AT THE EDGE

For Violoncello

by

Ali Riza SARAL

To Daria B.

Jan 2017

Copyright notice

Please send a digital recording of your performance
to the composer at arsaral(AATT)yahoo.com

The piece is written to be used for good causes.

CONTENTS

Acknowledgements

Foreword

Introduction

Analysis

Score

Sketches

Short Article

References

About the Composer

ACKNOWLEDGEMENTS

At the end of 2016 I received a letter from a cellist who I do not know well even today. She said that she and her German guitarist colleague have performed my recent work 'The Broken Olive Branch'.

I was astonished as I was expecting a news from Italy by that time. It is so pleasent to hear one of my works performed suddenly. I usually dedicate my work to the person who performs it. But 'The Broken Olive Branch' was requested from Italy first.

So I decided to write a new work for the courageous Russion lady from Germany:

Thank you Daria B.

FOREWORD

I travelled from Illinois to San Fransisco to write a computer generated music in 1990 Summer Course of Stanford University's CCRMA lab.

Piece for Computer and 4 Trombones
http://imslp.org/wiki/Piece_for_Computer_and_4_Trombones_(Saral,_Ali_Riza)

Its subtitle is 'Death on the Border'. It is so amazing that now I have similar possibilities on my table at my Home.

'At the Edge' is the situation of human being in the world of 2017. We are at the edge of enormous progress in science and technology, progress that we cannot even imagine.

The vast majority is only being swayed by the progresses that are happening and sometimes getting easily manipulated.

A moral and social response has to be developed with a special emphasis on education and equal opportunity.

I tried to give a sense of this human situation by the cello performer's unavoidable difficulty while playing 'At the Edge'.

INTRODUCTION

This piece is an open-ended self discussion of what can be done as a computer aided music work.

The computer generated parts are sometimes mixed with manually composed sections. Sometimes this mixture is done homogenuously, sometimes heterogenuously. This is achieved by manually changing/retouching computer generated parts.

The term 'Computer generated' should not mislead you. Computer generation is based on composition algorithms which are manually programmed so that monotenuous effects are avoided unless it is required to increase tension.

For example an array of 5 notes A, Gb, E, F, C# may be put into an array and a random number between 0 and 4 generated. This produces a real randomness effect. A note may be repeated a number of times by chance. This has to be changed, so that a pitch that has been played once may not be played for a couple of throws of the dice. This is the random repetition depth. An artificially beautiful randomness appears if repetition depth is increased. The depth decrease helps to increase tension where needed.

It is possible to imitate/approach human manual composition by deliberately changing and manipulating algorithms. For example, you can put artificial randomness in not only pitch but also rhytm values, etc.

It is also possible to imitate computer generation by hand but this effort is also limited by the sheer volume of the computer generation.

The computer generation may also be used just to produce the base line like a carpet veawer using a white layer of threads to tie the color nodes on.

Uniqueness and mutation are issues that has to be pondered upon in this approach to music composition.

ANALYSIS

Just a few words about the things I was faced with during the composition process of 'AT the Edge':

What is unique? What is mutated?

Something unique can be differed from others.
Something unique has a unity.
Something unique has an identity.

A truly unique thing is something that can not be mutated.
A truly unique thing exists on its own, with nothing similar to or copied from it.

On the other hand, a mutation is not unique in the presence of an original.

By definition, sound/music is an episodic event queue of frequency, duration.

A melody becomes unique when it is heard for the first time provided that it has a unity. If a change to the melody mutates the melody but does not effect its percieved identity it is a non-destructive mutation.

Randomly generated mutations may be limited not to cause (or to cause) destruction of the unique melody.

What happens in the case of a random or limited alleatory melody? Notes that are not generated according to the fundamental algorithm may destruct the limited alleatory melody depending on a definite condition. This condition may change according to the character of the original limited alleatory melody.

A limited alleatory melody may be mixed with another melody. For ex. introduction of constant pitches with constantly seperated instances, may give a unique perception of an atomic character. Mixing of limited alleatory with a unique melody breaks the randomness, rather balances its effect.

