

Naam:

Belangrijk: Schrijf je antwoorden kort en bondig op in de daartoe voorziene velden. Indien nodig gebruik je de achterkant van het blad (met duidelijke vermelding van de vraag). Het gebruik van een eenvoudig rekenmachine is toegestaan. Andere hulpmiddelen (smartphone, laptop, cursusnota's, ...) zijn niet toegestaan tijdens het examen. Elke theorie-vraag staat op 2 punten (totaal op 28). De oefeningen staan in totaal op 6 punten. Het geheel staat op 34 punten.

1. Introductie

[... / 2]

Leg uit, in je eigen woorden, wat verificatie en validatie zijn.

- a. Validatie: Komen de requirement specificaties overeen met wat de klant nodig heeft? Dit is belangrijk voor de "analysis" activiteit
 - i. Are we building the right product
- b. Verificatie: Komt het product overeen met de requirement specificaties? Dit is belangrijk voor de "testing" activiteit
 - i. Are we building the product right

2. Project Management

[... / 2]

Geef de definitie voor een Squad, een Tribe, een Chapter, en een Guild in het SCRUM model van Spotify.

- a. Squad: Scrum team
- b. Tribe: Groepering van scrum teams die aan gerelateerde functionaliteit/componenten werken
- c. Chapter: Leden van verschillende squads binnen één Tribe die gelijkaardige expertise hebben
- d. Guild: Leden van verschillende squads van verschillende tribes die gelijkaardige interesses hebben

3. Requirements

[... / 2]

Geef de INVEST criteria voor user stories en leg ze kort uit.

- a. Independent: Stories moeten onafhankelijk zijn van elkaar en mogen geen interdependencies hebben
- b. Negotiable: Te veel detail in een story duwen maakt de interactie met de klant moeilijker
- c. Value: elke story moet waardevol zijn voor een klant
- d. Estimate: Stories moeten klein genoeg zijn zodat men de tijd kan inschatten
- e. Small: Stories moeten klein genoeg zijn zodat men ze kan afwerken in één iteratie
- f. Testable: Acceptatie criteria moeten aanwezig zijn

Naam:

4. Domain models [... / 2]

Leg het verschil uit tussen een social fork en een variant fork.

- a. Social fork: gemaakt door een developer om het bestaande product uit te breiden met nieuwe functionaliteit die later terug wordt toegevoegd aan het originele product, daarna wordt aan de fork niet meer verder gewerkt.
- b. Variant fork: gemaakt door een developer om een nieuw product te maken met nieuwe functionaliteit, deze fork blijft bestaan

5. Testing [... / 2]

Wat is alpha-beta testing? Wanneer wordt het gebruikt?

- a. testing technieken met veel ongeïdentificeerde gebruikers voor software die "off-the-shelf" wordt verkocht
- b. Alpha: users worden uitgenodigd on-site in een gecontroleerde omgeving
- c. Beta: software wordt vrijgegeven aan bepaalde gebruikers in een real-world setting

6. Design by contract [... / 2]

Wat is behavioural subtyping?

- a. Dit is het Liskov Substitution principe
- b. Definitie:
 - i. If S is a subtype of T, then objects of type T may be replaced with objects of type S (i.e., objects of type S may substitute objects of type T) without altering any of the desirable properties of that program (correctness, task performed, etc.). Each instance of a superclass may be replacing by instances of a superclass without altering the correct behaviour.
- c. Belang:
 - i. It defines the rules for inheritance and therefore subcontracts.

Naam:

7. Formal specifications

[... / 2]

Waarom is een UML klassediagram een semi-formele specificatie?

- a. de syntax van UML diagrammen ligt vast, maar de semantiek bevat ambiguiteiten
- b. vb. Van ambigutiteieten
 - i. de # is protected, maar wordt niet gezegd welke versie van protected wordt gebruikt
 - ii. het protocol van een klasse definitie moet niet volledig zijn. Er kunnen dus methodes en attributen bestaan die niet op het UML diagram genoteerd staan

8. Software Architecture

[... / 2]

Wat is coupling? Wat is cohesion? Hoe moet je ermee omgaan indien je een goed design wilt bekomen?

- a. coupling: hoezeer is een component verbonden met andere componenten via een connector
- b. cohesion: hoezeer vormen de onderdelen van een component een samenhangend geheel
- c. Je wilt coupling minimaliseren en cohesion maximaliseren, deze afweging laat toe om architecturen te evalueren

9. Quality Control

[... / 2]

Geef en definieer de 5 niveaus van Capability Maturity Model Integration (CMMI)?

