**METODE SORTING DATA UNTUK PENGURUTAN DATA**

**PENDAHULUAN**

Sorting data adalah teknik yang dipakai untuk mengurutkan data. Terdapat 3 metode Sorting Data yang akan kami bahas yaitu Shell Sort, Quick Sort, Bubble Sort.

Kami akan membahas sejarah, cara kerja, kelebihan, kekurangan, dan kami juga akan melampirkan contoh metode Sorting Data yang kami buat sendiri untuk masing – masing metode Sorting Data yang kami sebutkan.

**Pembahasan**

**Shell Sort**

Metode Shell Sort disebut juga dengan metode pertambahan menurun(diminishing Increment). Metode ini dikembangkan oleh Donald L. Shell pada tahun 1959, sehingga sering disebut dengan metode Shell Sort.

Proses pengurutan dengan metode Shell dapat dijelaskan sebagai berikut:

1. Pertama-tama adalah menentukan jarak mula-mula dari data yang akan dibandingkan, yaitu N/2. Data pertama dibandingkan dengan data dengan jarak N/2. Apabila data pertama lebih besar dari data ke N/2 tersebut maka kedua data tersebut ditukar. Kemudian data kedua dibandingkan dengan jarak yang sama yaitu N/2. Demikian seterusnya sampai seluruh data dibandingkan sehingga semua data ke-j selalu lebih kecil daripada data ke-(j + N/2).
2. Pada proses berikutnya digunakan jarak (N/2)/2 atau N/4. Data pertama dibandingkan dengan data dengan jarak N/4. Apabila data pertama lebih besar dari data ke N/4 maka kedua data tersebut ditukar. Kemudian data kedua dibandingkan dengan jarak yang sama yaitu N/4. Demikianlah seterusnya hingga seluruh data dibandingkan sehingga semua data ke-j lebih kecil daripada data ke-(j + N/4).
3. Pada proses berikutkan, digunakan jarak(N/4) / 2 atau N/8. Demikian seterusnya sampai jarak yang digunakan adalah 1.

Proses Dalam algoritma:

1. Jarak = N
2. Selama (jarak > 1) kerjakan baris 3 sampai dengan 9
3. Jarak = jarak/2. Sudah = false
4. Kerjakan baris 4 sampai dengan 8 selama sudah = false
5. Sudah = true
6. J = 0
7. Selama (j < N – Jarak) kerjakan baris 8 dan 9
8. Jika (Data[j] > data[j + jarak] maka tukar data [j], data[j + jarak].
9. Sudah = True
10. J = j + 1

Dalam Python:

def shellSort(el):

    n = len(el)

    gap = n//2

    while gap > 0:

        for i in range(gap,n):

            temp = el[i]

            j = i

            while  j >= gap and el[j-gap] >temp:

                el[j] = el[j-gap]

                j -= gap

            el[j] = temp

        gap //= 2

        print("Sorted array is: ")

        for i in range(n):

            print(el[i])

Kelebihan Shell Sort:

1. Algoritma ini sangatrapat dan mudah untuk diimplementasikan.
2. Operasi pertukarannya hanya dilakukan sekali saja.
3. Waktu pengurutan dapat lebih ditekan.
4. Mudah menggabungkannya kembali.
5. Komplesitas selection sort relative lebih kecil

Kekurangan Shell Sort:

1. Membutuhkan method tambahan.
2. Sulit untuk membagi masalah.

**Quick Sort**

Quick Sort merupakan suatu algoritma pengurutan yang dikembangkan oleh Tony Hoare. Pengurutan data menggunakan teknik pemecahan data menjadi partisi – partisi, sehingga metode ini disebut juga dengan nama Partition Exchange Sort. Untuk memulai irterasi pengurutan, pertama – tama sebuah elemen dipilih dari data, kemudian elemen – elemen data akan diurutkan diatur sedemikian rupa.

Algoritma ini mengambil salah satu elemen secara acak (biasanya dari tengah) yang disebut dengan pivot lalu menyimpan semua elemen yang lebih kecil di sebelah kiri pivot dan semua elemen yang lebih besar di sebelah kanan pivot. Hal ini dilakukan secara rekursif terhadap elemen di sebelah kiri dan kanannya sampai semua elemen sudah terurut.

Algoritma Quick Sort:

1. Pilih satu elemen secara acak sebagai pivot.
2. Pindahakan semua elemen yang lebih kecil ke sebelah kiri pivot dan semua elemen yang lebih besar ke sebelah kanan pivot. Elemen yang nilainya sama bias disimpan di salah satunya.
3. Lakukan Sort secara rekursif terhadap sub-array sebelah kiri dan kanan pivot.