SCORE

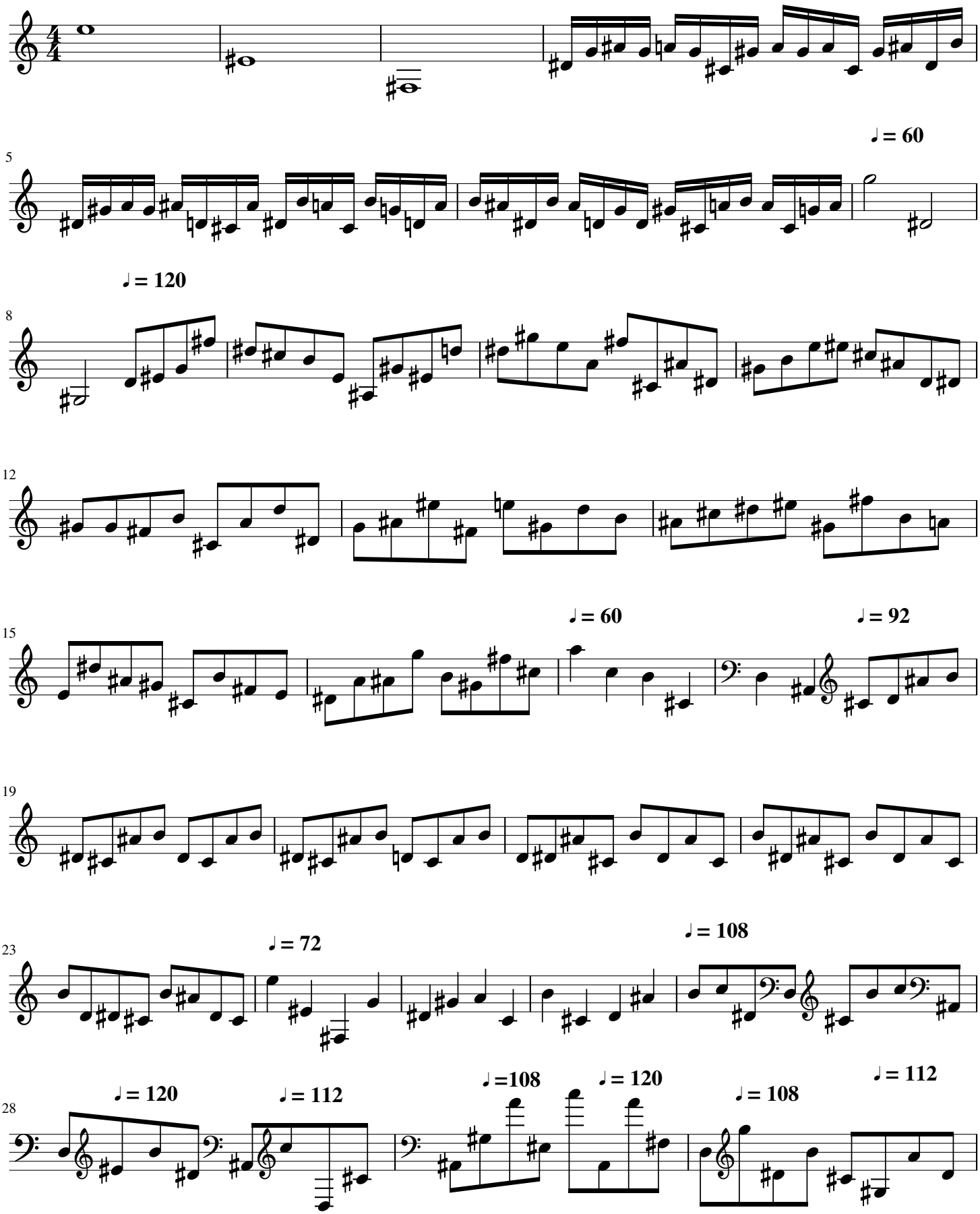
At the Edge

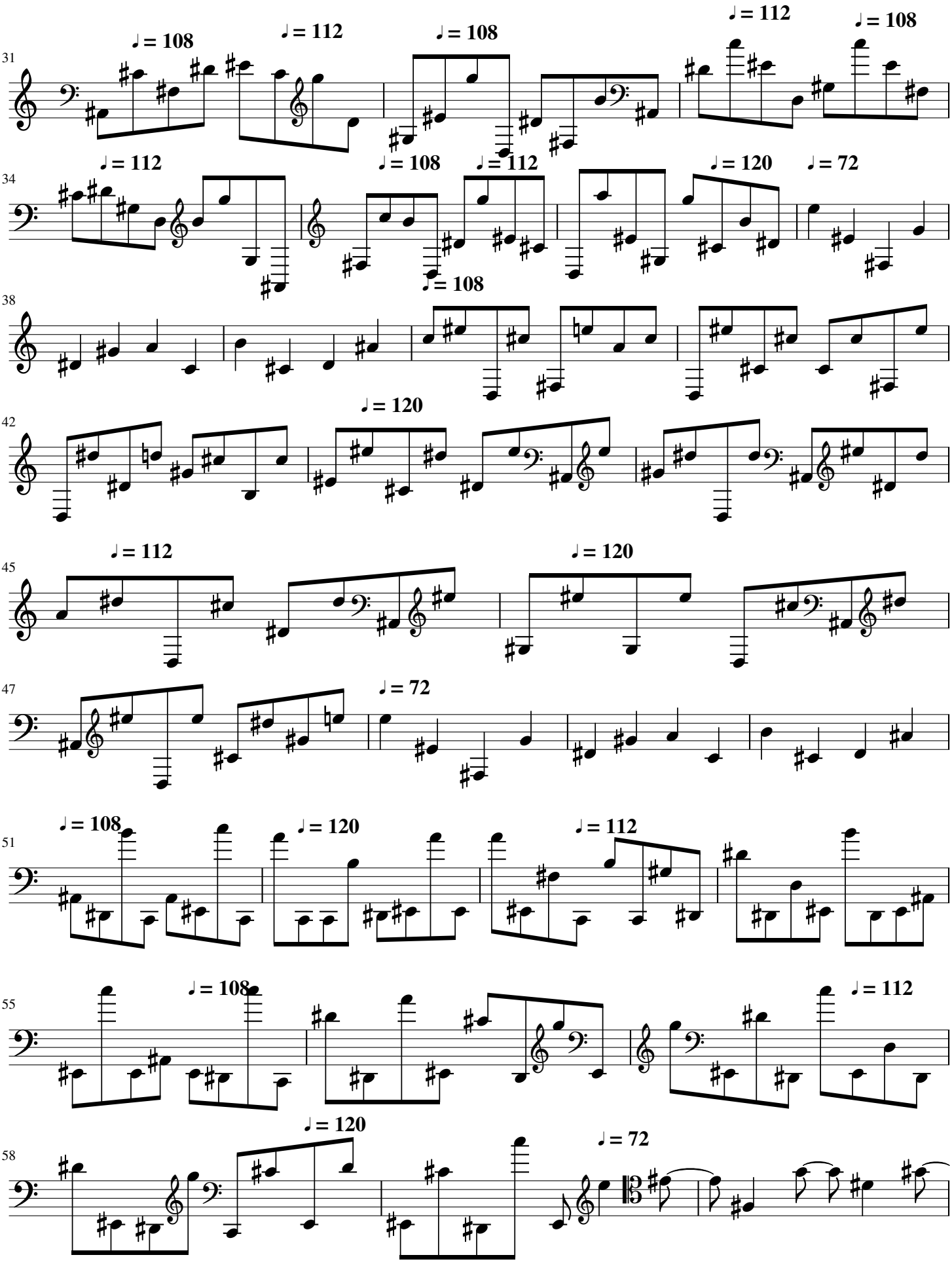
1- Bagatelle

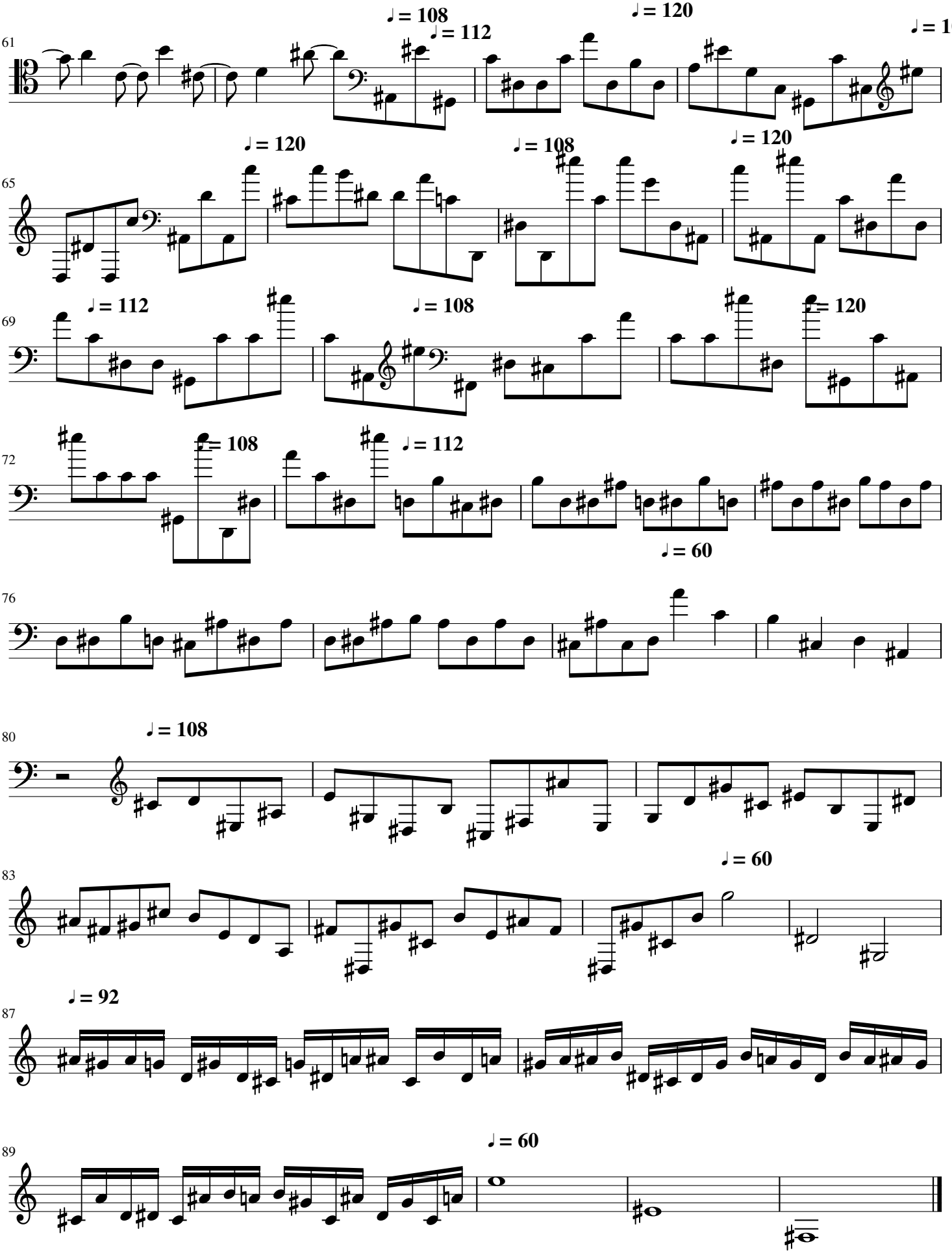
Ali Riza SARAL

Larghetto ♩ = 60

Andante ♩ = 80




31
♩ = 108
♩ = 112
♩ = 108
♩ = 112
♩ = 108
34
♩ = 112
♩ = 108
♩ = 112
♩ = 120
♩ = 72
38
♩ = 108
42
♩ = 120
45
♩ = 112
♩ = 120
47
♩ = 72
51
♩ = 108
♩ = 120
♩ = 112
55
♩ = 108
♩ = 112
58
♩ = 120
♩ = 72

