Multimedia IGDS MSc Exam 2000 SOLUTIONS

Setter: ADM
Checker: ACJ

Answer 3 Questions out of 4

Time Allowed 2 Hours

1. (a) Why is data compression, including file compression, highly desirable for Multimedia activities?

Multimedia files are very large therefore for storage, file transfer etc. file sizes need to be reduced. Text and other files may also be encoded/compressed for email and other applications.

2 MARKS --- BOOKWORK

(b) *Briefly explain, clearly identifying the differences between them, how entropy coding and transform coding techniques work for data compression. Illustrate your answer with a simple example of each type.*

Compression can be categorised in two broad ways:

Lossless Compression
-- where data is compressed and can be reconstituted (uncompressed) without loss of detail or information. These are referred to as bit-preserving or reversible compression systems also.
Lossy Compression
-- where the aim is to obtain the best possible fidelity for a given bit-rate or minimizing the bit-rate to achieve a given fidelity measure. Video and audio compression techniques are most suited to this form of compression.

Lossless compression frequently involves some form of entropy encoding and are based in information theoretic techniques

Lossy compression use source encoding techniques that may involve transform encoding, differential encoding or vector quantisation.

ENTROPY METHODS:

The entropy of an information source S is defined as:

$H(S) = SUM_I (P_I \, Log_2 \, (1/P_I)$

where P_I is the probability that symbol S_I in S will occur.

$Log_2 \, (1/P_I)$ indicates the amount of information contained in S_I, i.e., the number of bits needed to code S_I.

Encoding for the Shannon-Fano Algorithm:

A top-down approach

1. Sort symbols according to their frequencies/probabilities, e.g., ABCDE.

3 Recursively divide into two parts, each with approx. same number of counts.

(Huffman algorithm also valid indicated below)

A simple transform coding example

A Simple Transform Encoding procedure maybe described by the following steps for a 2x2 block of monochrome pixels:

1. Take top left pixel as the base value for the block, pixel A.
2. Calculate three other transformed values by taking the difference between these (respective) pixels and pixel A, i.e. B-A, C-A, D-A.
3. Store the base pixel and the differences as the values of the transform.

Given the above we can easily for the forward transform:

and the inverse transform is trivial

The above transform scheme may be used to compress data by exploiting redundancy in the data:

Any Redundancy in the data has been transformed to values, Xi. So We can compress the data by using fewer bits to represent the differences. I.e if we
use 8 bits per pixel then the 2x2 block uses 32 bits/ If we keep 8 bits for the base pixel, X0, and assign 4 bits for each difference then we only use 20 bits.
Which is better than an average 5 bits/pixel

7 MARKS --- BOOKWORK

(c) (*i) Show how you would use Huffman coding to encode the following set of tokens:*

BABACACADADABBCBABEBEDDABEEEBB

How is this message transmitted when encoded?

The Huffman algorithm is now briefly summarised:

1. Initialization: Put all nodes in an OPEN list, keep it sorted at all times (e.g., ABCDE).

2. Repeat until the OPEN list has only one node left:

(a) From OPEN pick two nodes having the lowest frequencies/probabilities, create a parent node of them.

