**GREEDY-TSP METHOD**

import java.util.\*;

public class MainTspGreedy{

public static void main(String[] args) {

char abjad[] = {'A','B','C', 'D', 'E', 'F','G','H','I'};

int jumlahKota;

System.out.println("\nProgram TSP - Greedy");

Scanner mScanner = new Scanner(System.in);

System.out.print("jumlah Kota: ");

jumlahKota = mScanner.nextInt();

int[][] dataArr = new int[jumlahKota][jumlahKota];

// inputArray

System.out.println("input jarak antar kota\n");

for (int i = 0; i < jumlahKota; i++) {

for (int j = 0; j < jumlahKota; j++) {

if (i != j && dataArr[i][j] == 0) {

System.out.print("jarak" + abjad[i]+"-"+abjad[j]+" : ");

dataArr[i][j] = mScanner.nextInt();

dataArr[j][i] = dataArr[i][j];

}

}

System.out.println("");

}

// print arrayKota

for(int i=0; i<jumlahKota; i++){

if(i<1){

System.out.print("\t"+abjad[i]);

}

else{

System.out.print("\t"+abjad[i]);

}

}

System.out.println("\n");

for (int i = 0; i < jumlahKota; i++) {

for (int j = 0; j < jumlahKota; j++) {

if(j<1){

System.out.print(abjad[i]+"\t"+dataArr[i][j]);

}

else{

System.out.print("\t"+dataArr[i][j]);

}

}

System.out.println("");

}

// proses TSP greedy

char startKota;

System.out.print("\ntentukan nilai start: ");

startKota = mScanner.next().charAt(0);

boolean jalan =false;

int lokasiStart=0,locNow=0, hapus=0, tmpStart=0;

int jarakKota = 0, count=0;

int[] indexStart =new int [abjad.length];

for (int i = 0; i <jumlahKota; i++) {

if (startKota == abjad[i]) {

jalan = true;

tmpStart=i;

lokasiStart=i;

}

}

if (jalan == true) {

while (count < jumlahKota-1) {

int kecil = 9999;

hapus = lokasiStart;

for (int i = 0; i <jumlahKota ; i++) {

if (dataArr[lokasiStart][i] != 0 ) {

if (kecil > dataArr[lokasiStart][i]) {

kecil = dataArr[lokasiStart][i];

locNow =i;

}

}

}

jarakKota +=kecil;

indexStart[count] = locNow;

lokasiStart = locNow;

for(int j=0; j<jumlahKota; j++){

dataArr[hapus][j]=0;

dataArr[j][hapus]=0;

}

count++;

}

}

System.out.print("lokasi yg ditempuh: "+abjad[tmpStart]);

for (int i = 0; i < count; i++) {

System.out.print("->"+ abjad[ indexStart[i] ]);

}

System.out.println("\nJarak yg ditempuh: "+jarakKota+"\n\n");

}

}

**Greedy-SortestPath**

import java.util.Scanner;

public class SortestPath {

public static void main(String[] args) {

char vertex[] = {'A','B','C','D'};

Scanner tulis = new Scanner(System.in);

int n = vertex.length;

int nilai[][] = new int[n][n];

int start = 0, end = 0, ulang = 0, hapus, lokasi = 0, tempuh = 0, urut[], lstart = 0;

Boolean sstart = false, send = false;

char input1, input2;

inputNilai(nilai, n, vertex, tulis);

cetakArray(nilai, vertex, n);

System.out.print("Masukkan Tujuan Awal \t: ");

input1 = tulis.next().charAt(0);

for (int i = 0; i < n; i++) {

if (vertex[i] == input1) {

sstart = true;

start = i;

lstart = i;

System.out.println(start);

}

}

System.out.print("Masukkan Tujuan Akhir \t: ");

input2 = tulis.next().charAt(0);

for (int i = 0; i < n; i++) {

if (vertex[i] == input2) {

send = true;

end = i;

System.out.println(end);