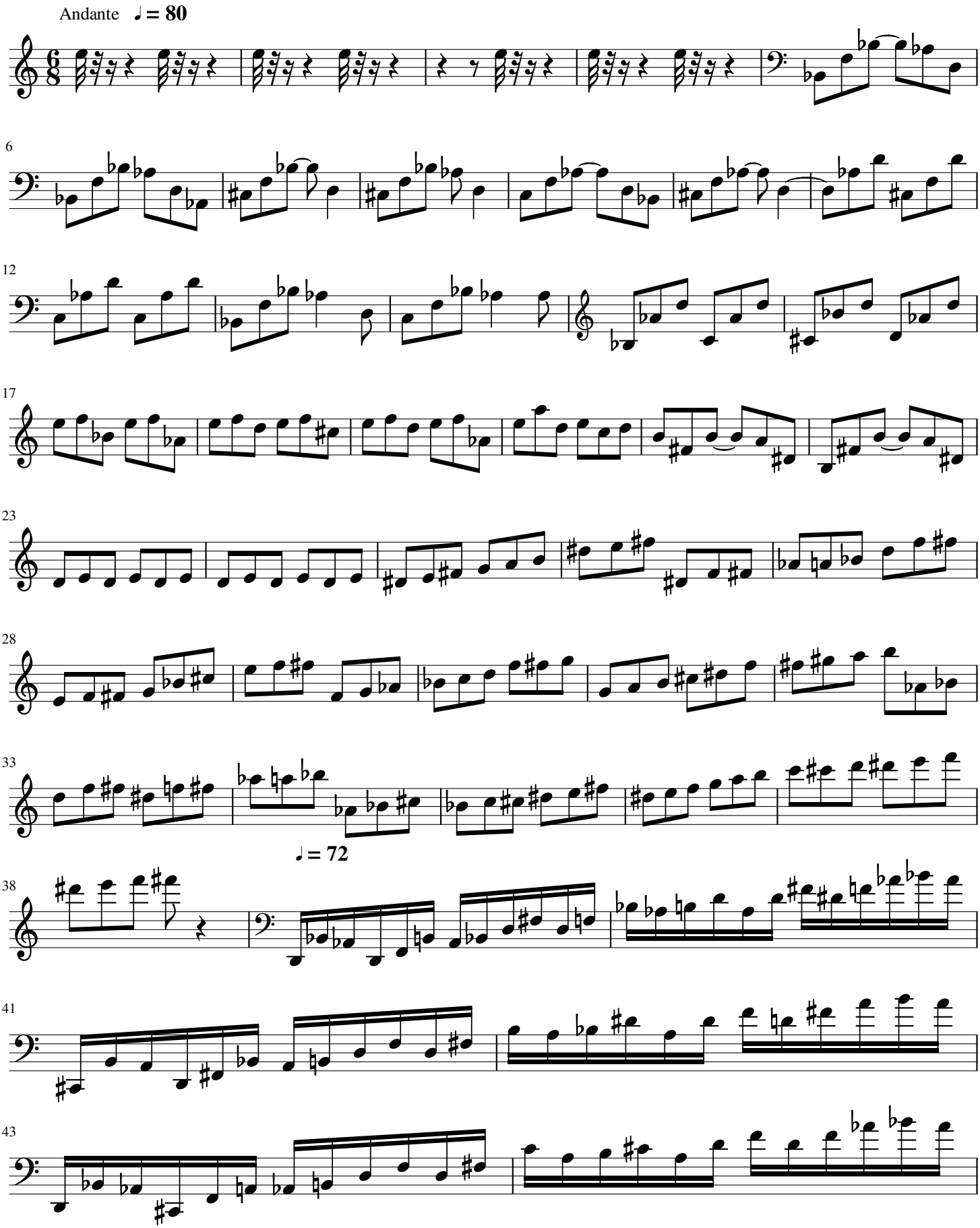

♩ = 108
♩ = 112
♩ = 120
♩ = 108
61
♩ = 120
♩ = 108
♩ = 120
65
♩ = 112
♩ = 108
♩ = 120
69
♩ = 108
♩ = 112
72
♩ = 60
76
♩ = 108
80
♩ = 60
83
♩ = 92
87
♩ = 60
89

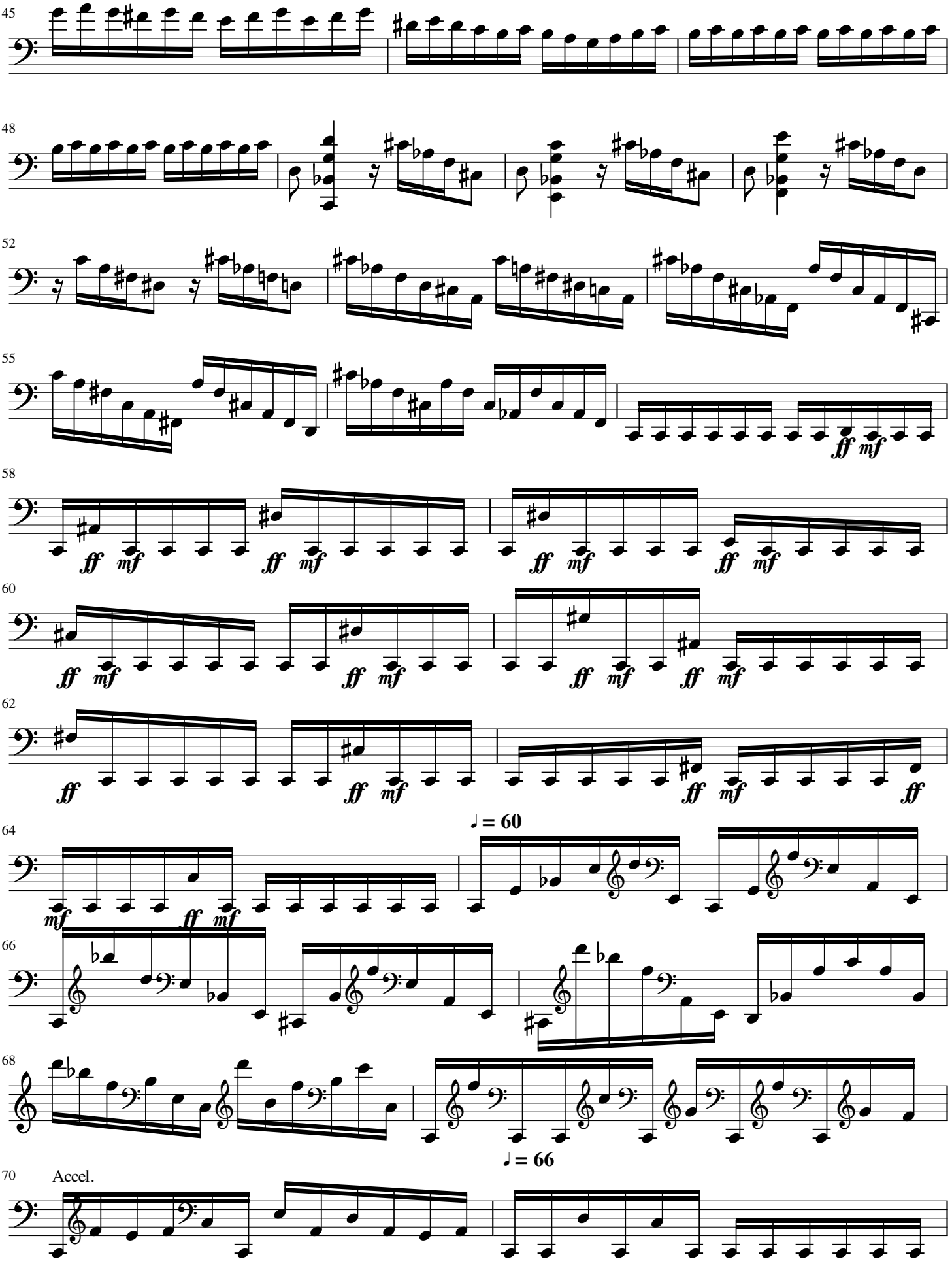
Duree:3' 35"

