Enhancing Angklung Performances with Python

EuroPython 2019

Trapsilo Bumi

HENNGE

tbumi

tbumi@thpd.io

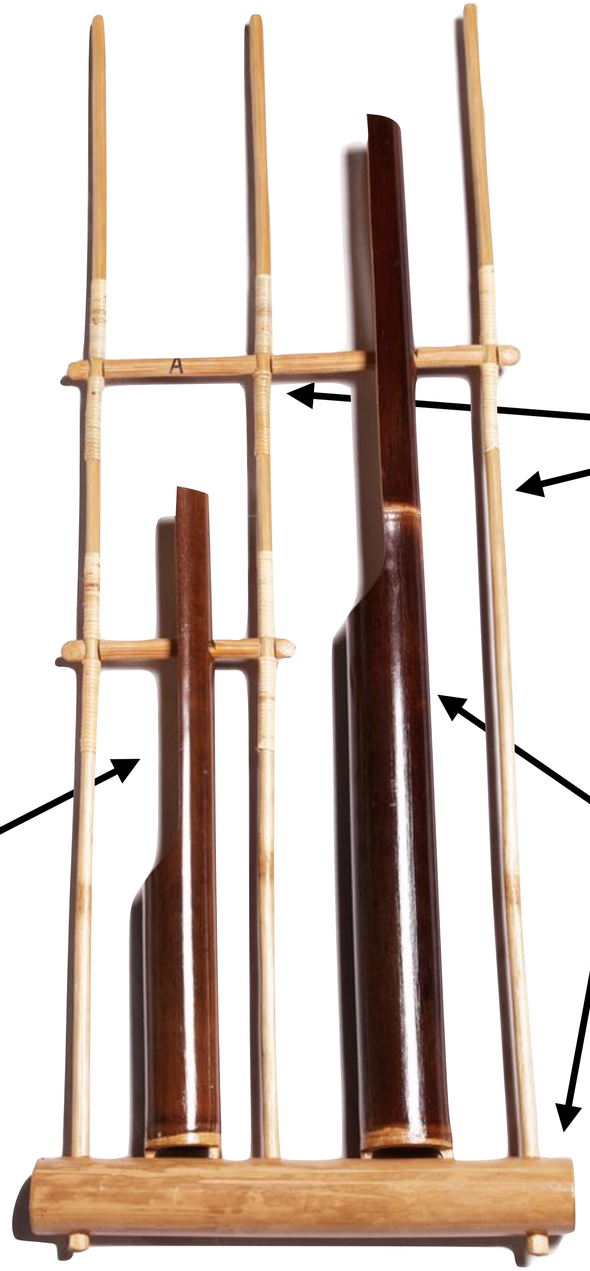

Source: Wikimedia Commons

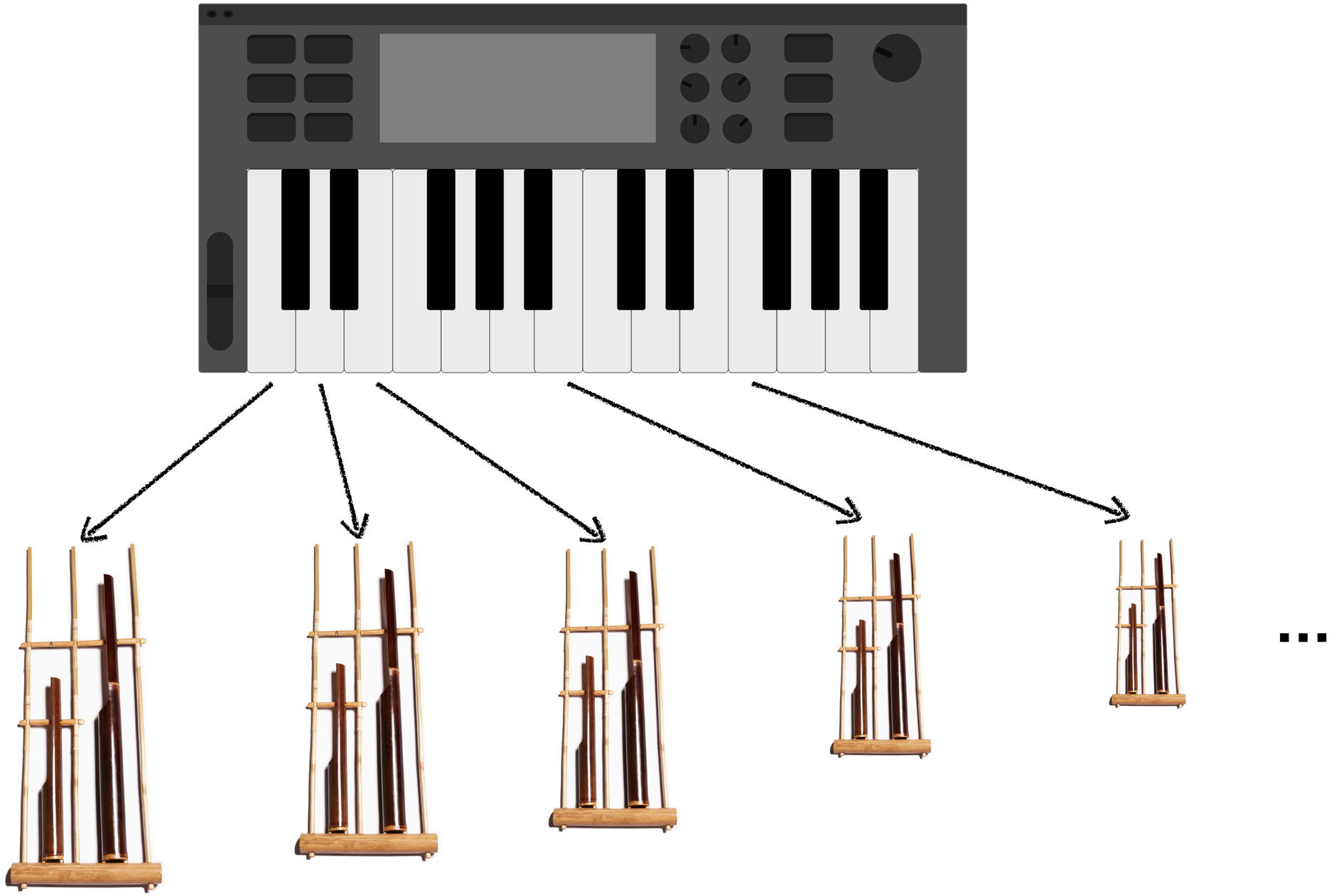
Source: Wikimedia Commons

About Me

