

Laboration 1, Grafer

Laborationen behandlar elementära grafalgoritmer.

Syfte

Implementera elementära grafalgoritmer. Sökning med DFS och BFS. Dijkstras algoritm.

Tidsåtgång

Ett tillfälle är avsatt för detta moment.

Uppgifter

Läs in indata, en adjacency matris.

Implementera Depth-First Search och Breath-First Search för att avgöra om grafen är sammanhängande **

Om du finner bekymmer i grafen ange hur du rättar till dessa.

Implementera Dijkstras algoritm.

- Hur lång är vägen mellan Nackstavägen till Förrådet? Beskriv promenaden.
- Hur lång är vägen mellan L319 och D025?
- Hur lång är vägen från Universitetet till Bite Line Västra?

Se även tips.

Verktyget som använts för att skapa export.txt kan du hämta.

** -(Clarification: DFS and BFS have to be used to check that the graph is valid. A disconnected graph would be an example of an invalid graph. So, your need to use BFS and DFS to make sure that EVERY node in the graph can be reached from ANY node in the graph. /Elijs)

Tips: labb grafer

Detta är den minsta av de tre labbarna i labbserien. I labben ska du lösa vissa uppgifter med hjälp av indata du erhållit och algoritmerna som presenterats under föreläsning.

Programmeringsmomenten i denna labb är

- Förstå datat i indatafilen, testa din implementations förståelse av indata genom att skriva ut de olika delarna separat. Kontrollräkna så att du får med alla noderna.
- Implementera den datastruktur som ska representera grafen. Både lista och matris är möjliga att använda. En matris kan i vissa delar möjliggöra en mer direkt översättning algoritm -> implementation.
- Implementera algoritmerna. Börja med DFS, testa implementationen. Din funktion ska kunna svara på frågan om grafen är sammanhängande utifrån en given startnod.
- Implementera funktioner som besvarar frågeställningarna.

Parsning av indata

Parsning av indata är en nödvändig del av labben. Eftersom den här delen av uppgiften inte berör varken en adjacency-matris eller någon av algoritmerna ger jag här ett exempel på hur det kan gå till. Kontrollera med din handledare om du får använda koden som den är, eller om du ska skriva egen kod.

Att förstå indata består av två steg. Lexning och parsning.

Lexning

Lexning består i att tillskriva speciella symboler eller uttryck i indata en betydelse. Symbolerna själva kallas lexemer, dess betydelse beskrivs av en token. I vårt indata måste vi tilldela en betydelse till åtminstone 3 olika typer av symboler.

Lexeme: #	Token: Comment
Lexeme: M	Token: Meta
Lexeme: 0-9	Token: Edge

En lexer måste också kunna berätta om indata är slut så ytterligare en token behövs:

Lexeme: [EOF] Token: EndOfFile

Parsning

Parsningsdelen består i att

1. bestämma en datastruktur som kan beskriva indata på ett lättanvänt sätt.
2. fråga lexern om hur indata ser ut.
3. Läs/förstå indata som din token indikerar, därefter sätta in indata i datastrukturen i (1). Detta är parse-steget.

Eftersom indata är radbaserat är det enkelt att skriva en lexer och parser för detta dataformat.

Det exempel jag ger nedan använder följande gränssnitt för inläsning av indatafilen.

reader.h

```
//  
// Created by martin on 2022-03-30.  
//  
  
#ifndef DOA_LABB1_READER_H  
#define DOA_LABB1_READER_H  
  
#include<istream>  
#include<map>  
#include<vector>  
  
using node_id_t = int;  
using weight_t = double;  
using meta_t = std::map<node_id_t, std::string>;  
  
struct edge{  
    node_id_t n1;  
    node_id_t n2;  
    weight_t weight;  
    std::string description;  
};  
  
using edge_list_t = std::vector<edge>;  
  
using adjacency_list_t = std::pair<meta_t, edge_list_t>;  
  
adjacency_list_t parse_file(std::string filename);  
  
#endif //DOA_LABB1_READER_H
```

Funktionen `parse_file` returnerar en adjacency list tillsammans med beskrivningen av varje nod. Du bör därefter göra om listan till en adjacency matris.