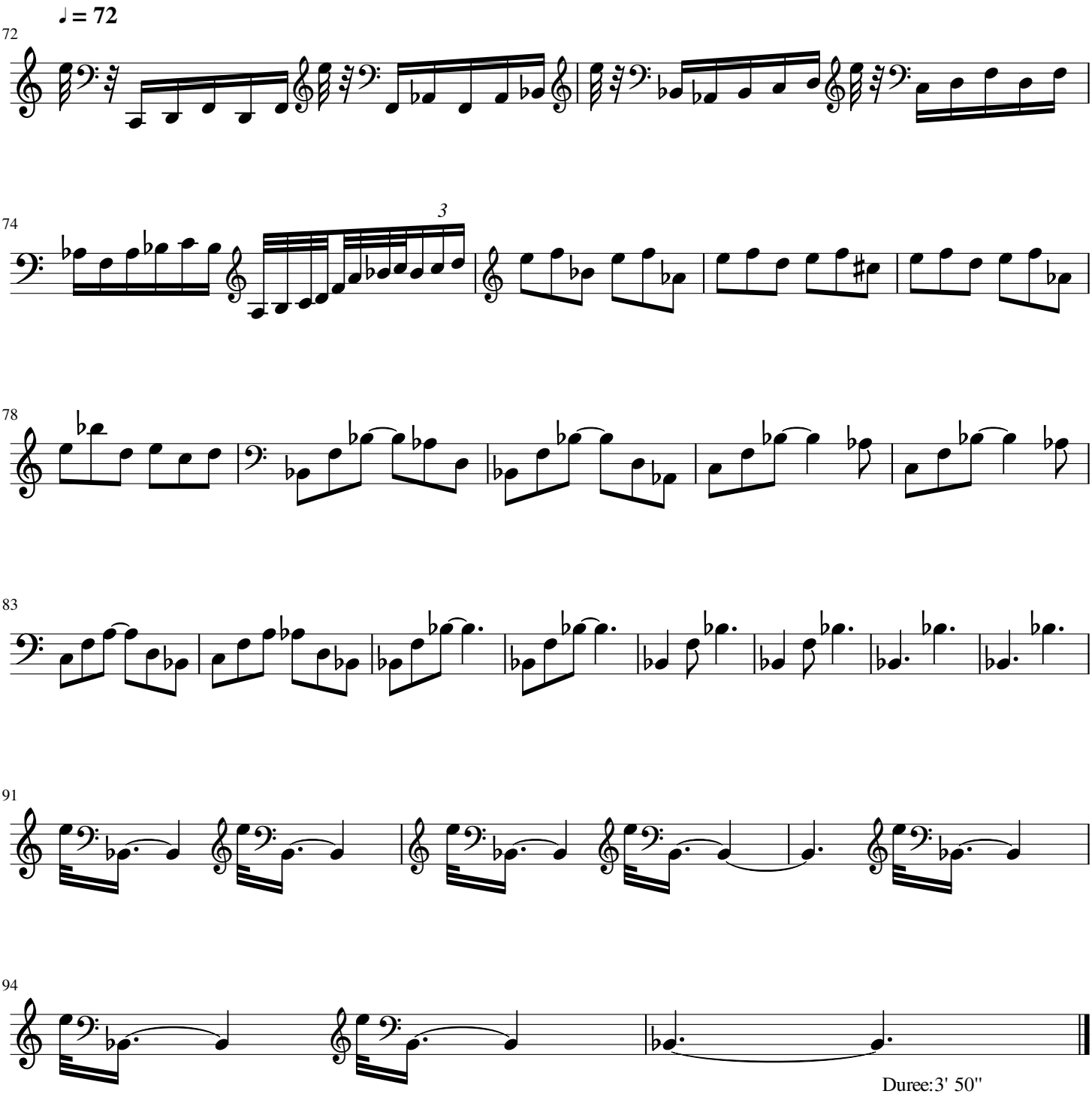
At the Edge

Barcarolle

Ali Riza SARAL



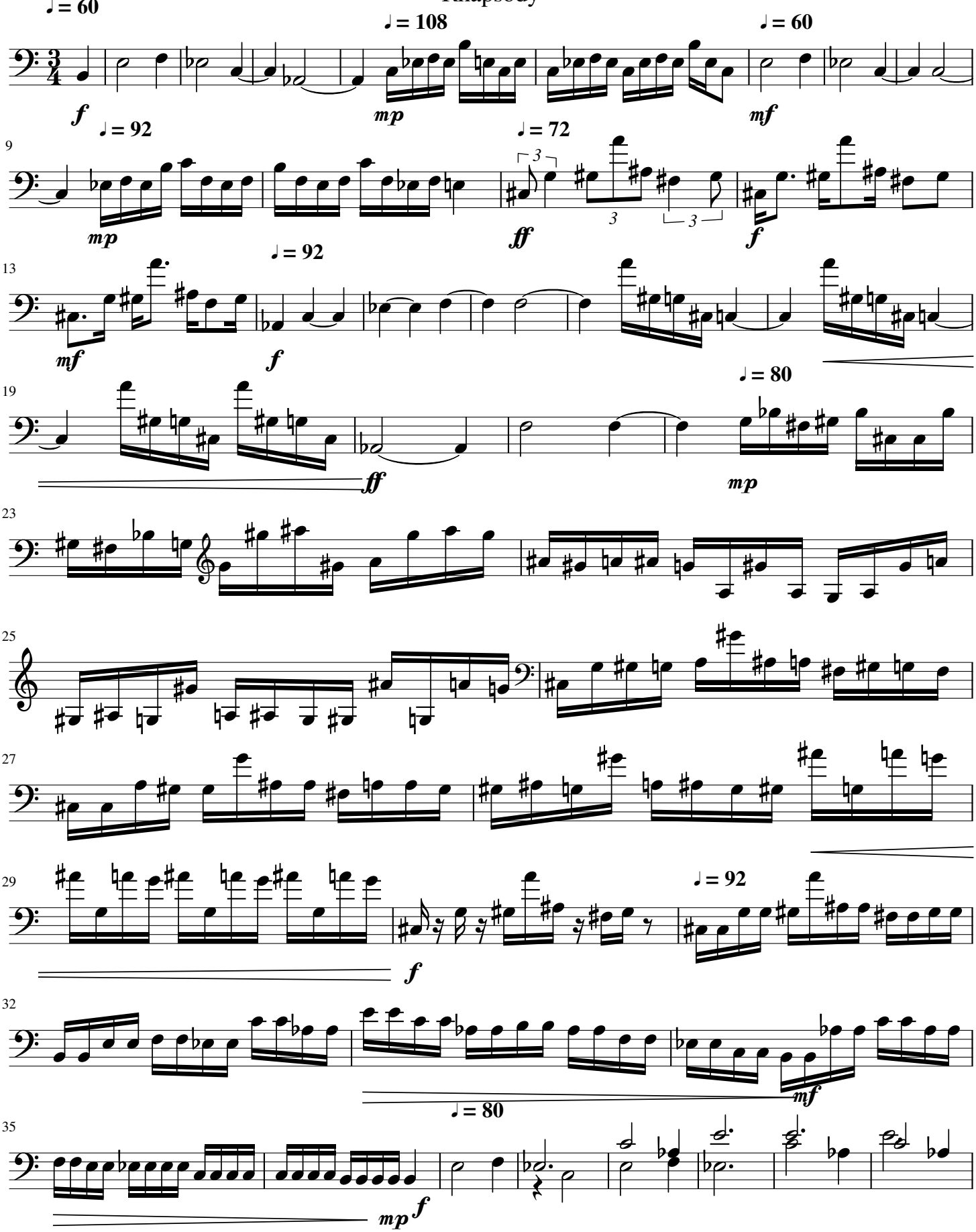

45
48
