StuDocu.com

Exam 2015, questions and answers

Data Structures and Algorithms (University of Pretoria)

StuDocu is not sponsored or endorsed by any college or university
Downloaded by Carl De Witt (carlisandriod@gmail.com)

Question 1 /Vraag 1 ..... (41 marks)

1.1 Aanvaar 'n B-boom van hoogte 71 en $m = 9$. Wat is die minimum aantal sleutels wat hierdie boom moet bevat? (1)
Assume a B-tree of height 71 and $m = 9$. What is the minimum number of keys that this tree must contain?

1.2 Beskou die volgende ontbrekende `BTreeNode` klas. Objekte van hierdie klas is bedoel om as nodusse in 'n B-boom gebruik te word: (4)
Consider the following partial `BTreeNode` class. Objects of this class are meant to be used as nodes in a B-tree:

```
public class BTreeNode<T>
{
        //Constructors, initialization and additional methods omitted.

        int m; // The order of the tree
        int numKeys; //The number of keys
        public Comparable<T> keys[]; //The keys
        public BTreeNode<T> children[]; //The children.
}
```

Skryf minimum kode vir 'n rekursiewe metode `int countNodes(BTreeNode<T> n)` om die aantal nodusse in 'n B-boom te tel.
Write minimal code for a recursive method `int countNodes(BTreeNode<T> n)` to count the number nodes in a B-tree.

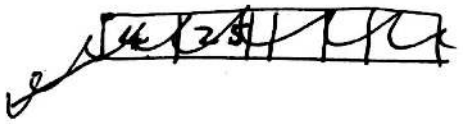
```
public int countNodes(BTreeNode<T> n)
{
  int count;
  int keep=0;
  if(n!=null)
     count=1;
  else
    return 0;
  while(keep<=n.numKeys && n.children[keep]!=null)
  {
    count+=countNodes(n.children[keep]);
    keep++;
  }

    return count;
}
```

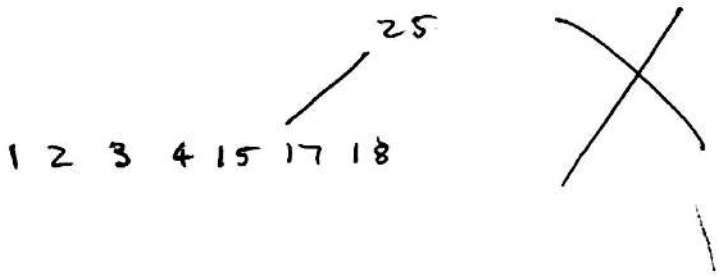

Downloaded by Carl De Witt (carlisandriod@gmail.com)

1.3 Indien die wortel nodus in 'n B*-boom moet splits sal daar nie genoeg sleutels wees om te verseker dat die kinders van die nuwe wortel nie ondervloei nie. Bespreek 'n strategie om hierdie geval te hanteer. (2)
If the root node in a B-tree needs to split then there won't be enough keys to ensure that the children of the new root do not underflow. Discuss a strategy to handle this case.*

Then one would ~~check to see if~~ redistribute the keys from the root the ~~left~~ child and its sibling into one node and discard the sibling.

1.4 Beskou die B*-boom in `figuur 1`. Maak gebruik van die direkte voorganger en verwyder die sleutel 10 uit hierdie boom. Teken die finale boom nadat die voorgenoemde verwydering suksesvol plaasgevind het. (8)
Consider the B-tree in `figure 1`. Use the direct predecessor and delete the key 10 from this tree. Draw the final tree after the aforementioned delete has successfully taken place.*





Downloaded by Carl De Witt (carlisandriod@gmail.com)

1.9 Aanvaar 'n trie struktuur wat die volgende woorde bevat:
Assume a trie structure containing the following words:

`no, non, nonsense, moose, meese, car, cat, cash, cashier, butter`

Elke interne nodus bevat die "einde-van-woord" karakter en slegs die karakters wat nodig is alfabeties gerangskik. Antwoord die volgende:
Each internal node contains the "end-of-word" character and only the necessary characters arranged alphabetically. Answer the following:

a) Wat is die hoogte van hierdie trie?
What is the height of this trie? (1)

6

b) Hoeveel interne nodusse is daar op die pad na die woord `cashier`?
How may internal nodes are there on the path to the word `cashier`? (1)

4

Question 2 /Vraag 2 ..... (32 marks)

BELANGRIK: Waar daar 'n keuse is tussen punte wat volgende verwerk word, kies hulle **alfabeties**.
IMPORTANT: *Wherever there is a choice among vertices to be processed next, choose them **alphabetically**.*

2.1 Aanvaar 'n pseudografiek G met 10 000 lyne. Wat is die minimum aantal punte wat G mag hê. (1)
Assume a pseudo graph G with 10 000 edges. Wat is the minimum number of vertices that G may have?