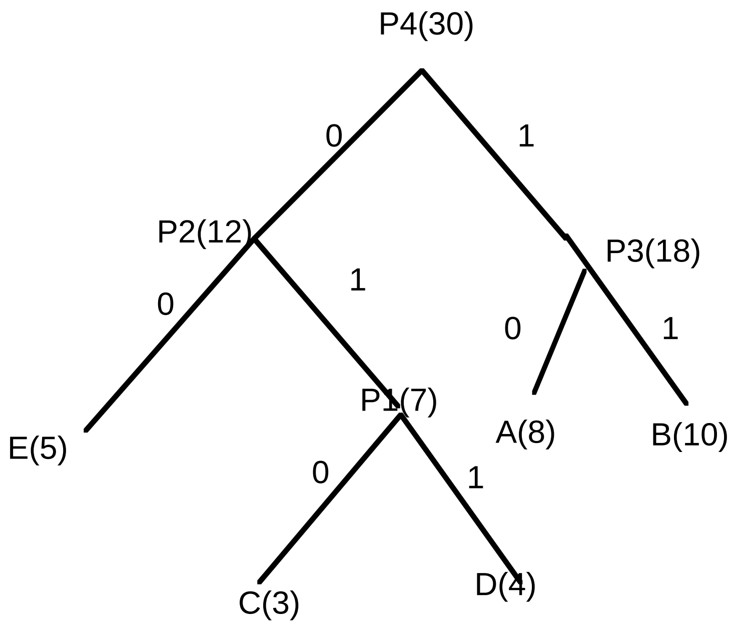
(b) Assign the sum of the children's frequencies/probabilities to the parent node and insert it into OPEN.

(c) Assign code 0, 1 to the two branches of the tree, and delete the children from OPEN.

Symbol	Count	OPEN (1)	OPEN (2)	OPEN (3)
A	8			18
B	10			-
C	3	7	12	
D	4	-		
E	5		-	
Total	30			

8 indicate merge node with other node with number in column

Finished Huffman Tree:



Symbol	Code
A	10
B	11
C	010
D	011
E	00

How is this message transmitted when encoded?

Send code book and then bit code for each symbol.

7 Marks --- UNSEEN

(ii) How many bits are needed transfer this coded message and what is its Entropy?

Symbol	Count	Subtotal # of bits
A	8	16
B	10	20
C	3	9
D	4	12
E	5	10

Total Number bits (excluding code book) = 62

Entropy = 62/30 = 2.06667

8 MARKS --- UNSEEN

(iii) What amendments are required to this coding technique if data is generated live or is otherwise not wholly available? Show how you could use this modified scheme by appending the tokens ADADA to the end of the above message.

Adaptive method needed:

Basic idea (encoding)

```
Initialize_model();
 while ((c = getc (input)) != eof)
   {
     encode (c, output);
     update_model (c);
   }
```

So encode message as before:

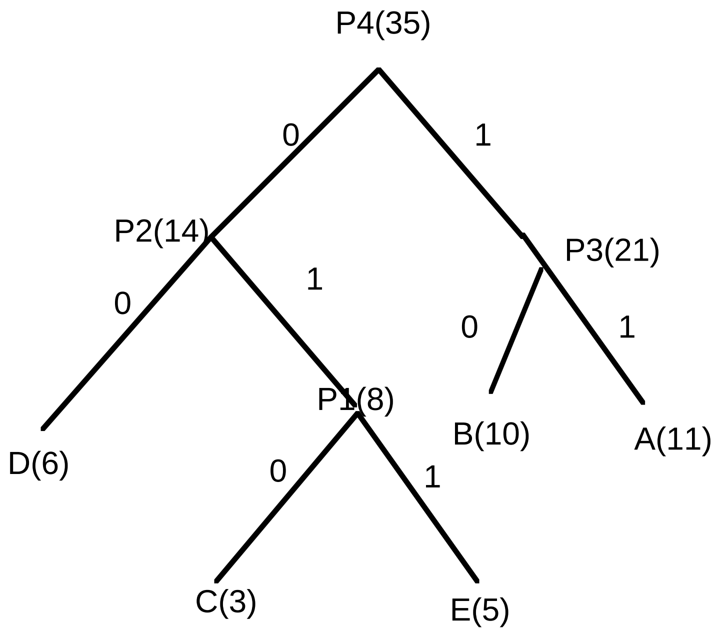
A= 01 D = 0000

So addd stream:

01000001000001

Modify Tree:

Symbol	Count	OPEN (1)	OPEN (2)	OPEN (3)
A	11			21
B	10			-
C	3	8	14	
D	6		-	
E	5	-		



5 Marks --- UNSEEN

2 (a) Give a definition of a Multimedia Authoring System. What key features should such a system provide?