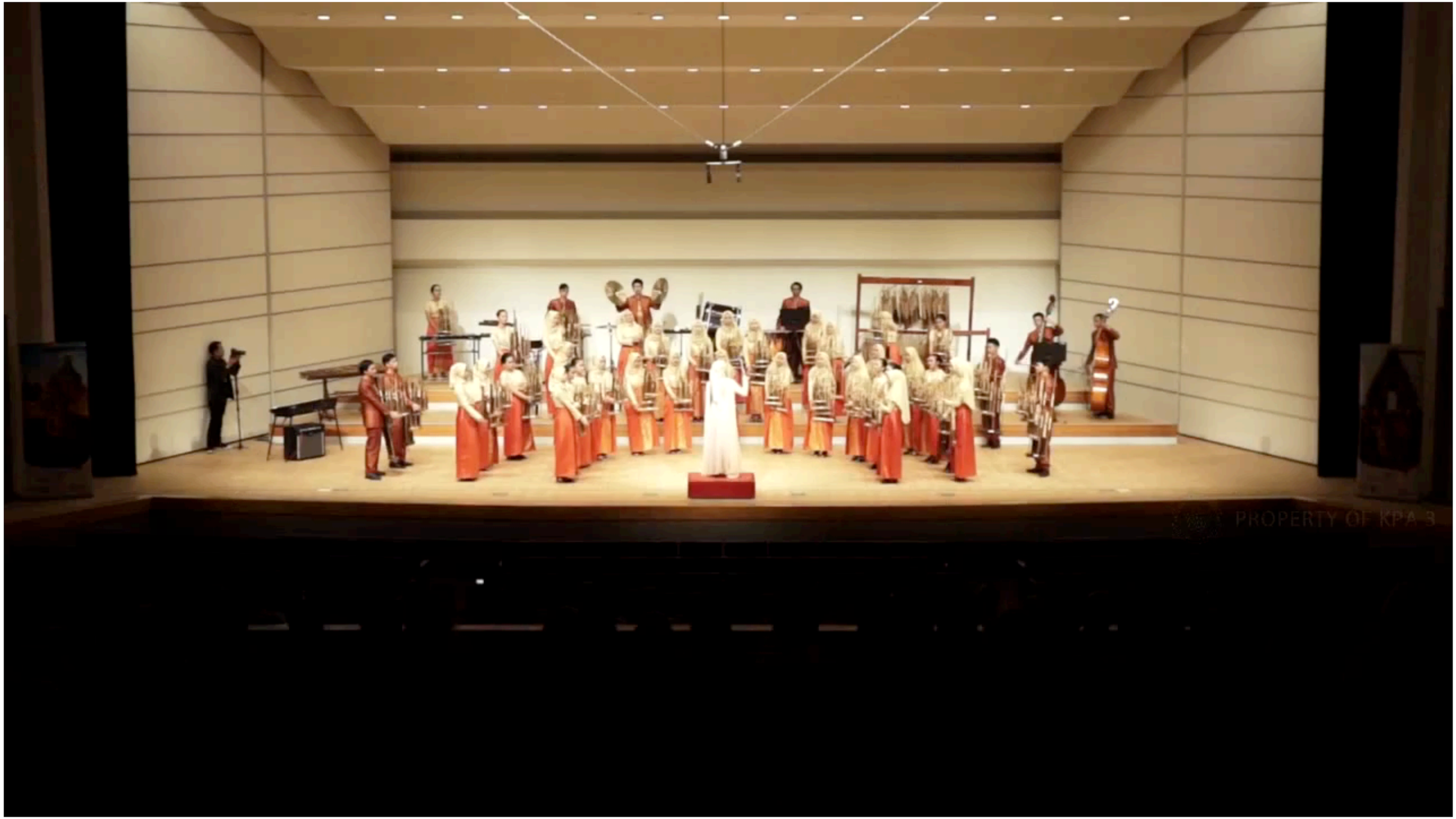
- Born and raised in Bandung, Indonesia
- Currently a Software Engineer for HENNGE (Tokyo, Japan)
- Angklung enthusiast for over 10 years
  - Performer in Italy, Greece, Malaysia, Singapore
  - Conductor in several concerts and performances