1

2.2 Die totale graad van 'n ongerigte grafiek $G = (V, E)$ is die som van die grade van al die punte in V. Bewys dat as G se totale graad ewe is dan bevat V 'n ewe aantal punte met onewe grade. Hanteer die punte met ewe grade en die punte met onewe grade as lede van twee disjunkte versamelings. (3)
The total degree of an undirected graph $G = (V, E)$ is the sum of the degrees of all of the vertices in V. Prove that if the total degree of G is even then V will contain an even number of vertices with uneven degrees. Handle the vertices with even degrees and the vertices with uneven degrees as members of two disjoint sets.

Downloaded by Carl De Witt (carlisandriod@gmail.com)

2.3 Aanvaar die volgende beskrywing van die grafiek G: Dit is enkelvoudig, ongerig, samehangend, bevat geen siklusse nie en het 5000 lyne. Hoeveel punte het G? (1)
Assume the following description of the graph G: It is simple, undirected, connected, contains no cycles and has 5000 edges. How many vertices does G have?

5001

2.4 Wat is die maksimum aantal vorentoe lyne wat enige samehangende grafiek G kan hê? (1)
What is the maximum number of forward edges which any connected graph G may have?

|V|

2.5 Die pseudokode vir 'n breedte eerste deurstappings algoritme kan gegee word as: (1)
The pseudo code for the breadth first search graph traversal algorithm could be given as:

```
BFS()
{
        for all vertices v
                v.visited=false

        while there is a v such that v.visited==false
        {
                v.visited = true
                q.enqueue(v)
                while !q.isEmpty()
                {
                        v = q.dequeue
                        output(v)
                        for all vertices u adjacent to v
                        {
                                if !u.visited
                                u.visited = true
                                q.enqueue(u)
                        }
                }
        }
}
```


Downloaded by Carl De Witt (carlisandriod@gmail.com)

In hierdie algoritme, instede daarvan om vir punte nommers te gee om aan te dui dat hulle besoek was, word 'n veld `visited` dienooreenkomstig na waar of vals gestel. In hierdie algoritme word 'n punt slegs in die ry geplaas nadat die punt se `visited` veld na waar gestel is. Wat sal die implikasie wees indien die `visited` veld na waar gestel word eers nadat 'n punt wat verwerk moet word uit die ry verwyder word?
In this algorithm, instead of giving vertices numbers to indicate that they have been processed, the field `visited` *is used and set to true or false accordingly. In this algorithm a vertex is only placed in the queue after its* `visited` *field has been set to true. What would the implication be when the* `visited` *field is set to true only after a vertex to be processed has been dequeued?*

Later on when it deques the vertex it would still see it as not visited so it would be processed twice.

2.6 Dijkstra se kortste pad algoritme hoef nie uit te voer totdat die struktuur `toBeChecked` leeg is nie. Wanneer sal dit geldig wees om die algoritme te staak voordat `toBeChecked` leeg is? (1)
Dijkstra's shortest path algorithm does not need to execute until the structure `toBeChecked` *is empty. When will it be valid to halt the algorithm before* `toBeChecked` *is empty?*

when the ~~path~~ shortest path to the ~~node~~ destination vertex you is found.

2.7 Gee een toepassing waarvoor topologiese sortering en die inkleur van grafieke gebruik kan word. (1)
Give one application for which both topological sorting and graph coloring may be used.

~~when trying to find the minimum number of nonoverlapping sets of vertices~~ when trying to find the order at which the vertices should be processed.

2.8 Beskou die grafiek in *figuur 6* en antwoord die volgende:
Consider the graph in figure 6 and answer the following:

a) Gee die volgorde waarin punte besoek sal word indien diepte eerste deurstapping op die grafiek toegepas word. (2)
Give the order in which vertices will be visited if depth first search was performed.

A C B G H J ~~Z~~ M D F

2

b) Lewer volledige kommentaar oor die samehang van die grafiek. Motiveer jou observasies. (2)
Comment fully on the connectivity of the graph. Motivate your observations.