Implementationen för `reader.h` kan vara

`reader.cpp`

```
//
// Created by martin on 2022-03-30.
//

#include "reader.h"

#include<fstream>
#include<string>
#include<map>

enum token{
    COMMENT, META, EDGE, END_OF_FILE
};

token get_line_type(std::istream& is){
    switch(is.peek()){
        case std::istream::traits_type::eof(): return END_OF_FILE;
        case '#': return COMMENT;
        case 'M': return META;
    };
    return EDGE;
}

meta_t meta;

edge read_edge(std::istream& is){
    edge e;
    is >> e.n1 >> e.n2 >> e.weight;
    std::getline(is, e.description);
    return e;
}

void read_meta(std::istream& is){
    char discard;
    node_id_t vertex_id;
    std::string name;
    is >> discard >> vertex_id;
    std::getline(is, name);
    meta[vertex_id] = name;
}

adjacency_list_t parse_file(std::string filename){
    std::ifstream in(filename);
    token l;
    edge_list_t edge_list;
    while((l = get_line_type(in)) != END_OF_FILE){
        edge e;
        switch(l){
            case token::EDGE:
                e = read_edge(in);
                edge_list.push_back(e);
                break;
        }
    }
}
```

```

        case token::META:
            read_meta(in);
            break;
        default:
            std::string comment;
            std::getline(in, comment);
    }
}
return adjacency_list_t{meta, edge_list};
}

```

```

# Name: Sundsvall västra
# Directional edges: 115
# M indicates Meta data
# Vertex ID:Name
M 0 Skyddsrums
M 1 Bultgatan 21
M 2 Bultgatan 19
M 3 Genväg 1
M 4 Genväg 2
M 5 Järnvägsövergång
M 6 Genväg Åkanten
M 7 Åkanten1
M 8 Åkanten infart parkering
M 9 Åkanten parkering
M 10 Sidsjövägen campus 1
M 11 Sidsjövägen campus 2
M 12 Sidsjövägen Kvarngatan
M 13 Bergsgatan Sidsjövägen 1
M 14 Bergsgatan Sidsjövägen 2
M 15 Bergsgatan Kvarngatan

```

M 16 Sidsjövägen Lekängsvägen
 M 17 Bergsgatan Hårdvallsgatan
 M 18 Bultgatan 4
 M 19 Bite Line West
 M 20 Storgatan Rondell Universitetet
 M 21 Bergsgatan Vinkeltået
 M 22 Bergsgatan Coop
 M 23 Bergsgatan Nackstavägen
 M 24 Nackstavägen Axvägen 1
 M 25 Infart Ica Kvantum
 M 26 Ica Kvantum
 M 27 Storgatan Väderkvarnsbacken
 M 28 Grönborgsgatan Väderkvarnsbacken
 M 29 Grönborgsgatan Dalgatan
 M 30 Rondell Campus
 M 31 Storgatan Dalgatan
 M 32 Dalgatan Norra Järnvägsgatan
 M 33 Norra Järnvägsgatan Västra Allén
 M 34 Grönborgsgatan Västra Allén
 M 35 Storgatan Västra Allén
 M 36 Storgatan Åkersviksgatan
 M 37 Förrådet
 M 38 Dalgatan Södra Järnvägsgatan
 M 39 Dalgatan Slutgatan
 M 40 Storgatan Ågatan
 M 41 Ågatan uppfart 2
 M 42 Ågatan uppfart 1
 M 43 Sporthallen parkering
 M 44 Entré Hus L
 M 45 L213
 M 46 L319
 M 47 D025
 # VertexID VertexID Length [Description]
 0 1 209.15 [Bultgatan]
 1 0 209.15 [Bultgatan]
 1 2 72.50 [Bultgatan]
 2 1 72.50 [Bultgatan]
 2 3 133.86 [Genväg Åkanten]
 3 2 133.86 [Genväg Åkanten]
 3 4 72.50 [Genväg Åkanten]
 4 3 72.50 [Genväg Åkanten]
 4 5 103.18 [Genväg Åkanten]
 5 4 103.18 [Genväg Åkanten]
 5 6 89.24 [Genväg Åkanten]
 6 5 89.24 [Genväg Åkanten]
 6 7 153.38 [Åkanten]
 7 6 153.38 [Åkanten]
 7 8 390.41 [Åkanten]
 8 7 390.41 [Åkanten]
 8 9 175.68 [Åkanten parkering]
 9 8 175.68 [Åkanten parkering]
 8 10 237.04 [Åkanten]
 10 8 237.04 [Åkanten]
 10 11 150.59 [Sidsjövägen]
 11 10 150.59 [Sidsjövägen]
 11 12 172.90 [Sidsjövägen]
 12 11 172.90 [Sidsjövägen]
 12 13 253.77 [Sidsjövägen]
 13 12 253.77 [Sidsjövägen]
 13 14 27.89 [Bergsgatan]
 14 13 27.89 [Bergsgatan]