- a. Level 1: Initial (Ad hoc): Geen QA procedures, kwaliteit is op goed geluk
 - i. Kwaliteit hangt af van de individuen
- b. Level 2: Managed (Repeatable): Formele QA procedures aanwezig
 - i. kwaliteit hangt af van de project managers
- c. Level 3: Defined: QA process is gedefinieerd op het niveau van de organisatie
 - i. Organisatie is pro-actief
- d. Level 4: Quantitatively managed: QA process + quantitative data verzameling
 - i. Quantitatieve data is noodzakelijk voor verbetering
- e. Level 5: Optimising: Observaties worden gebruik om het QA process te verbeteren
 - i. Continue verbeteringen

Naam:

10. Software Metric

[... / 2]

Wat zijn de belangrijkste voor en nadelen van Function Points?

Voordelen:

- Kan gemeten worden na design en voor implementatie,
- onafhankelijk van de programmeertaal,
- meet de hoeveelheid functionaliteit,
- goed voor data-processing toepassingen,
- FP is beter dan LOC om productiviteit te meten

Nadelen:

- subjectief,
- kan niet automatisch berekend worden,
- betekenisloos (wat betekent 2000 FP?),
- voor estimation effort te meten is FP niet goed want deze komt te laat in de lifecycle

11. Refactoring

[... / 2]

Leg uit waarom add class/method/attribute "behaviour preserving" moet zijn?

- a. Het toevoegen zijn enkel declaraties. Na het toevoegen van een methode wordt die nog niet opgeroepen, na het toevoegen van een attribuut wordt die nog niet geschreven of lezen, na het toevoegen van een klasse worden nog geen instances gemaakt

12. Conclusie

Leg de twee voornaamste stappen uit van test amplification: "input amplification" en "assertion amplification".

[... / 2]

- a. input: (Niet zeker, op de slide stond geen text, ik heb dit uit een paper: M. Adbi et al 2022) The existing test code is altered to cover more paths than before, with a focus on existing the current tests to also cover boundary conditions.
- b. assertion: (Niet zeker, op de slide stond geen text, ik heb dit uit een paper: M. Adbi et al 2022) Extra asserts worden toegevoegd om de te verwachten output van de nieuwe paths te testen. Om de uitkomst van de assert te weten (true of false) wordt het bestaande systeem als orakel gebruikt.

Naam:

13. Indien je het artikel “No Silver Bullet” hebt gelezen: [... / 2]

Waarom is object-georiënteerd programmeren geen “silver bullet”?

- a. OO removes a higher order kind of accidental difficulty and allows a higher order expression of design. Nevertheless OO can do no more than to remove all accidental difficulties from the expression of the design. The complexity of the design itself is essential and such attacks make no change whatever in that.

14. Indien je het artikel “Killer Robot” hebt gelezen: [... / 2]

Waarom was het gebruik van het waterfall process zo desastreus in dit geval?

- a. (From KillerRobotCase/articel-4.html) The waterfall model goes through definite stages of development. As the project passes from one stage to the next, there are limited opportunities to change earlier decisions. A drawback of this approach is that potential users are not able to interact iwth the sytem until very late in the process.
- b. The Robot project involves a high degree of interaction, both between the robot components and between the robot and the operator. Since operator interaction with the robot is so important, the interface cannot be designed as an afterthought.

Naam:

15. Oefening Testing

[... / 6]

Het kortstepad algoritme van Dijkstra is jullie wellicht bekend. Dit graaf-algoritme berekent, gegeven een gerichte graaf Graph, waarin de afstand tussen ieder tweetal verbonden knopen van Graph ten minste 0 bedraagt, voor een bepaalde beginknoop source in Graph, de kortste afstand van die beginknoop tot alle punten van de graaf.