52
55
58
60
62
64
66
68
70
Accel.
♩ = 60
♩ = 66

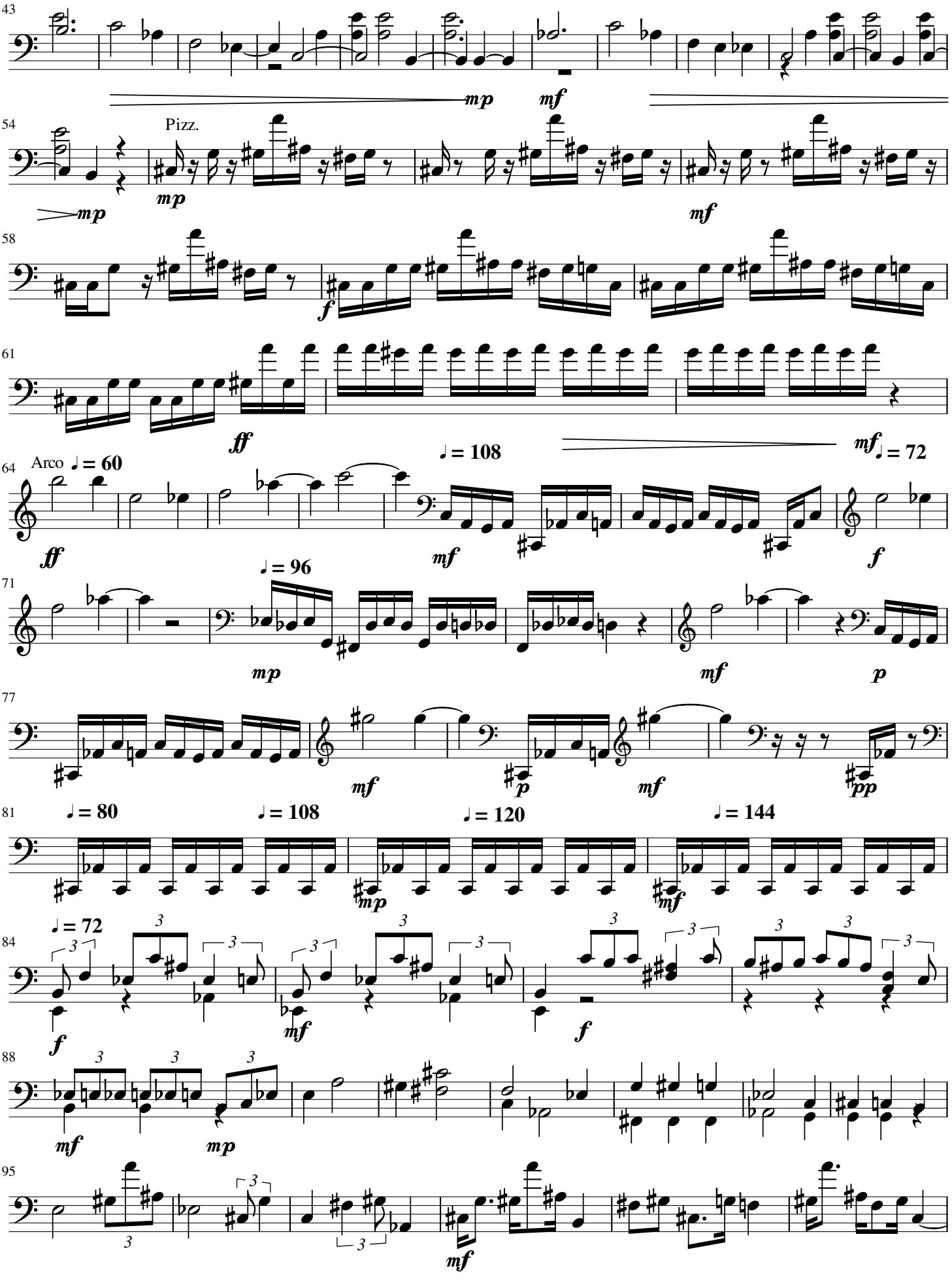

♩ = 72
72
74
3
78
83
91
94
Duree:3' 50''

