Student ID: .....

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI

VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS – 2014

TRIMESTER 1

COMP 261
ALGORITHMS and DATA STRUCTURES

Time Allowed: TWO HOURS

Instructions: Closed Book

Attempt ALL Questions.

Answer in the appropriate boxes if possible — if you write your answer elsewhere, make it clear where your answer can be found.

The exam will be marked out of 120 marks.

Only silent non-programmable calculators or silent programmable calculators with their memories cleared are permitted in this examination.

Non-electronic foreign language dictionaries are permitted.

Alphabetic order: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Questions

	Marks
1. Graphs and String Search	[25]
2. Graphics	[15]
3. Parsing	[30]
4. B and B+ Trees	[25]
5. Maximum Flow	[10]
6. Compression	[15]

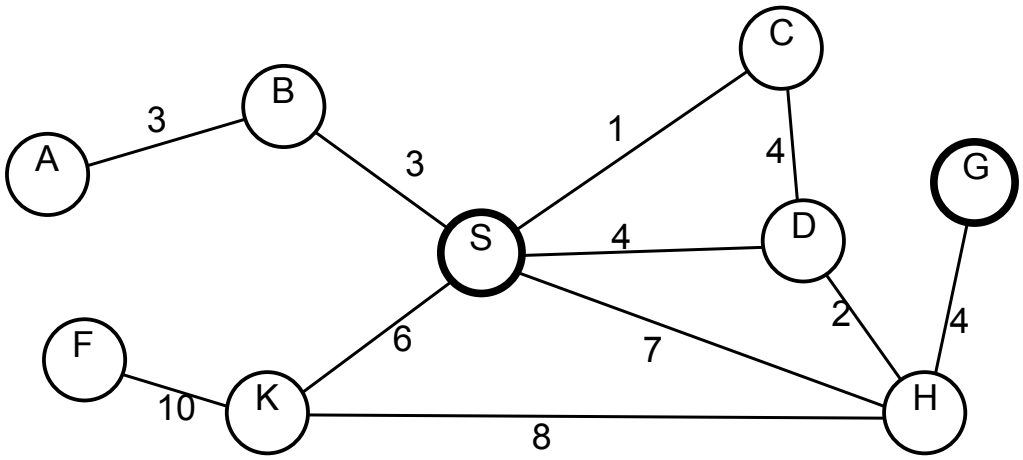
Question 1. Graphs and String Search [25 marks]

(a) [7 marks] Dijkstra's algorithm

Show how Dijkstra's algorithm finds the shortest path from **S** to **G** in the weighted undirected graph below.

On the facing page, you should show the queue and the solution:

- list each element that is added to the priority queue, showing the node, from-node, and the priority value.
  (The first element put on and removed from the priority queue is shown as an example.)
- add neighbours in alphabetical order
- indicate the order the elements are removed from the priority queue (eg, by numbering them)
- list the nodes of the shortest path found.



(Question 1 continued on next page)

(Question 1 continued)

Queue:

```
[Node ← From:  Priority Value] When removed
[S    ← null:      0    ]             #1
```