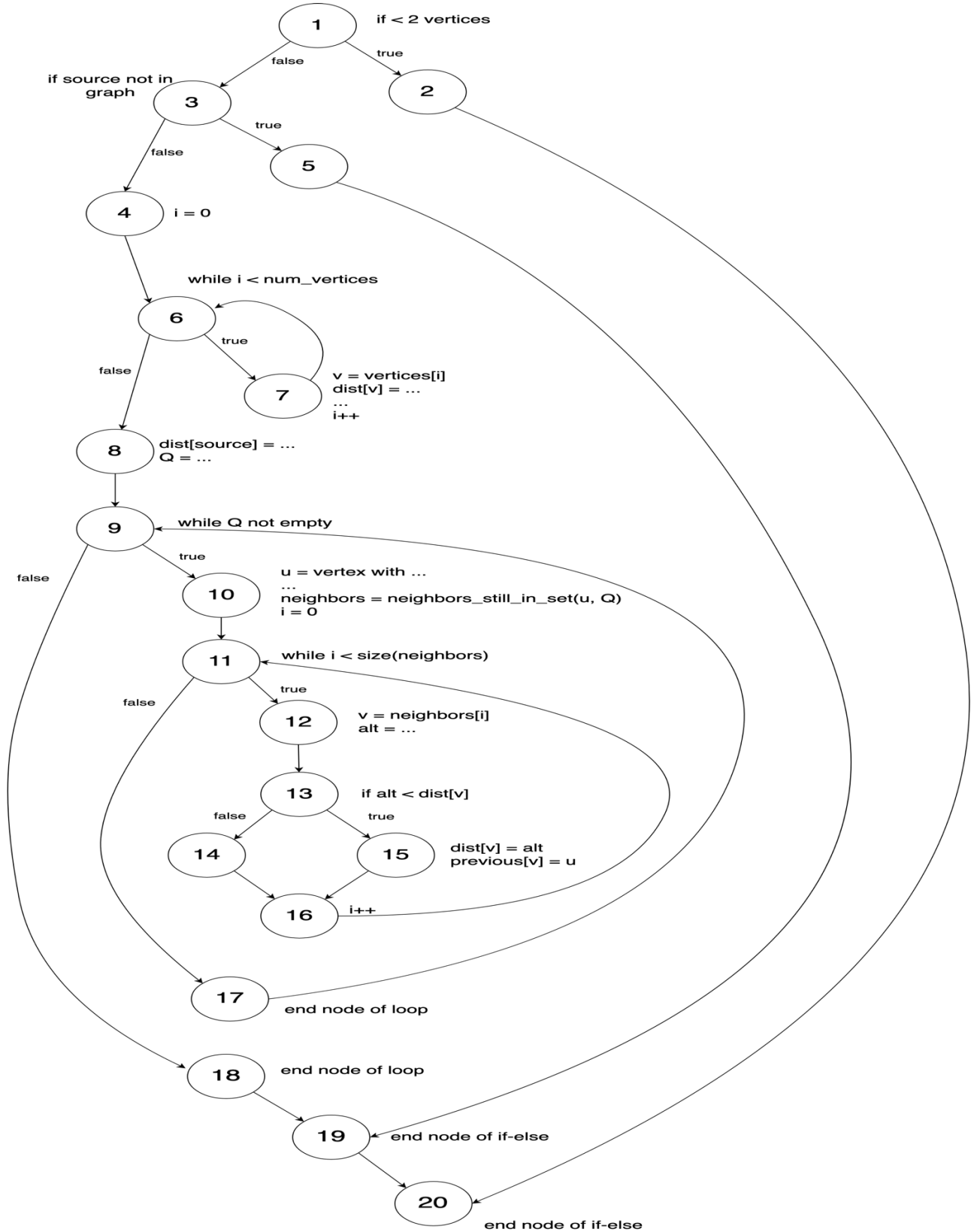
Hieronder vinden jullie het algoritme in pseudo-code:

```
function Dijkstra(Graph, source):  
  if Graph has less than 2 vertices or source vertex is not in Graph:  
    return  
  
  for each vertex v in Graph:  
    dist[v] = infinity  
    previous[v] = undefined  
  
  dist[source] = 0  
  Q = the set of all vertices in Graph  
  
  while Q is not empty:  
    u = vertex in Q with smallest value in dist array  
    remove u from Q  
  
    for each neighbor v of u that is still in Q:  
      alt = dist[u] + edge_length(u, v)  
      if alt < dist[v]  
        dist[v] = alt  
        previous[v] = u  
  
  return dist, previous
```

Naam:

Oplossing optie A

a) Teken de control flow graph (CFG) voor bovenstaande functie. **Zorg dat je voor elke node in de CFG aangeeft met welke lijnen in het algoritme deze overeenkomt.**



Naam:

b) Bereken de cyclomatische complexiteit, en geef kort aan hoe je hiertoe gekomen bent.