Tips Pemilihan Pivot:

Dalam algoritma Quick Sort, Pemilihan pivot adalah hal yang menentukan apakah algoritma Quick Sort tersebut akan memberikan performa terbaik atau terburuk. Berikut beberapa cara pemilihan pivot:

1. Pivot adalah elemen pertama, elemen terakhir, atau elemen tengah tabel. Cara ini hanya bagus jika elemen tabel tersusun secara acak, tetapi tidak bagus jika elemen tabel semula sudah terurut. Misalnya, jika elemen tabel semula menurun, maka semua elemen tabel akan terkumpul di upatabel kanan.
2. Pivot dipilih secara acak dari salah satu elemen tabel. Cara ini baik, tetapi mahal, sebab memerlukan biaya (cost) untuk pembangkitan prosedur acak. Lagi pula, itu tidak mengurangi kompleksitas waktu algoritma.
3. Pivot adalah elemen median tabel. Cara ini paling bagus, karena hasil partisi menghasilkan dua bagian tabel yang berukuran seimbang ( masing – masing = n/2 elemen). Cara ini memberikan kompleksitas waktu yang minimum. Masalahnya, mencari median dari elemen tabel yang belum terurut adalah persoalan tersendiri. Algoritma Quick Sort terdiri dari dua prosedur, yaitu prosedur PARTISI dan prosedur QUICKSORT.

Kelebihan Quick Sort:

1. Secara umum memiliki komplesitas.
2. Algoritmanya sederhana dan mudah diterapkan pada berbagai Bahasa pemrogaman dan arsitektur mesin secara efisien.
3. Dalam prakteknya adalah yang tercepat dari berbagai algoritma pengurutan dengan perbandingan, seperti mergesort dan sort lainnya.

Kekurangan Quick Sort:

1. Sedikit kesalahan dalam penulisan program membuatnya bekerja tidak beraturan (hasilnya tidak benar atau tidak pernah selesai).
2. Memiliki ketergantungan terhadap data yang dimasukkan, yang dalam terburuk memiliki kompleksitas.
3. Secara umum bersifat tidak stable, yaitu mengubah urutan input dalam hasil akhirnya (dalam hal inputnya bernilai sama).
4. Pada penerapan secara rekursif (memanggil dirinya sendiri) bila terjadi kasus terburuk dapat menghabiskan stack dan memacetkan program.
5. Pada Bahasa pemrograman, quicksort ada dalam pustaka stdlib.h untuk Bahasa C, class Tlist dan TstringList dalam Delphi (Object Pascal) maupun FreePascal.

Python Program:

1. def qsort(arr):
2. if len(arr) <= 1:
3. return arr
4. else:
5. return qsort([x for x in arr[1:] if x < arr[0]]) + \
6. [arr[0]] + \
7. qsort([x for x in arr[1:] if x >= arr[0]])

**Bubble Sort**

Bubble Sort merukan proses pengurutan yang secara berangsur-angsur berpindah ke posisi yang tepat karena itulah dinamakan Bubble yang artinya gelembung. Algoritma ini akan mengurutkan data dari yang terbesar ke yang terkecil (ascending) atau sebaliknya (descending).

Proses Bubble Sort:

1. Jumlah iterasi sama dengan banyaknya bilangan dikurang 1.
2. Di setiap iterasi, jumlah pertukaran bilangannya sama dengan jumlah banyaknya bilangan.
3. Dalam algoritma Bubble Sort, meskipun deretan bilangan tersebut sudah terurut, proses sorting akan tetap dilakukan.
4. Tidak ada perbedaan cara yang berarti untuk teknik algoritma Bubble Sort Ascending dan Descending.

Kelebihan Bubble Sort:

1. Algoritma ini adalah metode paling sederhana untuk mengurutkan data.
2. Algoritma ini mudah dipahami.

Kekurangan Bubble Sort:

1. Tingkan efisiensiny yang kurang. Bubble Sort ini merupakan metode pengurutan yang tidak efisien, khususnya ketika menangani data yang berjumlah besar. Hal tersebut karena mengurutkan data yang sangat besar akan sangat lambat prosesnya.
2. Jumlah pengulangan yang dilakukan oleh Algoritma ini akan tetap sama jumlahnya meskipun data yang diurutkan sudah cukup terurut.

Python Program:

def bubbleSort(el):

    n =len(el)

    for i in range(n):

        for j in range(0, n-i-1):

            if el[j] > el[j+1]:

                el[j], el[j+1] = el[j+1], el[j]

    print("Sorted array is:")

    for i in range(n):

        print("%d" %el[i])