An Authoring System is a program which has pre-programmed elements for the development of interactive multimedia software titles.
Authoring systems vary widely in orientation, capabilities, and learning curve.

There is no such thing (at this time) as a completely point-and-click automated authoring system; some knowledge of heuristic thinking and algorithm design is necessary.

Authoring is basically just a speeded-up form of programming --- VISUAL PROGRAMMING; you don't need to know the intricacies of a programming language, or worse, an API, but you do need to understand how programs work.

2 MARKS ---- BOOKWORK

(b) *What Multimedia Authoring paradigms exist? Describe each paradigm briefly.*

There are various paradigms, including:

Scripting Language

The Scripting paradigm is the authoring method closest in form to traditional programming. The paradigm is that of a programming language, which specifies (by filename) multimedia elements, sequencing, hotspots, synchronization, etc. A powerful, object-oriented scripting language is usually the centerpiece of such a system; in-program editing of elements (still graphics, video, audio, etc.) tends to be minimal or non-existent. Scripting languages do vary; check out how much the language is object-based or object-oriented. The scripting paradigm tends to be longer in development time (it takes longer to code an individual interaction), but generally more powerful interactivity is possible. Since most Scripting languages are interpreted, instead of compiled, the runtime speed gains over other authoring methods are minimal.

The media handling can vary widely; check out your system with your contributing package formats carefully. The Apple's HyperTalk for HyperCard, Assymetrix's

OpenScript for ToolBook and Lingo scripting language of Macromedia Director are examples of a Multimedia scripting language.

Here is an example lingo script to jump to a frame

```
glob
al
gNav
```

```
                                                    Spri
                                                    te
on exitFrame
go the frame
play sprite gNavSprite
end
```

Iconic/Flow Control

This tends to be the speediest (in development time) authoring style; it is best suited for rapid prototyping and short-development time projects. Many of these tools are also optimized for developing Computer-Based Training (CBT). The core of the paradigm is the Icon Palette, containing the possible functions/interactions of a program, and the Flow Line, which shows the actual links between the icons. These programs tend to be the slowest runtimes, because each interaction carries with it all of its possible permutations; the higher end packages, such as Authorware or IconAuthor, are extremely powerful and suffer least from runtime speed problems.

Frame

The Frame paradigm is similar to the Iconic/Flow Control paradigm in that it usually incorporates an icon palette; however, the links drawn between icons are conceptual and do not always represent the actual flow of the program. This is a very fast development system, but requires a good auto-debugging function, as it is visually un-debuggable. The best of these have bundled compiled-language scripting, such as Quest (whose scripting language is C) or Apple Media Kit.

Card/Scripting

The Card/Scripting paradigm provides a great deal of power (via the incorporated scripting language) but suffers from the index-card structure. It is excellently suitedfor Hypertext applications, and supremely suited for navigation intensive (a la Cyan's "MYST" game) applications. Such programs are easily extensible via XCMDs andDLLs; they are widely used for shareware applications. The best applications allow all objects (including individual graphic elements) to be scripted; many entertainment applications are prototyped in a card/scripting system prior to compiled-language coding.

Cast/Score/Scripting

The Cast/Score/Scripting paradigm uses a music score as its primary authoring metaphor; the synchronous elements are shown in various horizontal tracks withsimultaneity shown via the vertical columns. The true power of this metaphor lies in the ability to script the behavior of each of the cast members. The most popularmember of this paradigm is Director, which is used in the creation of many commercial applications. These programs are best suited for animation-intensive orsynchronized media applications; they are easily extensible to handle other functions (such as hypertext) via XOBJs, XCMDs, and DLLs.

Macromedia Director uses this .

<u>Hierarchical Object</u>

The Hierarchical Object paradigm uses a object metaphor (like OOP) which is visually represented by embedded objects and iconic properties. Although the learning curve is non-trivial, the visual representation of objects can make very complicated constructions possible.

<u>Hypermedia Linkage</u>

The Hypermedia Linkage paradigm is similar to the Frame paradigm in that it shows conceptual links between elements; however, it lacks the Frame paradigm’s visual linkage metaphor.

