DA-ALG1000

Obligatorisk oppgave 2: Prosjektrapport

Av: Martin Kowalik Gran

Forord

Takk.

Innholdsfortegnelse

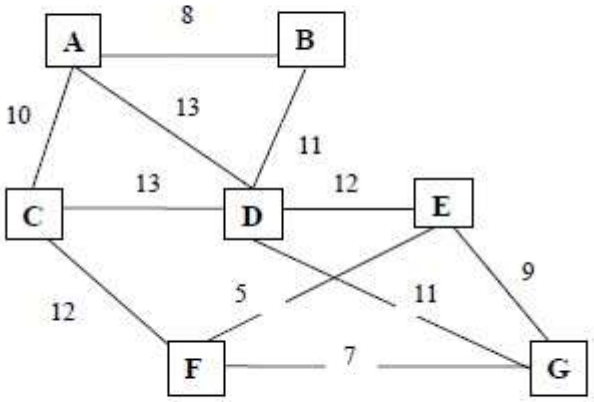
1.0 Innledning

Denne rapporten omhandler den andre obligatoriske programmeringsoppgaven i faget DA-ALG1000 våren 2015. Oppgaven var å skrive et program som finner billigste vei og skrive ut denne veien. Programmet og rapporten er skrevet av Martin Kowalik Gran uten hjelp av andre. Rapporten inneholder bestillingen (oppgavetekst), beskrivelse av fremgangsmåten av utviklingen av programmet, brukerveiledning for det ferdige programmet, UML og testing av programmet.

1.1 Oppgaveteksten

I følgende figur tenker vi oss at nodene A, B, C, D, E, F og G indikerer hus. Kantene mellom husene indikerer at det kan trekkes en kabel for telefon mellom 2 hus til en kostnad av det tallet som er oppgitt ved den tilhørende kanten. Prisen er oppgitt i antall 1000 kr. Programmer en løsning slik at alle hus er tilknyttet og at total kostnad er minst mulig. Man skal kunne finne samme løsning uansett hvilken node/hus man starter i. Programmet skal skrive ut:
total kostnad hvilke hus som parvis velges underveis med tilhørende kostnad

Ønsker ikke å se en løsning som kan spores til å ha blitt funnet på Internett. Oppgaven leveres som én fil (gjerne .zip) og rapport skal være med (i samme fila!). Oppgaven er tenkt løst vha. en tabell/matrise-representasjon av grafen, men forsøk dere gjerne på en annen metode. Lykke til!



2.0 Bakgrunnskunnskap for oppgaven

2.1 Prims algoritme

Prims algoritme er en grådig algoritme som finner minste «avstand/spenntre» i en vektet graf. Baserer seg på å hele tiden velge det «billigste» alternativet tilgjengelig. Oppdaget av Vojtech Jarnik i 1930 og senere gjenoppdaget av Robert Clay Prim i 1957 og Edsger Dijkstra i 1959.

3.0 Beskrivelse av programmet

3.1 Beskrivelse av fremgangsmåten

Skrev først metode for å skrive ut «tabell» av hus og veier. Metoden skriver ut hus i et multidimensjonalt array der indeksene tilsvarer «hus» og verdiene på i cellene tilsvarer kantene mellom husene.

Skrev deretter søkealgoritmen som egen metod, searchLeastValue(). Denne kjørte gjennom tre forløkker der den ytterste går antallet hus -1. Den neste løkken går det samme antallet som hus koblet til «nettverket», den siste løkken går antallet kanter hvert hus kan ha, i dette tilfellet 7.

Søket går ut på å hele tiden finne minste verdi. Det ble derfor valgt en startverdi, 666, større enn største verdi for kant i systemet. Hver kant i det sammensluttede nettet ble sammenlignet med nåværende minste verdi. Det ble også satt inn forutsetning om at verdien ikke var 0 eller at «huset» var besøkt før. Kontroll om hus var besøkt før ble gjennomført gjennom en egen metode som sjekket kanten mot besøkte hus. Metoden ble kalt isArray().

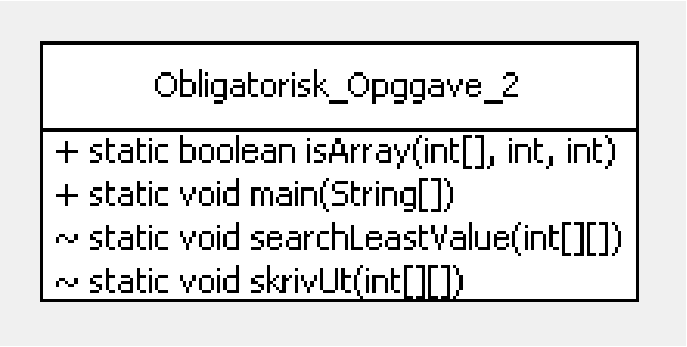
I tilfeller hvor kravene var oppfylt ble minste tilknyttede kant lagret i egen array. Antall besøk ble da økt med en og løkkene økte antallet søk.

Tilslutt ble det kodet en utskift av huspar etter fra hvor til hvor de var koblet. Denne kommer i rekkefølge. Tallindeksene ble oversatt til bokstaver gjennom to switcher. Tilslutt ble den totale verdien for kantene skrevet ut.

Programmet er skrevet slik at startposisjonen kan bestemmes av bruker. En inputvariabel av typen int ble valgt for letthets skyld.

4.0 Brukerveiledning

4.1 UML



Figur 2: Figuren viser UML for programmet med variabler og metoder.

5.0 Feiltesting

Det ble gjennomført en systematisk feiltesting av programmet. Den minste veien ble først regnet ut manuelt. Deretter ble det sjekket at programmet klarte å finne denne veien og verdien uansett startsted. Tilslutt ble rekkefølgen kontrollert slik at utskriften illustrerte veier på et korrekt vis.

	Startposisjon	Verdi	Rekkefølge
Hus	A	52	ok
Hus	B	52	ok
Hus	C	52	ok
Hus	D	52	ok
Hus	E	52	ok
Hus	F	52	ok
Hus	G	52	ok

Tabell1: *Tabellen viser de ulike testkriteriene som funksjonene har bestått.*

6.0 Konklusjon

Oppgaven ble løst innen fastsatt tid. Programmet utfører den algoritmen som er bestilt på grafen. Feiltesting er gjennomført og svaret er riktig uansett startpunkt. Et problem med programmet er at det er skrevet kun for å løse denne grafen eller andre grafer med 7 «hus».