}

}

urut = new int [30];

if (sstart == true && send == true) {

while (start != end) {

hapus = start;

int temp = 9999;

for (int i = 0; i < n; i++) {

if (nilai[start][i] != 0) {

if (temp > nilai[start][i]) {

temp = nilai[start][i];

lokasi = i;

}

}

}

tempuh += temp;

urut[ulang] = lokasi;

start = lokasi;

for (int i = 0; i < n; i++) {

nilai[hapus][i] = 0;

nilai[i][hapus] = 0;

}

ulang++;

}

System.out.print("\n");

System.out.print("Lokasi yang ditempuh : " + vertex[lstart]);

for (int i = 0; i <nilai.length-1; i++) {

System.out.print(" -> " + vertex[urut[i]]);

}

System.out.println("\nLokasi yang ditempuh adalah : " + tempuh + "\n");

}

else {

System.out.println("Lokasi Start dan End Tidak ketemu");

}

}

public static void cetakArray (int nilai[][], char vertex[],int n) {

System.out.println("---------------------------");

for (int i = 0; i < n; i++) {

System.out.print("\t" + vertex[i]);

}

System.out.print("\n");

for (int i = 0; i < n; i++) {

System.out.print(vertex[i] + "\t");

for (int j = 0; j < n; j++) {

System.out.print(nilai[i][j] + "\t");

}

System.out.print("\n");

}

System.out.println("---------------------------");

}

public static void inputNilai (int nilai[][], int n, char vertex[], Scanner tulis) {

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

System.out.print("Masukkan Nilai " + vertex[i] + " " + vertex[j] + " : ");

nilai[i][j] = tulis.nextInt();

}

}

}

}

**Greedy Jadwal Karyawan**

import java.util.\*;

public class MainJd{

public static void main(String[] args) {

int[][] dataArr= {

{5,7,2,1},

{3,4,4,2},

{8,7,1,3},

{5,1,2,6}

};

System.out.println("\nGreedy-JadwalKaryawan");

for (int i = 0; i <dataArr.length; i++) {

if(i<1){

System.out.print(" P"+(i+1));

}

else{

System.out.print(" P"+(i+1));

}

}

System.out.println("");

for (int i = 0; i < dataArr.length; i++) {

for(int j=0; j<dataArr.length; j++){

if (j <1) {

System.out.print("W"+(i+1)+" "+dataArr[i][j]);

}

else{

System.out.print(" "+dataArr[i][j]);

}

}

System.out.println("");

}

System.out.println("\n");

mJadwal(dataArr);

}

// method penjadwalan

private static void mJadwal(int[][] arr) {

int[] terbaik = new int[arr.length];

int kecil, index=0, tampung=0;

for (int i = 0; i <arr.length; i++) {

kecil=9999;

for (int j = 0; j <arr.length; j++) {

if (arr[j][i] != 9999) {

if(arr[j][i] < kecil){

kecil = arr[j][i];

index = j;

terbaik[i]=j;

}

}

}

tampung += kecil;

for (int k = 0; k <arr.length; k++) {

arr[index][k]= 9999;

}

}

for (int l = 0; l <arr.length; l++) {

System.out.println("P"+(l+1)+" = W"+(terbaik[l]+1));

}

System.out.println("\nTotal Waktu: "+tampung);

}

}

**BruteForce- TSP**

mainTsp.java

import java.util.Scanner;

public class MainTsp{

public static void main(String[] args) {

char abjad[] = {'A','B','C', 'D', 'E', 'F','G','H','I'};

char start;

int jumlahKota, kecil = 9999, terbaik = 0, tampung, mulai;

System.out.println("Program TSP - BruteForce");

Scanner mScanner = new Scanner(System.in);

System.out.print("jumlah Kota: ");

jumlahKota = mScanner.nextInt();

int[][] dataArr = new int[jumlahKota][jumlahKota];

FunctionTsp mTsp = new FunctionTsp();

int[][] p;

p = mTsp.permutasi(jumlahKota);