This is a strongly connected graph. It is also biconnected. ~~because for most~~

Downloaded by Carl De Witt (carlisandriod@gmail.com)

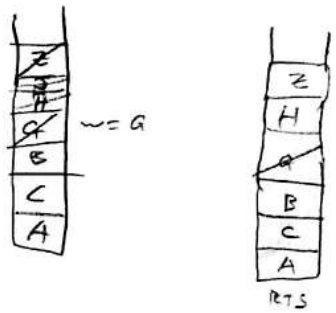
c) Pas Dijkstra se kortste pad algoritme toe met die begin punt **A** en vul die afstande vir die punte in die volgende tabel: (5)
*Apply Dijkstra's shortest path algorithm on this graph with starting vertex **A** and fill in the distances for the vertices in the following table:*

Vertex	Distance from A
A	0
B	17
C	10
D	∞
F	∞
G	27
H	25
J	26
M	20
Z	11

4'12

d) Pas die `strongDFS` algoritme toe op die grafiek en voltooi die volgende tabel deur die waardes van `pred` vir elk van die gelyste punte neer te skryf: (5)
Perform the `strongDFS` algorithm on the graph and complete the following table by filling in the values for `pred` for each of the listed vertices:

Vertex	Pred
A	1
B	1
C	1
D	4
F	4
G	4
H	1
J	1
M	1
Z	1



4

e) Kan 'n topologiese sortering uitgevoer word op hierdie grafiek? Indien wel, gee slegs die sortering en indien nie, motiveer waarom nie. (2)
Can topological sorting be applied to this graph? If your answer is yes, simply give the ordering of the vertices, otherwise motivate why not.

Yes. No it can't. There is a cycle between A Z & M

2

Downloaded by Carl De Witt (carlisandriod@gmail.com)

2.9 Verduidelik waarom die chromatiese nommer vir 'n tweeledige grafiek twee is. (2)
Explain why the chromatic number of a bipartite graph is two.

Because bi means 2 indicates that it is a graph of 2 vertices.

2.10 Beskou die grafiek in *figuur 7*. Aanvaar dat hierdie grafiek ingekleur moet word met die `BrelazColoringAlgorithm` prosedure. Die beskikbare kleure word in die volgende volgorde gegee: `[blou, pers, rooi, silver, goud, groen]`. Pas hierdie algoritme toe op die grafiek en voltooi die tabel hierdonder deur die korrekte waardes vir die versadigingsgrade vir elk van die gelyste punte in te vul: (5)
Consider the graph in figure 7. Assume that this graph should be coloured using the `BrelazColoringAlgorithm` procedure. The available colours are given in the following order: `[blue, purple, red, silver, gold, green]`. Apply this algorithm to the graph and complete the table below by filling in the correct saturation degrees for each of the listed vertices.

Vertex	saturationDeg
A	5
G	1
H	2
I	1
L	2
M	0
O	2
R	0
S	0
T	1

Question 3 / Vraag 3 ..... (25 marks)

3.1 Vir kwadratiese soek word i aanvanklik na 1 geinitialiseer en nie 0 nie. Waarom is dit die geval? (2)
For quadratic probing, the probe i is initialized with the value 1 and not 0. Why is this the case?

Because our formula for it to increase or decrease by i^2 i could not be 0.

3.2 Aanvaar 'n hutstabel van grote 1000 waar die hutsfunksie $h(K) = k*2 mod 1000$ gebruik word met kwadratiese soek as die botsingshantering strategie. Beantwoord die volgende: Wat sal die derde laaste adres in die tabel wees wat probeer sal word vir 'n sleutel met die hutswaarde 401?
Assume a hash table of size 1000 where the hash function $h(K) = (k*2 mod 1000$ is used and quadratic probing as the collision resolution strategy. Answer the following:

a) Waarom is die bogenoemde huts strategie nie 'n goeie strategie nie? (1
Why is the hashing strategy mentioned above not a good strategy?

It becomes inefficient when searching

Downloaded by Carl De Witt (carlisandriod@gmail.com)

b) Gestel 'n sleutel word aanvanklik na die adres 500 gehuts. In die geval van botsings, wat sal die derde laaste adres wees wat probeer sal word om die sleutel in die tabel te plaas? (1)
Assume a key is hashed to the address 500. In the case of collisions, what will be the third last address tried to insert the key into the table?

