawy - 1:

wykonuj kroki K04...K05

K04: Jeśli d[i] ≥ piwot,

to wykonaj kolejny obieg pętli K03

K05: d[i] ↔ d[j]; j ← j + 1

K06: d[prawy] ← d[j]; d[j] ← piwot

K07: Jeśli lewy < j - 1,

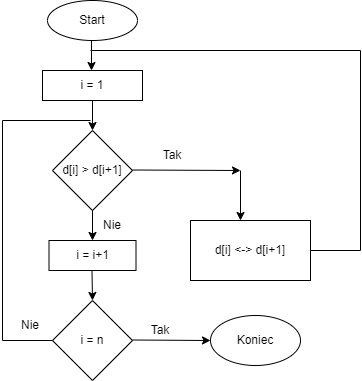
to Sortuj\_szybko(lewy, j - 1)

K08: Jeśli j + 1 < prawy,

to Sortuj\_szybko(j + 1, prawy)

K09: Koniec

Schemat blokowy gnome sort



Pseudokod gnome sort

**K01: Dla i = 1,2,...,n - 1:**

**Wykonuj kroki K02...K04**

**K02: Jeśli d[i] ≤ d[i + 1],**

**to wykonaj następny obieg K01**

**K03: d[i] ↔ d[i + 1]**

**K04: Idź do kroku K01**

**K05: Koniec**

Złożoność obliczeniowa i wykresy quicksort w przypadku optymistycznym i pesymistycznym

O(n log n) w przypadku optymistycznym, w przypadku pesymistycznym O(n2)

1. **Przypadek optymistyczny, również średni gdyż różnica między nimi jest nieznaczna**

Obraz zawierający tekst, ekran, zestaw, pomiar

Opis wygenerowany automatycznie

Rys. 1 Wynik programu dla 20 elementów



Rys. 2 Wynik programu dla 1000 elementów

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rys. 3 Wynik programu dla 10000 elementów



Rys. 4 Wynik dla 100 000 elementów

1. **Przypadek pesymistyczny**

Obraz zawierający tekst, wewnątrz, telewizja, ekran

Opis wygenerowany automatycznie

Rys. 5 Pesymistyczny dla 20 elementów



Rys. 6 Pesymistyczny 1000 elementów



Rys. 7 Pesymistyczny dla 10000 elementów



Rys. 8 Pesymistyczny dla 30000 elementów

Złożoność obliczeniowa i wykresy gnome sort w przypadku optymistycznym i pesymistycznym

Złożoność w przypadku optymistycznym wynosi O(n^2), w pesymistycznym O(n^3)

1. **Przypadek optymistyczny**

**Obraz zawierający tekst, monitor, wewnątrz, telewizja

Opis wygenerowany automatycznie**

Rys. 9 Dla 20 elementów

****

Rys. 10 Dla 1000 elementów

****

Rys. 11 Dla 10000 elementów

****

Rys. 12 Dla 100 000 elementów

* 1. **Wykres dla przypadku optymistycznego – tablicy z góry posortowana**

1. **Przypadek pesymistyczny**

**Obraz zawierający tekst, monitor, ekran, pomiar

Opis wygenerowany automatycznie**

Rys. 13 Dla 20 elementów

****

Rys. 14 Dla 1000 elementów

****

Rys. 15 Dla 3000 elementów

* 1. **Wykres dla przypadku pesymistycznego – przy ilości większej niż 3000 elementów program przestaje działać**

Kod źródłowy sortowania quicksort i gnome

**Quicksort void:**

Obraz zawierający tekst

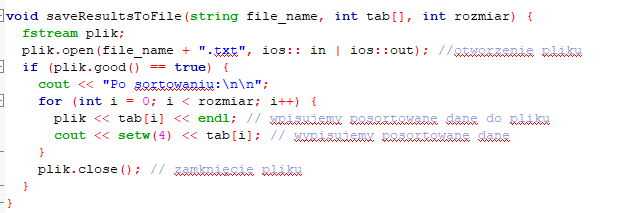
Opis wygenerowany automatycznie

**Gnome:**

Obraz zawierający tekst

Opis wygenerowany automatycznie

**Funkcja zapisu do pliku:**

****

Link do githuba

https://github.com/Dzbanowsky/Projekt2