// inputArray

System.out.println("input jarak antar kota\n");

for (int i = 0; i < jumlahKota; i++) {

for (int j = 0; j < jumlahKota; j++) {

if (i != j && dataArr[i][j] == 0) {

System.out.print("jarak" + abjad[i]+"-"+abjad[j]+" : ");

dataArr[i][j] = mScanner.nextInt();

dataArr[j][i] = dataArr[i][j];

}

}

}

// print arrayKota

for(int i=0; i<jumlahKota; i++){

if(i<1){

System.out.print(" "+abjad[i]);

}

else{

System.out.print("\t"+abjad[i]);

}

}

System.out.println("");

for (int i = 0; i < jumlahKota; i++) {

for (int j = 0; j < jumlahKota; j++) {

if(j<1){

System.out.print(abjad[i]+" "+dataArr[i][j]);

}

else{

System.out.print("\t"+dataArr[i][j]);

}

}

System.out.println("");

}

System.out.print("tentukan start Kota");

start = mScanner.next().charAt(0);

int factorial = mTsp.factorial(jumlahKota);

for (int i = 0; i < factorial; i++) {

tampung = 0;

if(abjad[ p[i][0] ] == start){

mulai = p[i][0];

for(int j=0; j<jumlahKota; j++){

if(abjad[ p[i][j] ] != start){

tampung =tampung+ dataArr[mulai][ p[i][j] ];

mulai = p[i][j];

}

}

if(tampung < kecil){

kecil = tampung;

terbaik = i;

}

}

}

System.out.print("\nKombinasi Terbaik adalah: ");

for(int i= 0; i < jumlahKota; i++){

if(i == 0){

System.out.print(abjad[ p[terbaik][i]]);

}

else{

System.out.print("->"+abjad[ p[terbaik][i]]);

}

}

System.out.println("\nTotal Jarak : "+kecil+"\n\n");

mScanner.close();

}

}

**BruteForce-TSP**

FungtionTsp.java

public class FunctionTsp{

private static int x;

String[] data = new String[x];

private static int index = -1;

static int factorial(int n){

if(n == 0){

return 1;

}

else{

return n\*factorial(n-1);

}

}

private String[] execute( String[] str, int awal, int akhir){

if(awal == akhir){

index++;

data[index] = str[0];

}

else{

for(int i=awal; i <=akhir; i++){

str = swap(str, awal, i);

execute(str, awal+1, akhir);

str = swap(str, awal, i);

}

}

return data;

}

public static String[] swap(String[] a, int i, int j){

char tmp;

char[] charArray = a[0].toCharArray();

tmp = charArray[i];

charArray[i] = charArray[j];

charArray[j] = tmp;

a[0] = String.valueOf(charArray);

return a;

}

public static int[][] permutasi(int jumlah){

String[] arr={""};

x = factorial(jumlah);

int[][] data = new int[x][jumlah];

for(int i = 0; i < jumlah; i++){

arr[0]=arr[0] + String.valueOf(i);

}

FunctionTsp mFunctionTsp = new FunctionTsp();

arr = mFunctionTsp.execute(arr, 0, jumlah-1);

char[] charArray;

for(int i = 0; i < x; i++){

charArray = arr[i].toCharArray();

for(int j = 0; j < jumlah; j++){

data[i][j] = Integer.parseInt(String.valueOf(charArray[j]));

}

}

return data;

}

}

**Greedy-Knapsack**

import java.util.Scanner;

/\*\* Class Knapsack \*\*/

public class knapsack {

public void solve(int[] berat, int[] nilai, int W, int N) {

int NEGATIVE\_INFINITY = Integer.MIN\_VALUE;

int[][] m = new int[N + 1][W + 1];

int[][] sol = new int[N + 1][W + 1];

for (int i = 1; i <= N; i++) {

for (int j = 0; j <= W; j++) {

int m1 = m[i - 1][j];

int m2 = NEGATIVE\_INFINITY;

if (j >= berat[i]) {

m2 = m[i - 1][j - berat[i]] + nilai[i];