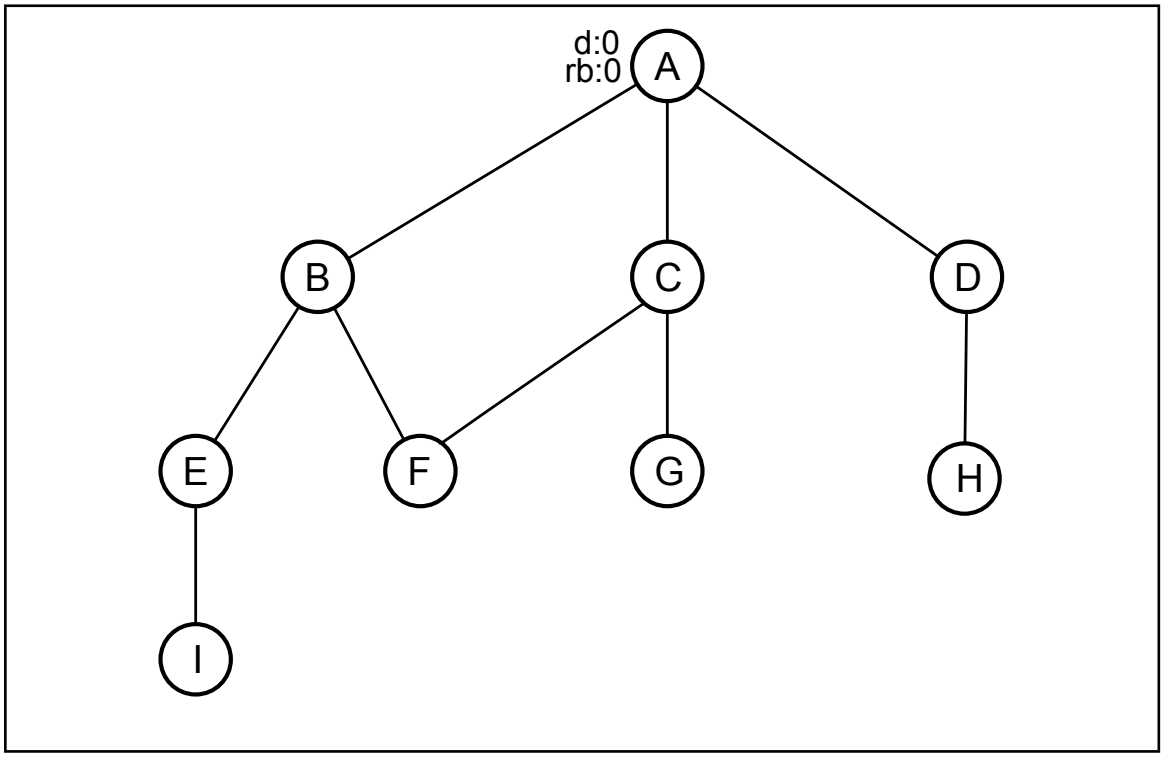
The Shortest Path Found:

(b) [8 marks] Articulation Points

Show how the recursive depth-first-search articulation points algorithm would find the articulation points in the following graph, assuming that the search starts with node A and considers neighbours of nodes in alphabetical order. Mark the following on the graph

- the depth ("d:") and the reach-back ("rb:") of each node (node A is done for you). If the reach-back is updated, give all the values e.g "rb: 3,1"
- the return values from each recursive call.
- the nodes that are articulation points, indicating why they were marked.

The algorithm is given on the facing page for your reference.



```
FindArticulationPoints (graph, start ):
    for each node: node.depth ← ∞, articulationPoints ← { }
    start .depth ← 0, numSubtrees ← 0
    for each neighbour of start
        if neighbour.depth = ∞ then
            RecursiveArtPts( neighbour, 1, start )
            numSubtrees ++
    if numSubtrees > 1 then add start to articulationPoints

RecursiveArtPts(node, depth, fromNode):
    node.depth ← depth, reachBack ← depth,
    for each neighbour of node other than fromNode
        if neighbour.depth < ∞ then
            reachBack ← min(neighbour.depth, reachBack)
        else
            childReach ← recArtPts(neighbour, depth +1, node)
            reachBack ← min(childReach, reachBack )
            if childReach ≥ depth then add node to articulationPoints
    return reachBack
```

(c) [10 marks] String Searching

The Knuth Morris Pratt algorithm searches for a string in a piece of text. It first analyses the string and builds a table, which enables KMP to search more efficiently than the naive string search algorithm.

Show the tables that the following KMP table building algorithm would construct for the two given strings: `"hello"` and `"abababaaafa"`.

```
computeKMPTable(string)
    initialise table to an array of integers
    table[0] ← −1;    table[1] ← 0;
    pos ← 2;          j ← 0;
    while pos < string.length
        if string[pos−1] = string[j]
            table[pos] ← j+1
            pos++
            j++
        else if j > 0
            j ← table[j]
        else
            table[pos] ← 0
            pos++
    return table
```

0	1	2	3	4
h	e	l	l	o
-1	0			

0	1	2	3	4	5	6	7	8
a	b	a	b	a	a	a	f	a
-1	0							

Question 2. Graphics [15 marks]

(a) [2 marks]

Suppose the light is located at the same place as the virtual camera, so it is exactly on the z (viewing) axis.

