

# Rättningsmanual OU4 DoA-C 2024

Niclas Börlin

2024-03-15

## Contents

<b>0</b>	<b>Dokumenthistorik</b>	<b>1</b>
<b>1</b>	<b>OU4 — Finns en väg?</b>	<b>2</b>
1.1	Beskrivning . . . . .	2
1.2	Rättningsscript . . . . .	2
1.3	Bedömningsmall med kommentarer . . . . .	3

## 0 Dokumenthistorik

- v1.0 2024-03-15** Första publicerade version.
- v1.01 2024-03-15** Lade till info om punkt 5.6 — kommentering av `struct` node.

# 1 OU4 — Finns en väg?

## 1.1 Beskrivning

Från handledarmanualen:

### OU4 Finns en väg?

- Designa och implementera två versioner av datatypen `graph` enligt specificerad gränsyta.
  - Nytt för i år är att alla ska implementera två versioner.
- Implementera bredden-först-sökning i grafen för att avgöra om det finns en väg mellan nod A och nod B.
- Implementera inläsning av grafspecifikationen från en textfil.
  - Nytt för i år är att tolkningen av textfilen förenklas.

## 1.2 Rättningscript

Se HOWTO i git-repot för 5dv149: `.../c/scripts/is_connected/doc/howto.pdf`. Här är listan av test som körs:

No	Directory	Comment
1	1-airmap1	The given map.
2	2-repeated-questions	Same as 1) but with repeated questions, including same start/end.
3	3-directed-graph	The graph is unsymmetric.
4	4-standard-test	Node names differ by last char only.
5	5-big-map	30 edges.
6	6-single-edge	2 nodes, 1 edge.
7	7-single-node	1 node, 1 edge.
x	x-bad-map-format	Bad map format: First edge line has only one node name.
y	y-no-such-file	Missing graph file.

### 1.3 Bedömningsmall med kommentarer

1 Inlämningen är enligt specifikation.

- Saknade filer, m.m.

2 Kodan kompilerar utan varningar.

3 Dynamisk minneshantering

3.1 Kodan saknar minnesfel och -läckor.

3.2 Programmet använder `dlist_kill()` enligt specifikation.

- Specifikationen (det står i beskrivningen av `graph_neighbours()` i `graph.h`) kräver att den `dlist` som returneras ska kunna avallokeras med `dlist_kill()`. De som tidigare gjort **en** implementation och då använder en `dlist` internt för grannskapslistan kan frestas att returnera pekaren till den listan och sedan låta bli att anropa `dlist_kill()` i `find_path()`. Det **fungerar** men bryter mot specifikationen. Den korrekta lösningen är att `graph_neighbours()` bygger upp en kopia (helst en *shallow copy*) av `dlist`an och returnerar den.

I år måste alla implementera två versioner, och då bör det bli uppenbart att `graph_neighbours()` som använder en grannskapsmatris måste skapa en ny `dlist`.

4 Användning

4.1 Programmet fungerar med korrekt kartfil och indata från specifikationen. Motsvarar Test 1 ovan.

4.2 Programmet fungerar med korrekt kartfil och upprepade frågor.

- Test 2. Om den misslyckas här men klarar 4.1 kan det bero på att dom inte nollställer "visited"-status i grafen mellan anrop.

4.3 Programmet fungerar med korrekt riktad kartfil och upprepade frågor.

- Test 3. Misslyckas dom här men klarar 4.1 och 4.2 så kan det vara att dom antar att en båge från A till B implicerar en båge från B till A. Eller att deras kod på annat sätt förutsätter att grafen är symmetrisk (typ oriktad).

4.4 Programmet fungerar med kartfil med gränsfall i antal noder och/eller bågar.

- Motsvarar Test 6 - en båge och Test 7 - en nod.

4.5 Programmet fungerar med alla testade kartfiler.

- Test 4 och Test 5 har nodnamn som liknar varandra `Nod1`, `Nod2`, osv. respektive `n1`, `n2`, osv.
  - Om man gör nåt fel i parsning av nodnamn så inte hela nodnamnet kommer med (t.ex. för att man gör nån egen stränghantering där man antar att end-of-line är två `char \r\n`) så kan det bli fel här.
- Annars fångar upp andra fel än 4.x ovan.

4.6 Programmet avbryts med felmeddelande och `EXIT_FAILURE` för felaktig kartfil.

- Test x. Bör inte vara något problem i år om dom använder mina utdelade parsningsfunktioner. De som kommer från ifjol++ kan dock ha problem.

4.7 Programmet avbryts med felmeddelande och `EXIT_FAILURE` för saknad kartfil.

- Text y.

4.8 Programmet hanterar felaktig frågeinmatning på rimligt sätt.

- Gör en egen bedömning av vad "rimligt sätt" betyder. Inget meddelande alls är t.ex. fel.

4.9 Programmet hanterar specialfallet startnod=slutnod korrekt.

- Testas i Test 2 m.fl. Notera att det finns inget krav på att **graf**en måste ha en båge A till A, bara att programmet ska returnera sant.

4.10 Textutskrifterna är enligt specifikation.

- Var petiga här, åtminstone vid den första inlämningen. Dom har kunnat testa via labres och har dom valt icke-standard-utskrifter så skapar dom merarbete för oss (er). Sånt ska stävas.

## 5 Kodkvalité

5.1 Koden följer kraven i specifikationen.

5.2 Koden är vettigt indenterad.

5.3 Radlängden överstiger ej 120 tecken.

5.4 Filkommentar med korrekt innehåll finns i början av varje inlämnad kodfil.

5.5 Funktionskommentarer finns före varje funktion.

5.6 Koden är rimligt kommenterad.

- Speciellt viktigt är att `struct node`-fälten är kommenterade.

5.7 Namngivning av variabler, funktioner, m.m. är konsistent och ändamålsenlig.

5.8 Funktionerna är rimligt långa (max 60 rader)

5.9 Koden är välstrukturerad och har t.ex. inga globala variabler.

## 6 Rapport

6.1 Framsidan följer specifikationen.

6.2 Rapporten har en innehållsförteckning.

6.3 Rapporten är indelad i sektioner med numrerade rubriker.

6.4 Sidorna är numrerade.

6.5 Introduktionen och problembeskrivningen är acceptabel.

6.6 Användarhandledning inkl. exempel finns och fyller sitt syfte.

6.7 Beskrivningen av hur grafen är representerad fyller sitt syfte.

6.8 Beskrivningen av vilka funktioner i grafen gränssnitt som är implementerade finns.

6.9 Beskrivningen av övriga datatyper finns, inkl. gränssyta.

6.10 Beskrivning av informationsflödet mellan de centrala funktionerna finns.

- Utgår!

6.11 Algoritmbeskrivningar finns för de centrala algoritmerna.

- Viktigast är att algoritmen är **läsbar**, **tydlig** och i formen **punktlista** — inga algoritmer i löptext eller ”den här noden”...
- För denna gång ska minst `find_path` finnas med. Jag ska ändra spec:en till att även inkludera `construct_graph` el. motsvarande.

6.12 Testkörningar är dokumenterade.

6.13 Arbetsfördelningen är beskriven och acceptabel.

6.14 Reflektionerna är rimligt omfattande.

6.15 Rapporten uppfyller rimliga krav vad gäller språk och struktur