}

/\*\* pilihan max dari m1, m2 \*\*/

m[i][j] = Math.max(m1, m2);

sol[i][j] = m2 > m1 ? 1 : 0;

}

}

/\*\* Buat Daftar dari Semua Item Pilihan \*\*/

int[] selected = new int[N + 1];

for (int n = N, w = W; n > 0; n--) {

if (sol[n][w] != 0) {

selected[n] = 1;

w = w - berat[n];

} else {

selected[n] = 0;

}

}

/\*\* Cetak item pilihan \*\*/

System.out.println("\nPilihan Item : ");

for (int i = 1; i < N + 1; i++) {

if (selected[i] == 1) {

System.out.print(i +" ");

System.out.println();

}

}

}

/\*\* Main function \*\*/

public static void main (String[] args) {

Scanner scan = new Scanner(System.in);

System.out.println("Knapsack Algorithm Test\n");

/\*\* Buat sebuah objek class knapsack greedy \*\*/

knapsack ks = new knapsack();

System.out.println("Masukkan Jumlah Element ");

int n = scan.nextInt();

int[] berat = new int[n + 1];

int[] nilai = new int[n + 1];

System.out.println("\nMasukkan Berat Untuk "+ n +" Element");

for (int i = 1; i <= n; i++)

berat[i] = scan.nextInt();

System.out.println("\nMasukkan Nilai Untuk "+ n +" Element");

for (int i = 1; i <= n; i++)

nilai[i] = scan.nextInt();

System.out.println("\nMasukkan Berat Maks Knapsack ");

int W = scan.nextInt();

ks.solve(berat, nilai, W, n);

}

}

**Bubble Sort**

import java.util.\*;

public class BubbleSort{

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int batas;

System.out.print("Masukkan Batas : ");

batas = input .nextInt();

int[] data = new int[batas];

for(int i=0; i<batas; i++){

System.out.print("Data index " + (i+1) + " = ");

data[i] = input.nextInt();

}

System.out.println("\nData Sebelum di urutkan : ");

for(int i=0; i<batas; i++){

System.out.print(data[i] + ", ");

}

//Proses Bubble sort

for(int i=0; i<batas; i++){

System.out.println("\nIterasi ke - " + (i+1));

for(int j=0; j<batas-1; j++){

if(data[j] > data[j+1]){

int tmp = data[j];

data[j] = data[j+1];

data[j+1] = tmp;

}

if(j < batas-(i+1)){

for(int k=0; k<batas; k++){

System.out.print(data[k] + ", ");

}

System.out.println(" ");

}

}

System.out.println(" ");

}

System.out.println("Data Setelah di sorting : ");

for(int i=0; i<batas; i++){

System.out.print(data[i] + ", ");

}

}

}

**Selection Sort**

import java.util.\*;

public class Selection{

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int batas;

System.out.print("Masukkan Batas : ");

batas=input.nextInt();

int[] data = new int[batas];

for(int i=0; i<batas; i++){

System.out.print("Input data indeks " + (i+1) + " : ");

data[i] = input.nextInt();

}

System.out.println("\nData sebelum di sorting ");

for(int i=0; i<batas; i++){

System.out.print(data[i] + ", ");

}

System.out.println("\n Proses Selection Sort");

for(int i=0; i<batas; i++){

System.out.println("\nIterasi ke " + (i+1));

boolean tukar = false;

int index = 0;

int min = data[i];

for(int j=i+1; j<batas; j++){

if(min > data[j]){

tukar = true;

index = j;

min = data[j];

}

}

if(tukar == true){

int tmp = data[i];

data[i] = data[index];

data[index]=tmp;

}

for(int k=0; k<batas; k++){

System.out.print(data[k] + ", ");

}

System.out.println();

}

System.out.println("Data setelah di sort");

for(int i=0; i<batas; i++){

System.out.print(data[i] + ", ");

}

}

}