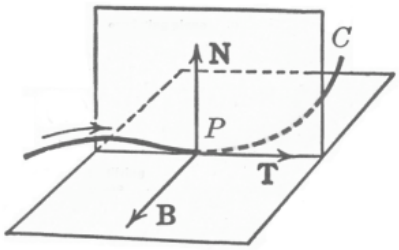
Now call L the vector from the origin to the light, and call V the vector from the origin to the camera. The dot product of these two vectors will be:

a) positive
b) negative
c) zero

Write the letter corresponding to the correct answer:

(b) [3 marks]

Imagine that you are writing a graphics program that controls the orientation of an airplane as it moves along a curve C in 3D space. The plane is at a particular point P, and at this point it is moving in the direction T (this is the tangent vector to the curve). You are given the vector N, which points in the direction of curvature.



You want to find the vector B that will determine the *banking* of the plane, i.e. the rotation about the tangent vector. This vector should be perpendicular to N and T. How can you compute this vector?

(c) [3 marks] Suppose your rendering program has one directional light, L, and an ambient light. Both are white.

Consider the following shading code:

```
public Color computeShade(Vector3D L, float ambient){
    float cosAngle = normal.cosTheta(L);
    float lightamount = ambient + cosAngle;
    int red = Math.max(0, Math.min(255,(int)( reflectivity .getRed()*lightamount)));
    int green = Math.max(0, Math.min(255,(int)( reflectivity .getGreen()*lightamount)));
    int blue = Math.max(0, Math.min(255,(int)( reflectivity .getBlue()*lightamount)));
    shade = new Color(red, green, blue);
    return shade;
}
```

There is something wrong with the way that `cosAngle` is used. What is wrong? (Assume that the normal vector has been calculated correctly)

(d) [2 marks] What operation or operations are represented by the following matrix?

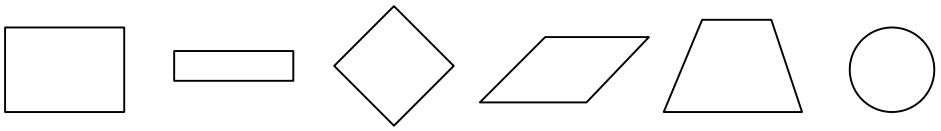
$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

a) scaling and translation
b) rotation by 2 degrees about x, and 1 degree each about the y and z axes, followed by translation by 5 units in y.
c) neither of the above.

Write the letter corresponding to the correct answer:

(e) [2 marks]

Imagine a cube that is viewed with the camera looking perpendicular to one of its faces, so it looks like a square. The camera uses orthogonal projection, as was done in the assignment, i.e. there are no perspective effects. If this cube is transformed by a 4x4 matrix, which of the following shapes *cannot* be the result?



Write the letters (a-f) corresponding to the correct answer(s):

(f) [3 marks]

Consider using the rendering program you wrote to render a very thin object, such as a pole, a wire, or a needle. Imagine that it viewed from a distance such that it is many pixels in length, but less than one pixel in width. Now imagine rotating the object about the viewing (z) axis, while keeping its distance from the camera fixed.

Explain why the width of the object will appear to change at different rotations:

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

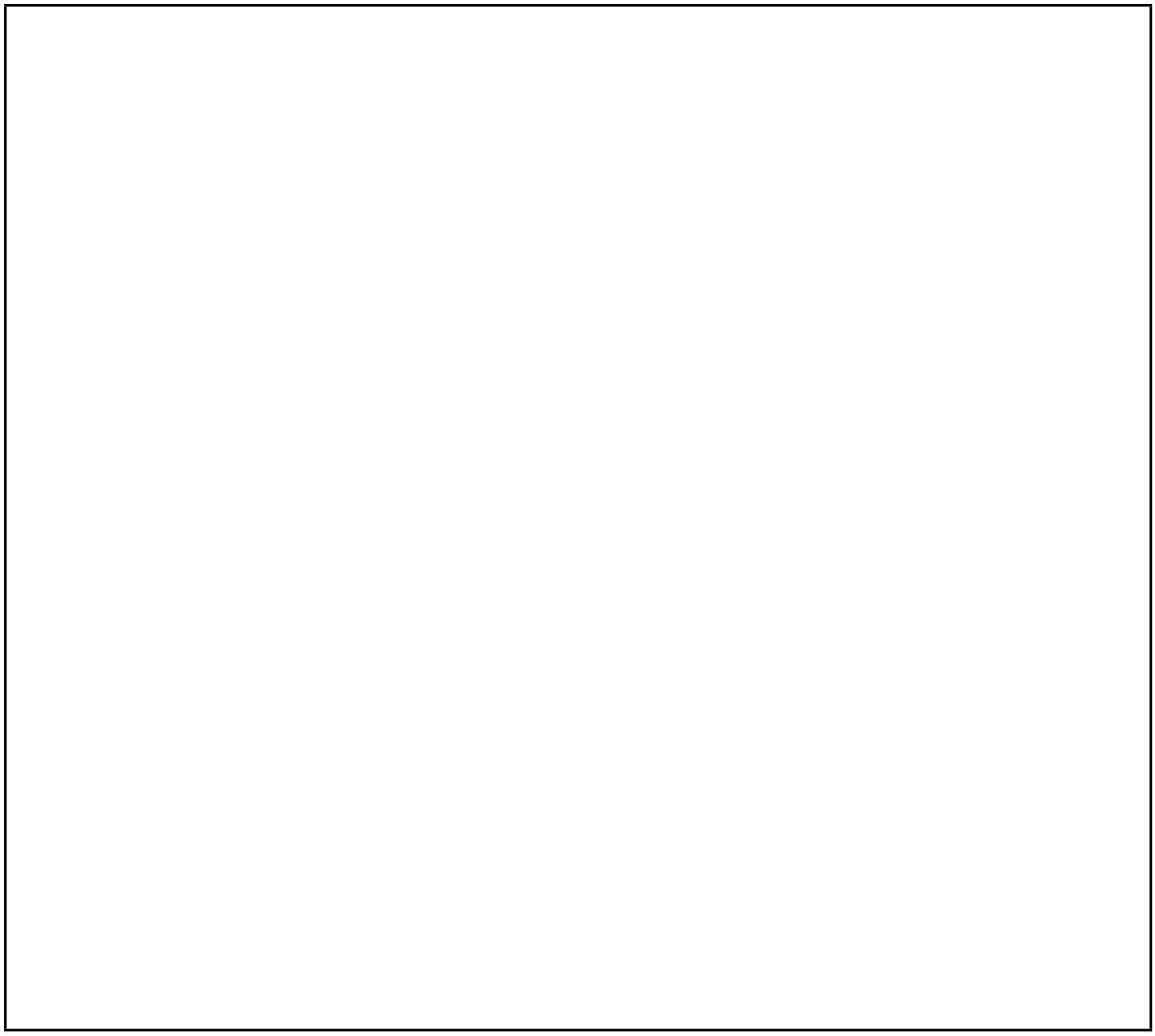
Question 3. Parsing [30 marks]

Consider a specialised query language for data analysis of claims data in an insurance company:

```
QUERY ::= "select" KIND "claims where" DETAILS
DETAILS ::= KEY "is" DATA [ ["and" | "or"] KEY "is" DATA ]*
KIND ::= "new" | "open" | "closed"
KEY ::= "ID" | "Name"
DATA ::= 4DIGITID | STRING
STRING ::= [A-Za-z]+[A_Za-z0-9]*
4DIGITID ::= [0-9][0-9][0-9][0-9]
```

(a) [5 marks] Give a concrete syntax tree for the following script

```
select open claims where
  ID is 1234 or Name is Alex and Name is Ambiguous
```



(Question 3 continued on next page)

(Question 3 continued)

(b) [5 marks] Give an abstract syntax tree for the following (different) script, and give a brief justification for what you included and what you ignored in the abstract syntax tree.

```
select new claims where
  Name is Alex and Name is Ambiguous and ID is 5678
```

Justification:

(Question 3 continued on next page)

(Question 3 continued)

(c) [10 marks] Suppose you are writing a recursive descent parser for the query language above. Give *the names of the classes* and for each class *list the necessary fields* for each Node that your AST will use (make them all implement interface `Node` given below).

```
interface Node {}
```

// Use spare page is required .

