

Lab06

CSGE601020

23 Oktober, 2020

Dasar-Dasar Pemrograman 1

Pengurutan Data

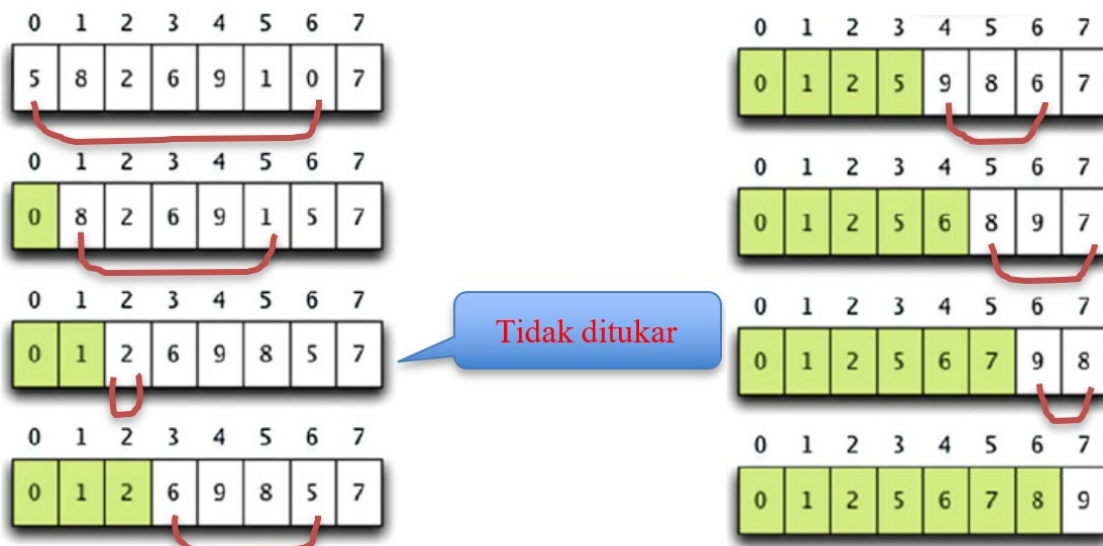
Baca dokumen ini dengan saksama sebelum membuat program

Gambaran Umum

Salah satu hal yang sering dilakukan saat memproses data adalah mengurutkan data (*sorting*). Misalnya daftar mahasiswa sering kali ingin ditampilkan dalam urutan leksikografis atau diurutkan berdasarkan NPM. Pada lab tutorial ini, akan dipelajari cara **membuat fungsi**, **memproses string** (`.split()`, `.strip()`, dan sebagainya), **menggunakan struktur data list**, dan **algoritme selection-sort**. Anda **tidak diizinkan** untuk menggunakan fungsi bawaan Python, seperti `sort()` atau `min()`.

Algoritme Selection-Sort

Dalam mengurutkan data, salah satu algoritme yang sering digunakan adalah *selection-sort*. Algoritme ini bekerja dengan mengatur ulang urutan dari data yang dipunya, sehingga datanya tersimpan dalam keadaan terurut (menaik atau menurun). Cara kerja dari algoritme ini adalah dengan berulang kali mencari elemen terkecil pada bagian tersisa yang belum terurut, dan memindahkannya ke depan dengan **menukarkannya**. Perhatikan contoh berikut.



Secara total, terdapat enam kali penukaran yang dilakukan untuk mengurutkan data tersebut. Pada awalnya, algoritme ini bekerja dengan mencari elemen terkecil dan meletakkannya pada posisi pertama dari *list* tersebut. Elemen yang terletak pada posisi pertama tentunya harus dipindahkan ke tempat lain. Dapat **ditukar** saja dengan posisi dari elemen terkecil yang ingin dipindahkan. Kemudian akan dicari nilai kedua terkecil pada *list* tersebut. Karena nilai terkecilnya terdapat pada posisi pertama *list*, maka akan dicari elemen terkecil keduanya dari posisi kedua. Setelah ditemukan (nilai kedua terkecil pada *list* tersebut), posisinya akan ditukar dengan posisi kedua. Kemudian akan dicari nilai ketiga terkecil, dan pola ini terus berulang hingga semua elemen pada barisan tersebut terurut.

Untuk mencari elemen terkecil dari list, dapat digunakan **sequential processing strategy** seperti pada *slide 2* halaman 46. Anda **tidak diizinkan** untuk menggunakan fungsi `min()`.

Contoh Interaksi dengan Pengguna

```
>>> ===== RESTART =====
Type a sequence of numbers (example: 3,100,-5,3):
65, 46, 14, 52, 38, 2, 96, 39, 14, 33, 13, 4, 24, 99, 89, 77, 73, 87, 36, 81
Input list:
[65, 46, 14, 52, 38, 2, 96, 39, 14, 33, 13, 4, 24, 99, 89, 77, 73, 87, 36, 81]
Sorted list:
[2, 4, 13, 14, 14, 24, 33, 36, 38, 39, 46, 52, 65, 73, 77, 81, 87, 89, 96, 99]
Number of swaps in Selection-Sort: 16
```

```
>>> ===== RESTART =====
Type a sequence of numbers (example: 3,100,-5,3):
5,3,3,1
Input list:
[5, 3, 3, 1]
Sorted list:
[1, 3, 3, 5]
Number of swaps in Selection-Sort: 1
```

```
>>> ===== RESTART =====
Type a sequence of numbers (example: 3,100,-5,3):
2,2,2,2,2
Input list:
[2, 2, 2, 2, 2]
Sorted list:
[2, 2, 2, 2, 2]
Number of swaps in Selection-Sort: 0
```

Template Program

```
# *****
# lab06.py
#
# Sorting a list of numbers using the Selection-Sort Algorithm
# *****

##
# The selectionSort function sorts a list using the selection sort algorithm.
#
# @param values: the list to sort
# @return the total number of swaps
#
# calls the function minimumPosition
#
def selectionSort(values):

    #
    # your code here
    #

##
# Finds the smallest element in a tail range of the list.
# @param values: the input list
# @param start: the first position in values to compare
# @return the position of the smallest element in the
#         range values[start] . . . values[len(values) - 1]
#
def minimumPosition(values, start) :

    #
    # your code here
    #

##
# Demonstrates the selection sort algorithm by sorting a
# list of numbers given by the user.

def main():
    input_string = input("Type a sequence of numbers (example: 3,100,-5,3):
\n")

    #
    # your code here
    #

    print('Number of swaps in Selection-Sort:', swap)

main()
```

Petunjuk Pengumpulan

- Simpan pekerjaan Anda (*source code* dengan **comments secukupnya**) dengan format penamaan **<KODEASDOS>_<NamaLengkap>_<NPM>_lab06.py** dan kumpulkan melalui *link* yang tersedia pada SCellE.

That's all for today. Happy programming! 'Met ngoding!

L. Y. Stefanus & The Asdos Team