<u>Tagging</u>

The Tagging paradigm uses tags in text files (for instance, SGML/HTML, SMIL (Synchronised Media Integration Language), VRML, 3DML and WinHelp) to link pages, provide interactivity and integrate multimedia elements.

8 Marks --- BOOKWORK

(c) You have been asked to provide a Multimedia presentation that can support media in both English and French. You have been given a sequence of 10 images and a single 50 second digitised audio soundtrack in both languages. Each Image should be mapped over consecutive 5 second fragments of the audio. All Images are of the same 500x500 pixel dimension.

Describe, giving suitable code fragments, how you would assemble such a presentation using SMIL. Your solution should cover all aspects of the SMIL presentation

```
<smil>
            <head>
             <layout>
              <root-layout height="500" width="500" background-
color="#000000" title="MultiLingual"/>
              <region id="image1" width="500" height="500" top="0"
left="0" background-color="#000000" z-index="1" />
              <region id="image2" width="500" height="500" top="0"
left="0" background-color="#000000" z-index="1" />
……….
            </layout>
             </head>

<body>
      <par>
             <switch>
```

```
                <!-- English only -->
                 < audio system-language="en" src ="english.au" />

                    <!-- French only -->
                    <audio system-language="fr" src ="francais.au" />
                </switch>

<seq>
                  <img src="image1.jpg" region="image1" begin="0.00s"
dur="5.00s" />
                  <img src="image2.jpg" region="image2" begin="5.00s"
dur="5.00s" />

…….

</seq>
</par>

</body>
</smil>
```

14 Marks ---- UNSEEN

3. (a) *What is meant by the Quality of Service of a multimedia application?*

Quality of Service (Qos) DEFN:

- this is basically a collection of parameters that relate to a sequence as seen at the source and as seen at the destination of a multimedia presentation.
-

2 Marks --- BOOKWORK

(b) *What major factors affect the Quality of Service of a multimedia application?*

The QoS measure probably is the ultimate measure of a multimedia system. Four essential parameters are:

- Bandwidth -- capacity of the transfer mechanism between source and destination.
- Delay -- the time a multimedia unit spends in tranmission from sourse to destination.
- Delay Jitter -- Variation in delay delivery of data
- Loss Probability -- the ratio of units of information that an application can afford to lose.

- Hardware problems:
  - Network Connection
  - Traffic Analysis --- scheduling, buffer design, congestion control and synchronisation

8 Marks -- BOOKWORK

(c) *Perform a CORR analysis on the following Traffic scheduling problem:*

A scheduler has an allocation cycle time of length 5 seconds and is serving 3 connections. The connections have rates of 2.5, 1.5 and 1 cells per cycle respectively. You should assume that all three connections are initially backlogged. What is the ideal delivery rate where fractional slots can be allocated? How should the connections be scheduled using CORR?

BASIC CORR ALGORITHM:

time-line divide and allocate policy moderated by a queuing schedule. Distributes excess bandwidth over active connections.

- The Maximum length of allocation cycle is T
- Max. No. of cells transmitted in on cycle is T -- assuming cell transmission time is unit of time
- At time of admission, each connetion Ci is allocated a Rate Ri ---
- Ri s real numbers.
- *Scheduling Goal*: To allocate each connection Ciclose to Ri
- slots in each cycle.

- Excatly Ri slots of a long time frame

- *3 Event CORR Scheduler*: Asynchronous ~~~
    - Initilaize --- invoked for each new admitted connection
        - add connection to list Ci .
        - Sort Ci in decreasing Ri .
    - Enqueue--- Activate on arrival of a packet
        - Put packet in appropriate connection queue
        - update cell count of connection

    - Dispatch--- invoked at begining of busy periods
        - Ensures that connections cannot swamp othersa

CORR Algorithm:

For each connetion Ci scheduler:

- maintains a separate queue, ni

- keeps count of the waiting cells
- holds the number of current slots credited, ri .