a. slide 18: $CC = \#edges - \#nodes + 2 = \# \text{ binary conditions} + 1$

b. $\# edges = 25, \#nodes = 20 \Rightarrow CC = 25-20+2 = 7$

c. $\# \text{ bin cond} = 6 \Rightarrow CC = 6 + 1 = 7$

c) Hoe verhoudt het aantal onafhankelijke paden zich tot de cyclomatische complexiteit? Waarom?

a. slide 20: cyclomatische complexiteit \leq aantal onafhankelijke paden

d) Bepaal een volledige verzameling van onafhankelijke paden en nummer ze. **Kies je paden zodanig dat ze ook genomen kunnen worden.**

Geef aan welke input vereist is voor elk van de onafhankelijke paden die je in vraag (c) hebt opgesomd. (**Gebruik de nummering voor verwijzingen** naar elk onafhankelijk pad.)

➤ 4 paden:

⇒ 1-2-20

- Minder dan 2 nodes

⇒ 1-3-5-19-20

- Source niet in graph

⇒ 1-3-4-6-7-8-9-10-11-12-13-14-16-11-17-9-18-19-20

⇒ 1-3-4-6-7-8-9-10-11-12-13-15-16-11-17-9-18-19-20

➤ 3 paden:

⇒ 1-2-20

- Minder dan 2 nodes

⇒ 1-3-5-19-20

- Source niet in graph

⇒ 1-3-4-6-7-8-9-10-11-12-13-14-16-11-12-13-15-16-11-17-9-18-19-20

➤ Er is tenslotte een input die onderstaand traject door de code induceert:

⇒ > 2

⇒ Source in graph

⇒ Neem loop 4x voor A,B,C,D

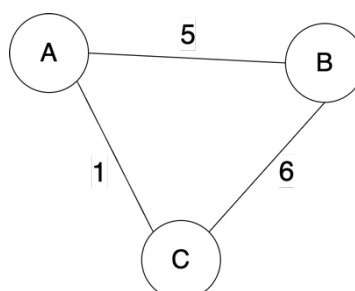
⇒ While iteratie met $u = A$

- For iteratie met $v = B$, if-statement is true
- For iteratie met $v = C$, if statement is true

⇒ While iteratie met $u = C$

- For iteratie met $v = A$: A niet meer in Q
- For iteratie met $v = B$: geen update want if-statement is false
- ...

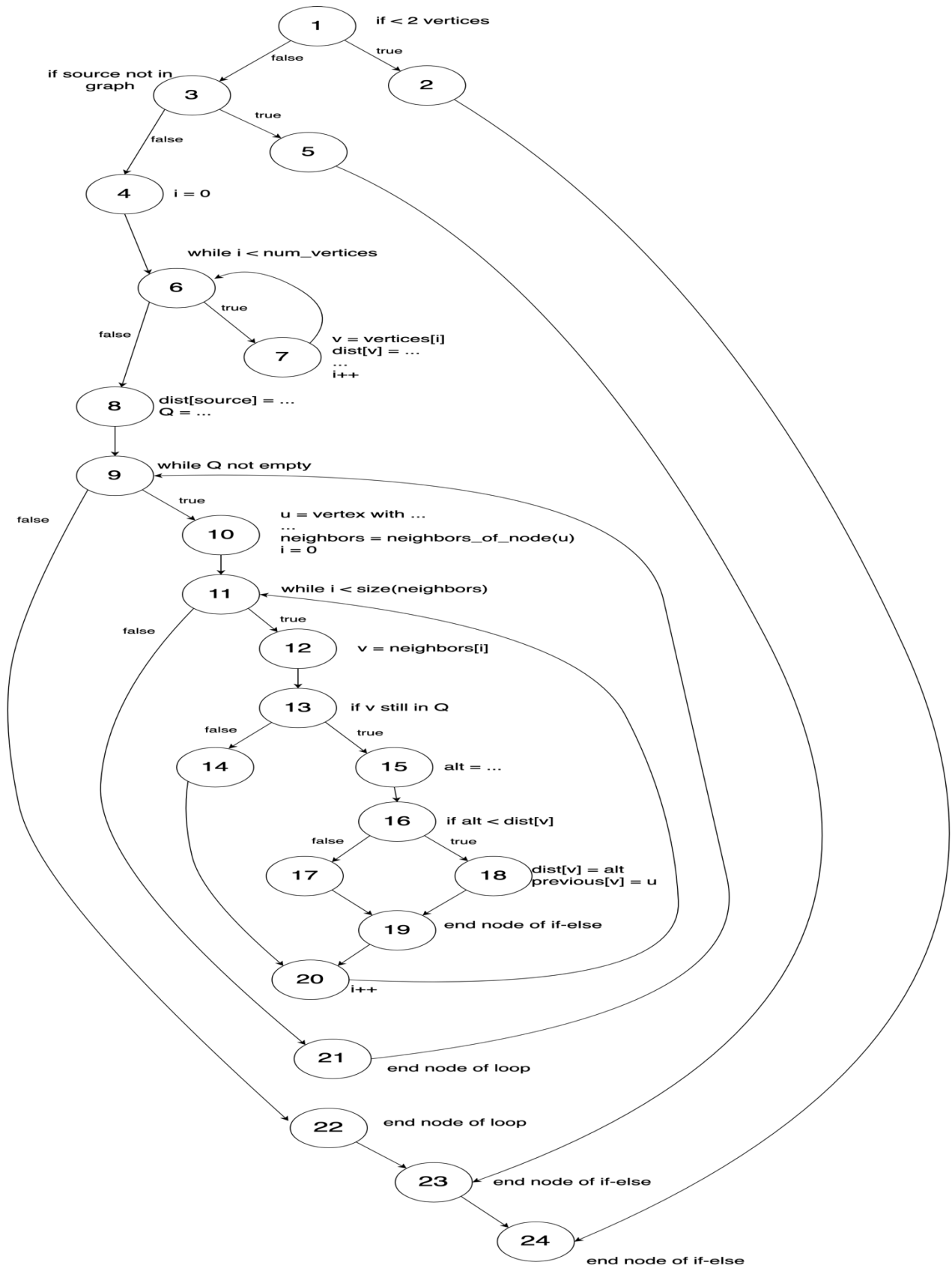
⇒ ...