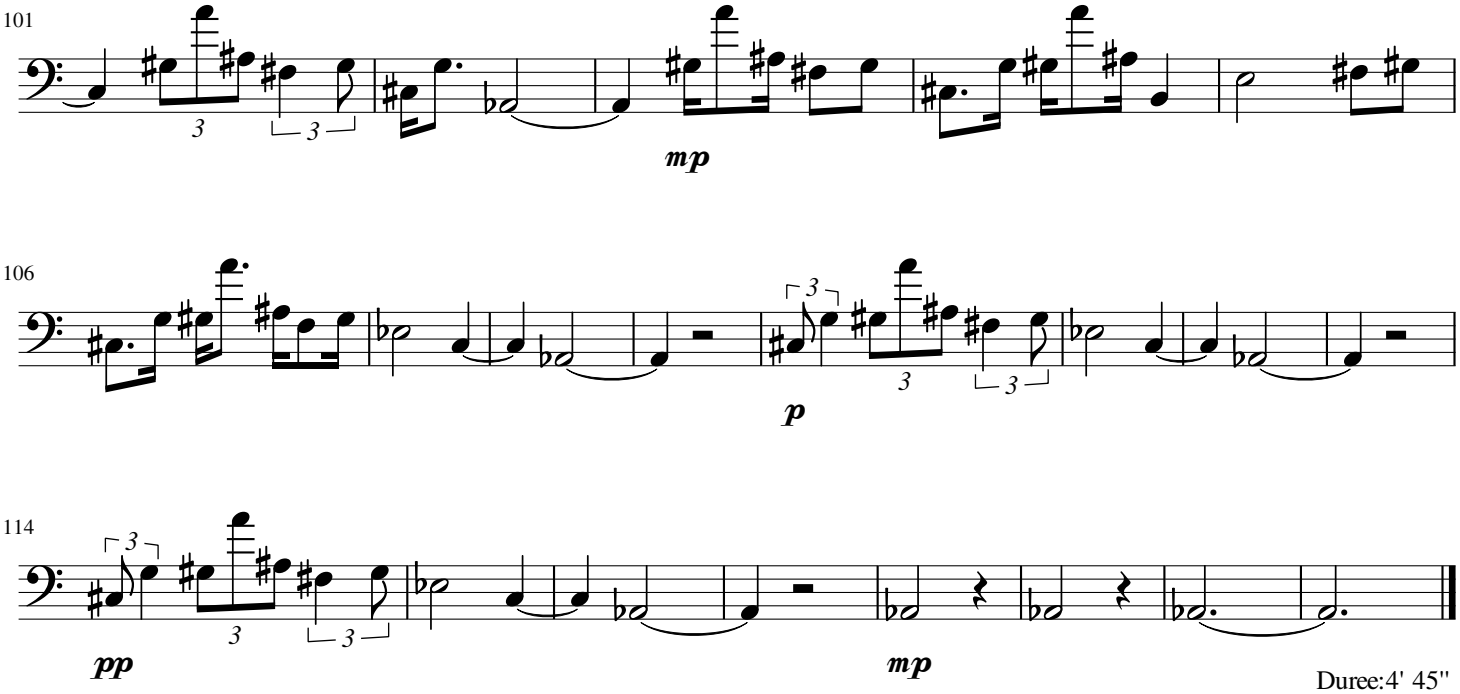
At the Edge

Ali Riza SARAL

Rhapsody




43
mp
mf
54
Pizz.
mp
mp
mf
58
f
61
ff
♩ = 108
mf
♩ = 72
64
Arco ♩ = 60
ff
mf
f
♩ = 96
71
mp
mf
p
77
mf
p
mf
pp
♩ = 80
♩ = 108
♩ = 120
♩ = 144
81
mp
mf
♩ = 72
84
f
mf
f
88
mf
mp
95
mf



Duree:4' 45"

Tot. Duree:12' 10"