(Question 3 continued on next page)

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

(Question 3 continued)

(d) [10 marks] Now implement the `parseDetails` method from the grammar above.

```
private Node parseDetails(Scanner s){




}
```

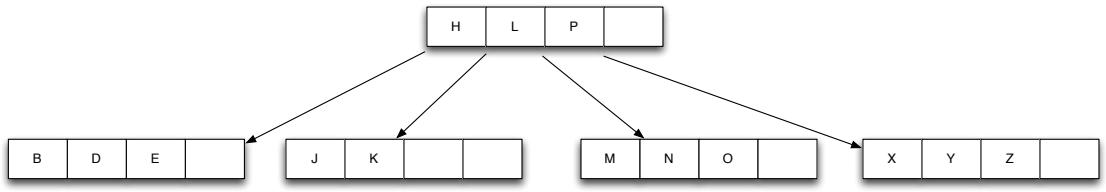
(Question 3 continued on next page)

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

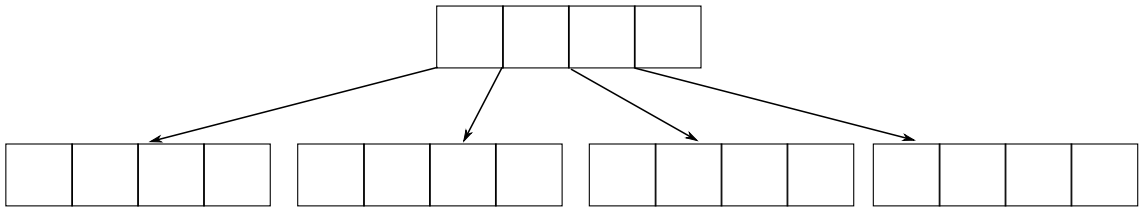
Question 4. B and B+ Trees [25 marks]

(a) [10 marks] Consider the *B*-tree of order 5 illustrated below.



Update the *B*-tree by successively deleting the key values Z, M, X, L. In your answer, show the *B*-tree after each deletion and briefly describe what you have done.
Note, the empty trees below are to save you time; you may modify their structure if you choose.

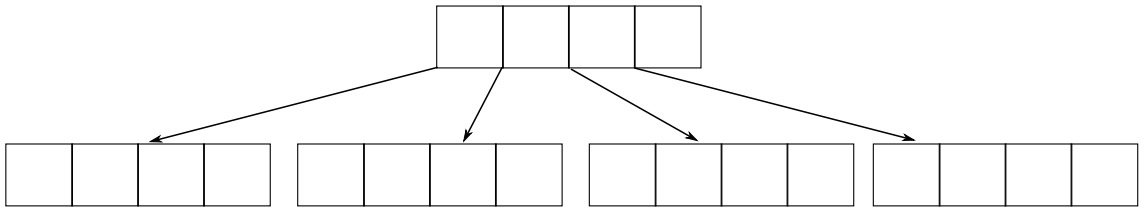
The *B*-tree after deleting key value Z:



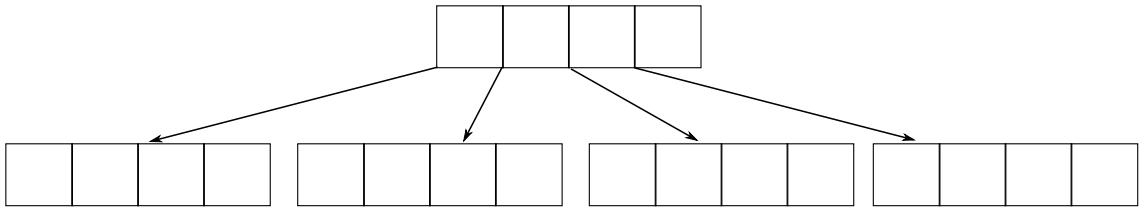
(Question 4 continued on next page)

(Question 4 continued)