Naam:

Oplossing Optie B

a) Teken de control flow graph (CFG) voor bovenstaande functie. **Zorg dat je voor elke node in de CFG aangeeft met welke lijnen in het algoritme deze overeenkomt.**



Naam:

b) Bereken de cyclomatische complexiteit, en geef kort aan hoe je hiertoe gekomen bent.

a. slide 18: $CC = \#edges - \#nodes + 2 = \# \text{ binary conditions} + 1$

b. $\# edges = 24, \#nodes = 30 \Rightarrow CC = 30 - 24 + 2 = 8$

c. $\# \text{ bin cond} = 7 \Rightarrow CC = 7 + 1 = 8$

c) Hoe verhoudt het aantal onafhankelijke paden zich tot de cyclomatische complexiteit? Waarom?

a. slide 20: cyclomatische complexiteit \leq aantal onafhankelijke paden

d) Bepaal een volledige verzameling van onafhankelijke paden en nummer ze. **Kies je paden zodanig dat ze ook genomen kunnen worden.**

Geef aan welke input vereist is voor elk van de onafhankelijke paden die je in vraag (c) hebt opgesomd. (**Gebruik de nummering voor verwijzingen** naar elk onafhankelijk pad.)

➤ 5 paden:

⇒ 1-2-24

- Minder dan 2 nodes

⇒ 1-3-5-23-24

- Source niet in graph

⇒ 1-3-4-6-7-8-9-10-11-12-13-14-20-11-21-9-22-23-24

⇒ 1-3-4-6-7-8-9-10-11-12-13-15-16-17-19-20-11-21-9-22-23-24

⇒ 1-3-4-6-7-8-9-10-11-12-13-15-16-18-19-20-11-21-9-22-23-24

➤ 3 paden:

⇒ 1-2-20

- Minder dan 2 nodes

⇒ 1-3-5-19-20

- Source niet in graph

⇒ 1-3-4-6-7-8-9-10-11-12-13-14-20-11-12-13-15-16-17-19-20-11-12-13-15-16-18-19-20-11-21-9-22-23-24

➤ Voor input die het derde, vierde, vijfde pad neemt:

⇒ > 2

⇒ Source in graph

⇒ Neem loop 4x voor A,B,C,D

⇒ While iteratie met $u = A$

- For iteratie met $v = B$, if-statement is true
- For iteratie met $v = C$, if statement is true

⇒ While iteratie met $u = C$

- For iteratie met $v = A$: A niet meer in Q
- For iteratie met $v = B$: geen update want if-statement is false
- ...

⇒ ...