SKETCHES

From BAGATELLA:

```
  public static String produceMusic(){
String m ="";
m="T20 V0 I[Piano] Ro Eo Co Do Eo Eo Co Do Eo| Eo Co Do Eo Eo Co Do Eo";
m="V0 I[Piano] Eq Ch. | Eq Ch. | Dq Eq Dq Cq | Cw   V1 I[Flute] Rw | Rw | GmajQQQ CmajQ | Cmajw";
m="V0 I[Cello] Eq Ch. | Eq Ch. | Dq Eq Dq Cq | Cw ";
m= "T100 V0 C6qa81 D6qa90 E6ha100 F6ha111";
String note[]= {"C","C#","D","D#","G","G#","A","A#","B"};
int seriesLength=9;
int noteCount=48; int depthRepeat=2, currentPOS=0;
String playedNotes[] = new String[noteCount];
m="T60 V0 I[Cello] E5w F4w F#3w T80";
int noteSeq=-1;
for (int n=0;n<noteCount;n++){
   noteSeq=randomInteger(1,seriesLength-1);
   currentPOS=n;
   if (!existsPrev(playedNotes,note[noteSeq], depthRepeat, currentPOS) || n==0){
      m+=" "+note[noteSeq]+"4s";
       playedNotes[n]=note[noteSeq]+"4s";
   }
   else n=n-1;
}
System.out.println(m);
return m;

// Note letter (C, D, E, F, G, A, or B), accidental (#, b, natural), and octave
//• Rest is specified with an R • Duration: w, h, q, i, s, t, x, o (whole, half, etc. down to 128th)
//                                    4  8  16 32
   }



   public static String produceMusic4(){
   String sheetStr ="";
   //String note[]= {"C","C#","D","D#","A#","B"};
   String note[]= {"E5", "F4", "F#3", "G5", "D#4", "G#3", "A5", "C5", "B4", "C#4", "D3", "A#2"};
   int seriesLength=12;

   int spanDepth=1, spanBottom=3;
   int noteCount=80, depthRepeat=4;
   int currentPOS=0, noteSeq=-1;
   int octave=0;

   String playedNotes[] = new String[noteCount];
```

```
   String tempi[] ={"T126","T120","T132","T144"};
   int tempiCount=4, tempiSeq=0;

    sheetStr="T92 V0 I[Cello] E5q F4q F#3q G4q D#4q G#4q A4q C4q B4q C#4q D4q A#4q T120";


   for (int n=0;n<noteCount;n++){
      //octave = randomInteger(spanBottom,spanBottom+spanDepth);
      noteSeq=randomInteger(1,seriesLength-1);
      if (n % 4==0) tempiSeq=randomInteger(1,tempiCount-1);
      currentPOS=n;
      if (!existsPrev(playedNotes,note[noteSeq], depthRepeat, currentPOS) || n==0){
//          m+=" "+note[noteSeq]+octave+"i";
//           playedNotes[n]=note[noteSeq]+octave+"i";
             sheetStr+=" "+note[noteSeq]+"i "+tempi[tempiSeq];
             playedNotes[n]=note[noteSeq]+"i "+tempi[tempiSeq];
                }
      else n=n-1;
   }
   System.out.println(sheetStr);
   return sheetStr;
}

    public static String produceMusic5(){
   String sheetStr ="";
   //String note[]= {"C","C#","D","D#","A#","B"};
   String note[]= {"E5", "F4", "F#3", "G5", "D#4", "G#3", "A5", "C5", "B4", "C#4", "D3", "A#2"};
   int seriesLength=12;
   String note2[]= {"E6", "F6", "D#6", "C6"}; int noteSeq2=-1;
   int series2Length=4;


   int spanDepth=1, spanBottom=3;
   int noteCount=32, depthRepeat=4;
   int currentPOS=0, noteSeq=-1;
   int octave=0;
   int remainingCount=2 * noteCount;
   String playedNotes[] = new String[noteCount];

   String tempi[] ={"T126","T120","T132","T144"};
   int tempiCount=4, tempiSeq=0;

    sheetStr="T92 V0 I[Cello] E5q F4q F#3q G4q D#4q G#4q A4q C4q B4q C#4q D4q A#4q T120";


   for (int n=0;n<noteCount;n++){
      //octave = randomInteger(spanBottom,spanBottom+spanDepth);
      noteSeq=randomInteger(1,seriesLength-1);
      noteSeq2=randomInteger(1,series2Length-1);
      if (n % 4==0) tempiSeq=randomInteger(1,tempiCount-1);
      currentPOS=n;
      if ((!existsPrev(playedNotes,note[noteSeq], depthRepeat, currentPOS)) || n==0){
```

```
//          m+=" "+note[noteSeq]+octave+"i";
//           playedNotes[n]=note[noteSeq]+octave+"i";
             sheetStr+=" "+note[noteSeq]+"i "+tempi[tempiSeq];
             playedNotes[n]=note[noteSeq]+"i "+tempi[tempiSeq];
             sheetStr+=" "+note2[noteSeq2]+"i ";
             playedNotes[n]=note[noteSeq2]+"i ";
                 }
      else n=n-1;
      remainingCount--;
      if (remainingCount < 8) break;
   }
   System.out.println(sheetStr);
   return sheetStr;
}

    public static String produceMusic83(){
   String m ="";
   String note[]= {"C","C#","D","D#","A#","B"};
   int seriesLength=6, spanDepth=1, spanBottom=3, noteCount=48, depthRepeat=2;
   int currentPOS=0, noteSeq=-1, octave=0;
   int remainingCount=noteCount-1;
   String playedNotes[] = new String[noteCount];

   m+="T132 V0 I[Cello] ";

   for (int n=0;n<noteCount;n++){
      octave = randomInteger(spanBottom+1,spanBottom+spanDepth);
      noteSeq=randomInteger(1,seriesLength-1);
      currentPOS=n;
      if (remainingCount < 8) break;
      if (!existsPrev(playedNotes,note[noteSeq], depthRepeat, currentPOS) || n==0){
         m+=" "+note[noteSeq]+octave+"i";
          playedNotes[n]=note[noteSeq]+octave+"i";
                }
      else {n=n-1;remainingCount++;}
      remainingCount--;
   }
    m+=" T72 A5q C5q B4q C#4q D3q A#2q Rq Rq";
    System.out.println(m);
   return m;
}
```

SHORT ARTICLE

Events are consequent.
We perceive events in an episodic sequence.

We perceive sentences word by word. But we understand them as a whole.
The meaning arises from the interaction of words grammatically, syntactically and lexiacally.

Knowledge is drived semantically and stored as such.
We remember episodic events but we know semantic information.

Music is an ever continuing series of sequential events, whether polyphonic or mono.
Remembering music depends on remebering the next item of the episodic sequence.

The meaning of music lies in the semantic elements.
Elements that can be perceived not sequentially but in parallel processing.
Human aural memory is not very long but it can be enhanced by GROUPING.
Each note being perceived may be related with a group of notes in the aural working memory.
Motives, sentences phrases have semantic relations such as IS A, BELOGS TO, INHERITS FROM etc.

Music has a semantical meaning.

About the Composer

Ali R+ SARAL has studied double majors and has two Master's degrees,

MSc and BSC from Istanbul Technical University and MM from Illinois State University.

He was accepted to the Istanbul State Conservatoir Composition and Theory Department and

studied harmony, counterpoint and others with Erçivan SAYDAM,

composition with İlhan USMANBAŞ.

He was a student of Roque CORDERO and Arthur CORRA at USA.

He has attended courses at Stanford University CCRMA lab and Darmstad 1996.

He has written 27 works of music approx. 6,5 hours long.

You can find his compositions at Petrucci Library: http://imslp.org/wiki/Category:Saral,_Ali_Riza

He has served EUROCONTROL and German airspace as an air traffic control engineer 1992-1997.
He has specialized in LARGESYSTEMS and Java-J2EE.
His areas of interest include Systems Psychology, ANN and parsers, automatic language translation.

He writes blogs at:
http://largesystems-atc.blogspot.com/
http://largesystems-atc-en.blogspot.com/
http://alirizasaral.blogspot.com/

You can reach him at:

Addr: Barbaros Mah Sedef Sk Onur Sit. Akasya Apt. 13/3G Uskudar ISTANBUL TÜRKİYE 34662

Tel: 90 506 887 8401

e-mail: Arsaral(aatt)yahoo.com