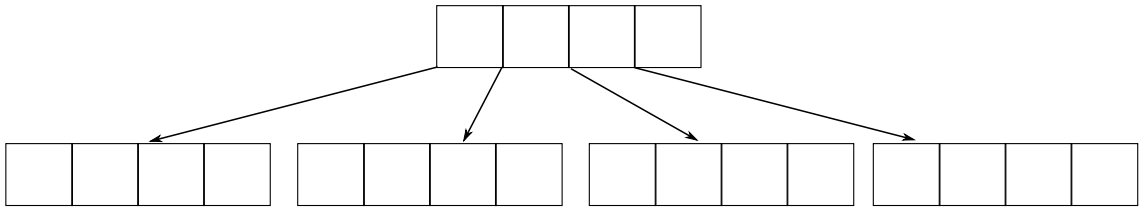
The *B*-tree after deleting key values Z and M:



The *B*-tree after deleting key values Z, M, and X:



The *B*-tree after deleting key values Z, M, X, and L:



(Question 4 continued on next page)

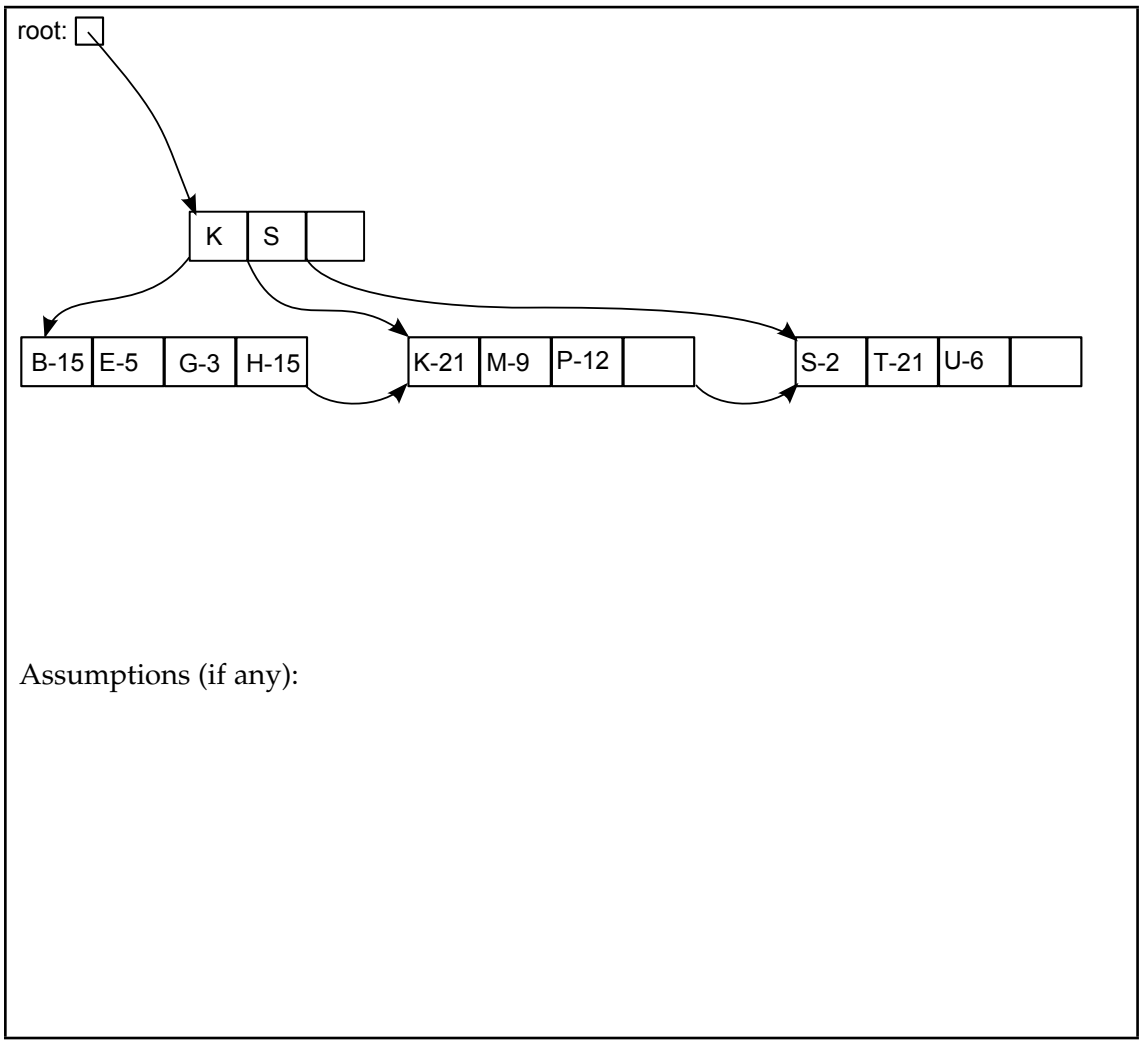
(Question 4 continued)