3.3 Dit is moontlik om 'n implementasie van 'n hutstabel te skep, wat gebruik maak van emmer-adresering, as 'n 2-dimensionele skikking. Elke ry indeks in hierdie skikking stel die adres voor van 'n hele emmer (die sleutels wat na dieselfde adres gaan word in dieselfde ry geplaas). As daar byvoorbeeld gepraat word van 'n $X \times Y$ matriks, dan stel die indeks 2 die derde emmer voor wat Y elemente kan bevat. Aanvaar die volgende sleutels moet in die gegewe volgorde in 'n hutstabel van grote 5 geplaas word met emmers van grote 3. Gebruik deling as jou hutsfunksie en beantwoord die volgende vrae: (6)
It is possible to implement a hash table, making use of **bucket addressing**, *as a two dimensional array. Each row index in this matrix is the address of an entire bucket (the keys hashed to the same address are placed in the same row). For example, in a $X \times Y$ matrix, the index 2 corresponds to the third bucket which can take up to Y elements. Assume the following keys need to be placed, in the given order, into a hash table of size 5 with buckets of size 3. Use division as your hash function and answer the following questions:*

`6, 12, 18, 8, 2, 14, 5, 16, 24, 77, 301,444`

Kombineer die emmer adresering skema met kwadratiese soek en skryf die hutswaardes (emmer adresse) vir die sleutels in die tabel hieronder. Indien 'n sleutel nie in die tabel geplaas kan word nie, gee die adres as **NA**.
Combine the bucket addressing scheme with quadratic probing and write the addresses (bucket addresses) for each of the keys in the table below. If a key cannot be placed in the table, give the address as ***NA***.

Key	Address
6	
12	
18	
8	
2	
14	
5	
16	
24	
77	
301	
444	

Downloaded by Carl De Witt (carlisandriod@gmail.com)

3.4 Aanvaar die volgende sleutels is alreeds gerangskik om Cichelli se perfekte hutsmetode uit te voer:
Assume the following keys have already been sorted for Cichelli's perfect hash method.

```
alpha
beta
theta
sigma
kappa
omicron
nu
```

Pas Cichelli se metode toe met `max=3` en antwoord die vrae wat volg:
Apply Cichelli's method with ***max=3*** *and answer the questions that follow:*

a) Vul die hutwaardes vir elk van die sleutels in die tabel hieronder in. (7)
Fill in the hash values for each of the keys in the table below.

Key	Hash Value
alpha	4
beta	5
theta	2
sigma	1
kappa	3
omicron	7
nu	6

b) Vul die g-waardes vir elk van die karakters in die tabel hieronder. (7)
Fill in the g-values for each of the characters in the table below.

Character	G value
a	7
b	3
k	0
n	0
o	0
s	0
t	0

c) Afgesien van sleutels van identiese lengte wat identiese eerste en laaste karakters deel, gee een ander scenario waar Cichelli se algoritme definitief sal faal. (1)
Apart from keys of identical length and sharing identical first and last characters, give one other scenario where Cichelli's algorithm will definiately fail.

Downloaded by Carl De Witt (carlisandriod@gmail.com)