Slot allocation divided into 2 cycles:

- Major Cycle --- integral requrement of each connection satisfied first
    - Slots left over assigned to minor cycle.
- Minor Cycle --- Elligible connections assigned allocated a full slot each in minor cycle (can't have fraction slot allocations!!)
-

At the begining of a busy period ({\em Dispatch})

- All ri 's set to zero
- New cycle started for each new cycle,

1. current number of unallocated slots T is initialised to T.
2. For each major cycle, cycle through each Ci update ri to ri + Ri

- If number of cells queued, ni < ri set ri to n
- Dispatch minumum of T and ri cells from C_{i}

3. A minor cycle starts with slots left over from preceding major cycle

- `cylce through connection list, if slots are left dispatch connection

- IF AND ONLY IF (a) packets queued or (b) ri > 0
- if t =0 or no connection end cycle

SOLUTION:

CYCLE LENGTH T = 5

RATES = CELLS PER CYCLE
R1 = 2.5, R2 = 1.5, R3 = 1

<u>FIRST MAJOR CYCLE:</u>

[R1] = 2, [R2] = 1, [R3] = 1

1ST MINOR Cycle:
r1 = 0.5, r2 = 0.5, r3 = 0

SLOT left over for C1
Update ri
r1 = -0.5, r2 = 0.5, r3 = 0

<u>Second MAJOR CYCLE:</u>

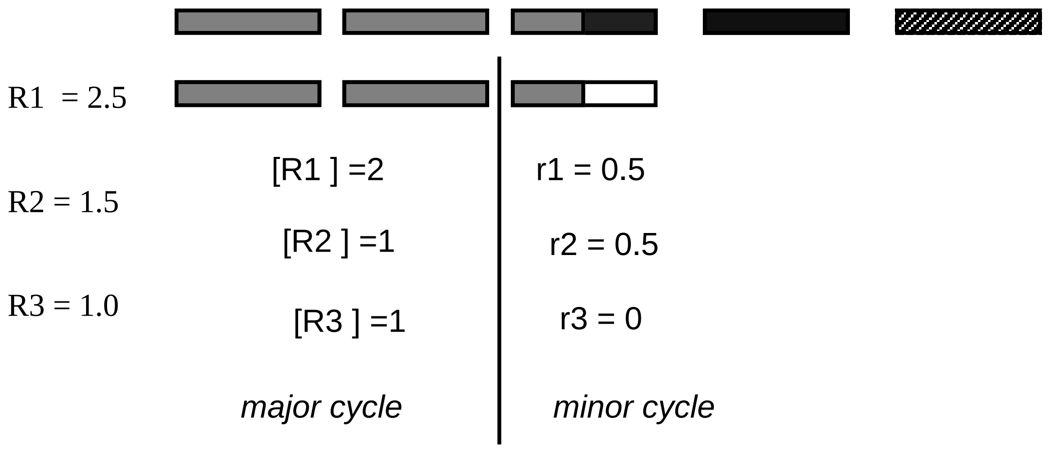
Update Ri = ri + Ri

So R1 = 2.0, R2 = 2.0, R3 = 1.0

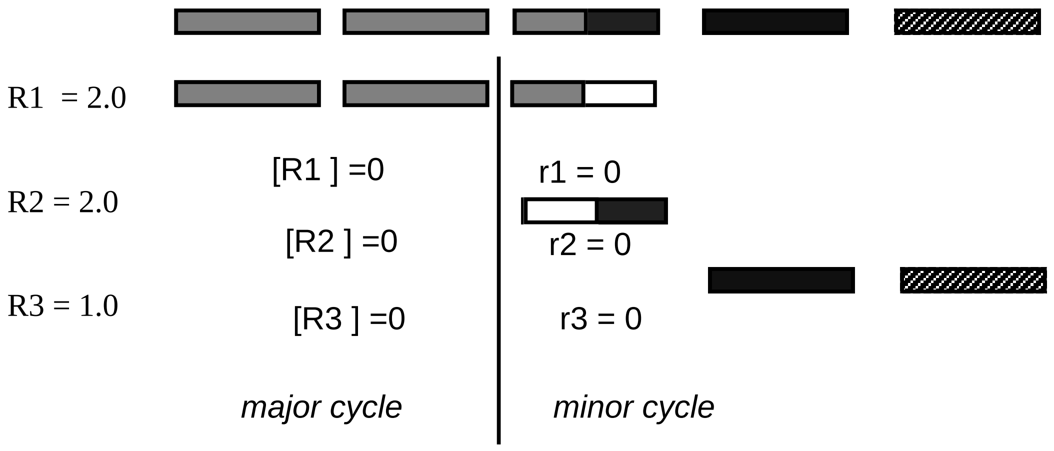
All integer so satisfy MAJOR CYCLE

Schedule is:

CYCLE 1



CYCLE 2



3 slots r1, 2 slots r2 1 slots R3 (as expected?)

14 Marks -- UNSEEN

4. (a) *What is MIDI? How is a basic MIDI message structured?*

MIDI: a protocol that enables computer, synthesizers, keyboards, and other musical or (even) multimedia devices to communicate with each other.

MIDI MESSAGE:

- MIDI message includes a status byte and up to two data bytes.
    - Status byte

      The most significant bit of status byte is set to 1.
    - The 4 low-order bits identify which channel it belongs to (four bits produce 16 possible channels).
    - The 3 remaining bits identify the message.
- The most significant bit of data byte is set to 0.

4 Marks --- Bookwork

(b) *In what ways can MIDI be used effectively in Multimedia Applications, as opposed to strictly musical applications*?

Many Application:

Low Bandwidth/(Low Quality?) Music on Web, Quicktime etc supports Midi musical instrument set
Sound Effectts --- Low Bandwidth alternative to audio samples, Sound Set part of GM soundset