The following subquestions concern a B+ tree that has internal nodes holding up to 3 keys, and leaf nodes holding up to 4 key-value pairs. The keys are letters; the values are numbers.

(b) [10 marks] Show how the B+ tree below would be changed if the following key-value pairs were added to it:

```
Z-5  A-18  Y-2  N-34
```

State any assumptions you are making.



Assumptions (if any):

(Question 4 continued on next page)

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

(Question 4 continued)

(c) [5 marks] Suppose you are implementing a B+ tree in a file, storing each node of the tree in one block. The keys of the B+ tree are share market codes (for example "AAPL"), and the values are company names. Each share market code is up to 10 characters long, and the company names are limited to 100 characters. The blocks are 1024 bytes long.

How many key-value pairs can be stored in a leaf node? Explain your reasoning, show your working, and state any assumptions you make about the information stored in the blocks.

Question 5. Maximum Flow [10 marks]

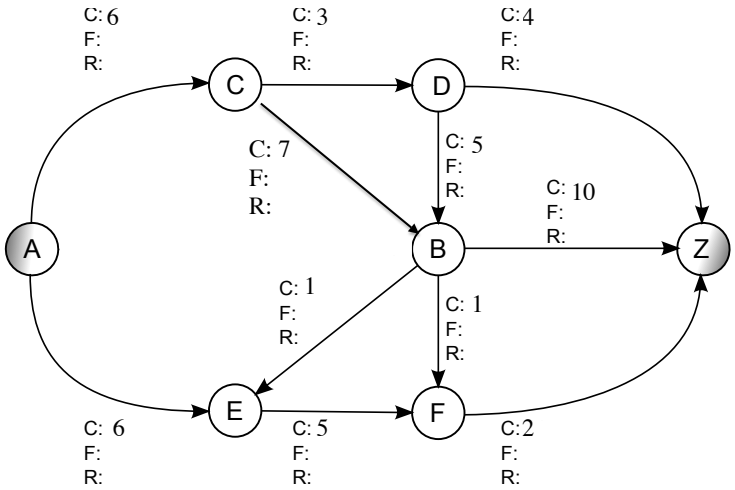
(a) [10 marks] Show how the Edmonds-Karp algorithm for Maximum Flow would find the maximum flow through the graph shown on the next page from the node A to the node Z. Each edge is labeled with its capacity, its flow, and its remaining capacity.

1. Show how the flow and remaining capacity change during the algorithm.
2. Below the graph, show the path found in each iteration, along with the flow that can be added along that path.
3. Show the maximum flow from A to Z found by the algorithm.

Hint: Remember that Edmonds-Karp repeatedly uses breadth first search to find a path from source to sink in which every edge has non-zero remaining capacity, sets the new flow to the minimum remaining capacity along the path, and adds this flow to the flow on each edge of the path (and subtracts from each of the reverse edges).

(Question 5 continued on next page)

(Question 5 continued)



Path	Flow added along path

Maximum Flow =

Question 6. Compression [15 marks]

(a) [5 marks] What is the key idea behind *any* lossless compression algorithm? Please explain using example.

(b) [5 marks] Construct a trie that can be used to encode the following sentence using Huffman Encoding. (Note: label the links of the Trie, not the node.)

```
WHAT IF WHAT IF WHY NOT WHY NOT
```

(Question 6 continued on next page)

(Question 6 continued)

(c) [5 marks] Show the resulting encoding and state how many bits will be required to store the message from the previous question:

```
WHAT IF WHAT IF WHY NOT WHY NOT
```

(Question 6 continued on next page)

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

* * * * * * * * * * * * * * *