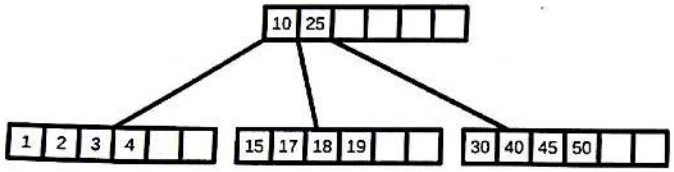


Figure 1: B* tree

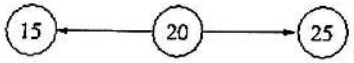


Figure 2: v-h tree

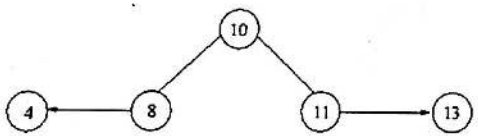


Figure 3: v-h tree

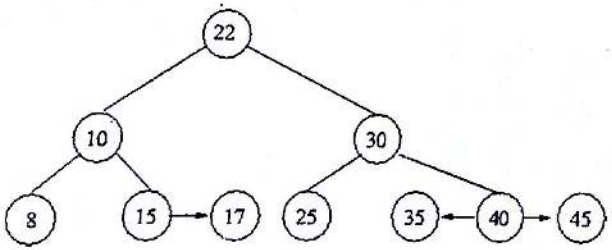


Figure 4: v-h tree

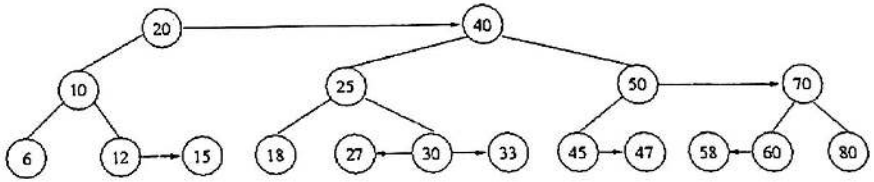


Figure 5: v-h tree

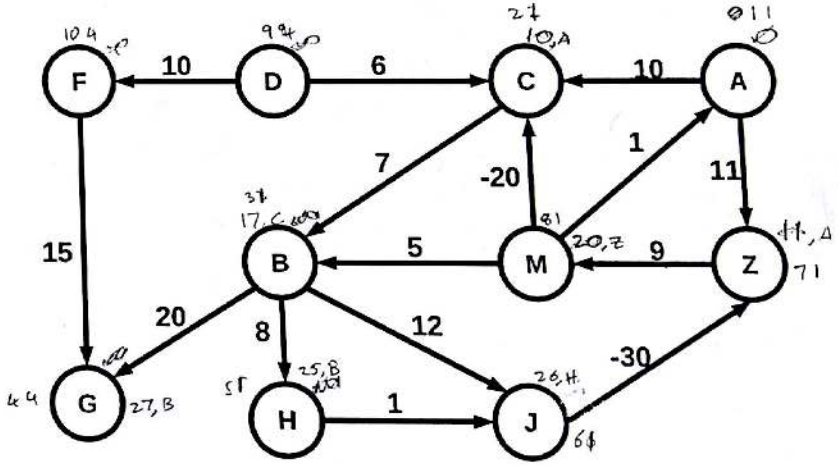


Figure 6: Graph

Downloaded by Carl De Witt (carlisandriod@gmail.com)

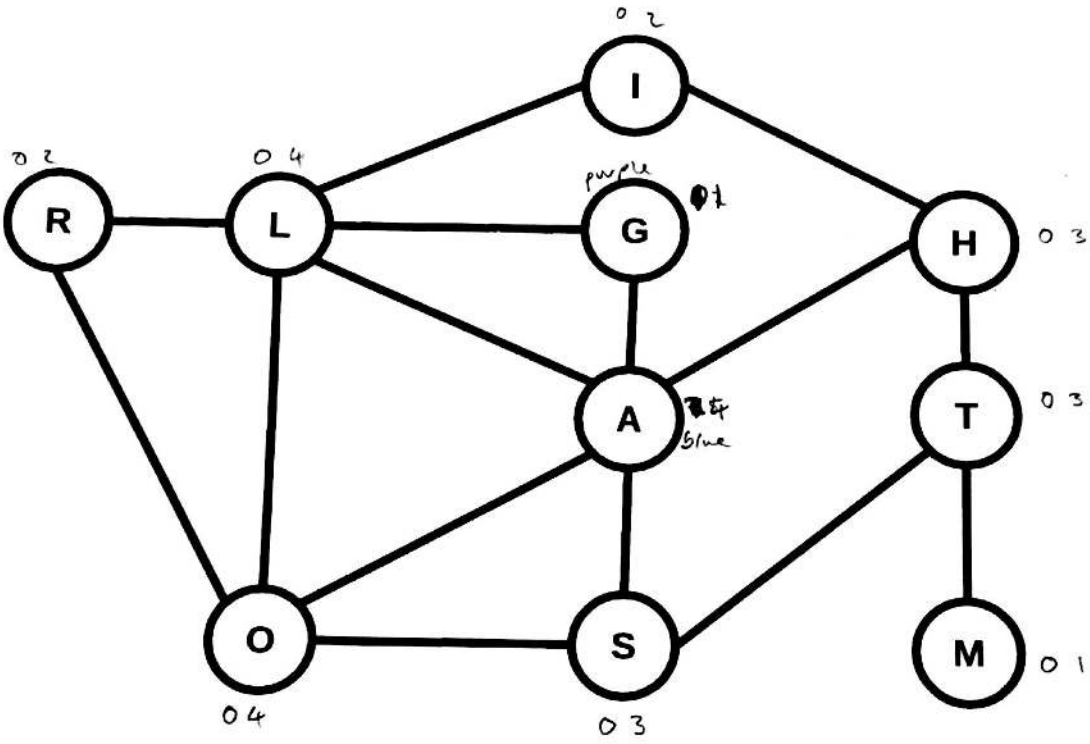


Figure 7: Graph

Downloaded by Carl De Witt (carlisandriod@gmail.com)