Control of external devices --- e.g Synchronistaion of Video and Audio (SMPTE), Midi System Exclusive, AUDIO RECORDERS, SAMPLERS
Control of synthesis --- envelope control etc
MPEG 4 Compression control --- see Part (c)
Digital Audio

8 Marks --- Applied Bookwork: Discussion of Information mentioned in Notes/Lectures.

(c) How can MIDI be used with modern data compression techniques? Briefly describe how such compression techniques may be implemented?

- We have seen the need for compression already in Digital Audio -- Large Data Files

- Basic Ideas of compression (see next Chapter) used as integral part of audio format -- MP3, real audio etc.

- Mpeg-4 audio -- actually combines compression synthesis and midi to have a massive impact on compression.

- Midi, Synthesis encode what note to play and how to play it with a small number of parameters -- Much greater reduction than simply having some encoded bits of audio.

- Responsibility to create audio delegated to generation side.

MPEG-4 comprises of 6 Structured Audio tools are:

- SAOL the Structured Audio Orchestra Language
- SASL the Structured Audio Score Language
- SASBF the Structured Audio Sample Bank Format
- a set of MIDI semantics which describes how to control SAOL with MIDI
- a scheduler which describes how to take the above parts and create sound
- the AudioBIFS part of BIFS, which lets you make audio soundtracks in MPEG-4 using a variety of tools and effects-processing techniques

MIDI IS the control language for the synthesis part:

- As well as controlling synthesis with SASL scripts, it can be controlled with MIDI files and scores in MPEG-4. MIDI is today's most commonly used representation for music score data, and many sophisticated authoring tools (such as sequencers) work with MIDI.

- The MIDI syntax is external to the MPEG-4 Structured Audio standard; only references to the MIDI Manufacturers Association's definition in the standard. But in order to make the MIDI controls work right in the MPEG context, some semantics (what the instructions "mean") have been redefined in MPEG-4. The new semantics are carefully defined as part of the MPEG-4 specification.

  10 Marks --- Basic Ideas mentioned in lectures. Details application of Lecture notes material