13 15 153.38 [Bergsgatan]
15 13 153.38 [Bergsgatan]
15 16 175.68 [Bergsgatan]
16 15 175.68 [Bergsgatan]
16 17 270.50 [Bergsgatan]
17 16 270.50 [Bergsgatan]
16 18 119.91 [Bultgatan]
18 16 119.91 [Bultgatan]
18 2 658.12 [Bultgatan]
2 18 658.12 [Bultgatan]
19 3 92.03 [Stängd pizzeria]
10 20 69.72 [Sidsjövägen]
20 10 69.72 [Sidsjövägen]
17 21 276.08 [Bergsgatan]
21 17 276.08 [Bergsgatan]
21 22 89.24 [Bergsgatan]
22 21 89.24 [Bergsgatan]
22 23 181.26 [Bergsgatan]
23 22 181.26 [Bergsgatan]
23 1 44.62 [Nackstavägen]
1 23 44.62 [Nackstavägen]
23 24 105.97 [Nackstavägen]
24 23 105.97 [Nackstavägen]
24 25 75.29 [Nackstavägen]
25 24 75.29 [Nackstavägen]
25 26 103.18 [Parkering Ica Kvantum]
26 25 103.18 [Parkering Ica Kvantum]
27 28 94.81 [Väderkvarnsbacken]
28 27 94.81 [Väderkvarnsbacken]
28 29 161.74 [Grönborgsgatan]
29 28 161.74 [Grönborgsgatan]
20 30 89.24 [Holmgatan]
30 20 89.24 [Holmgatan]
20 27 184.05 [Storgatan]
27 20 184.05 [Storgatan]
31 29 86.45 [Dalgatan]
29 31 86.45 [Dalgatan]
31 27 139.43 [Storgatan]
27 31 139.43 [Storgatan]
29 32 209.15 [Dalgatan]
32 29 209.15 [Dalgatan]
32 11 326.27 [Norra Järnvägsgatan]
11 32 326.27 [Norra Järnvägsgatan]
33 34 211.94 [Västra Allén]
34 33 211.94 [Västra Allén]
29 34 131.07 [Grönborgsgatan]
34 29 131.07 [Grönborgsgatan]
34 35 128.28 [Västra Allén]
35 34 128.28 [Västra Allén]
31 35 100.39 [Storgatan]
35 31 100.39 [Storgatan]
36 35 136.64 [Storgatan]
35 36 136.64 [Storgatan]
32 33 100.39 [Norra Järnvägsgatan]
33 32 100.39 [Norra Järnvägsgatan]
32 37 75.29 [Förrådet entré]
37 32 75.29 [Förrådet entré]
32 38 78.08 [Dalgatan]
38 32 78.08 [Dalgatan]
38 39 131.07 [Södra Järnvägsgatan]
39 38 131.07 [Södra Järnvägsgatan]

39 12 122.70 [Södra Järnvägsgatan]
12 39 122.70 [Södra Järnvägsgatan]
36 40 287.23 [Storgatan]
40 36 287.23 [Storgatan]
40 41 69.72 [Ågatan]
41 40 69.72 [Ågatan]
41 42 429.45 [Ågatan]
42 41 429.45 [Ågatan]
42 35 47.41 [Ågatan]
35 42 47.41 [Ågatan]
42 43 100.39 [Bro Sporthallen]
43 42 100.39 [Bro Sporthallen]
37 28 167.32 [Genväg Förrådet Campus]
28 37 167.32 [Genväg Förrådet Campus]
30 44 69.72 [Campus L]
44 30 69.72 [Campus L]
44 45 22.31 [Hus L Våning 2]
45 44 22.31 [Hus L Våning 2]
45 46 22.31 [Trapphus L]
46 45 22.31 [Trapphus L]
46 44 0.00 [Trapphus L]
44 46 0.00 [Trapphus L]
28 47 94.81 [Campus D. En labyrint av automatiska dörrar]
47 28 94.81 [Campus D. En labyrint av automatiska dörrar]
28 10 156.16 [Grönborgsgatan]
10 28 156.16 [Grönborgsgatan]