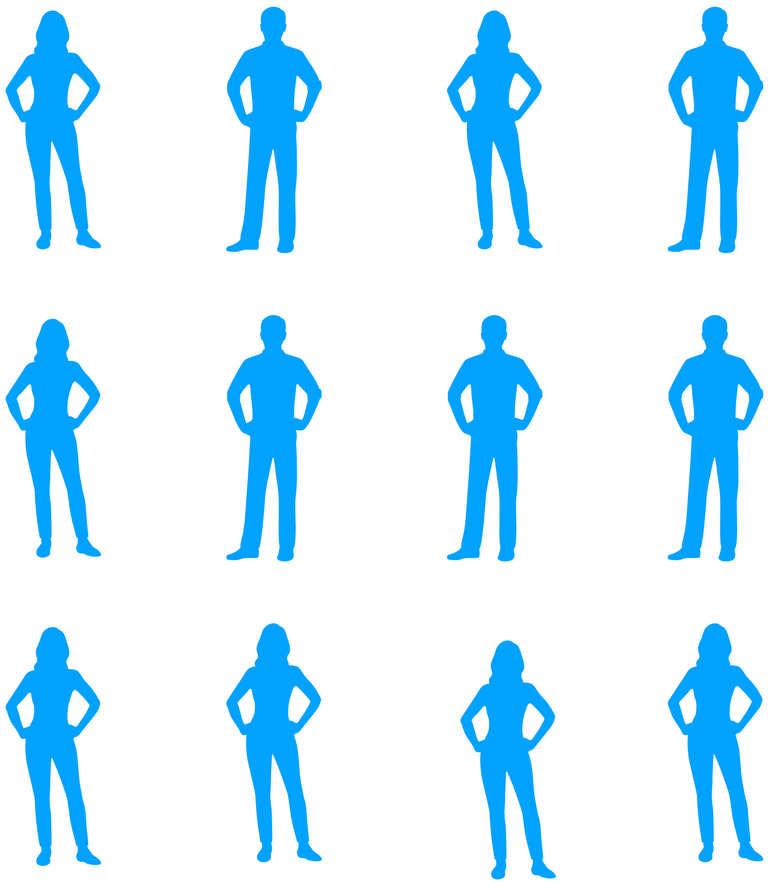
What’s an Angklung?


Frame
Large Tube
Small Tube
Vibration produces a certain tone



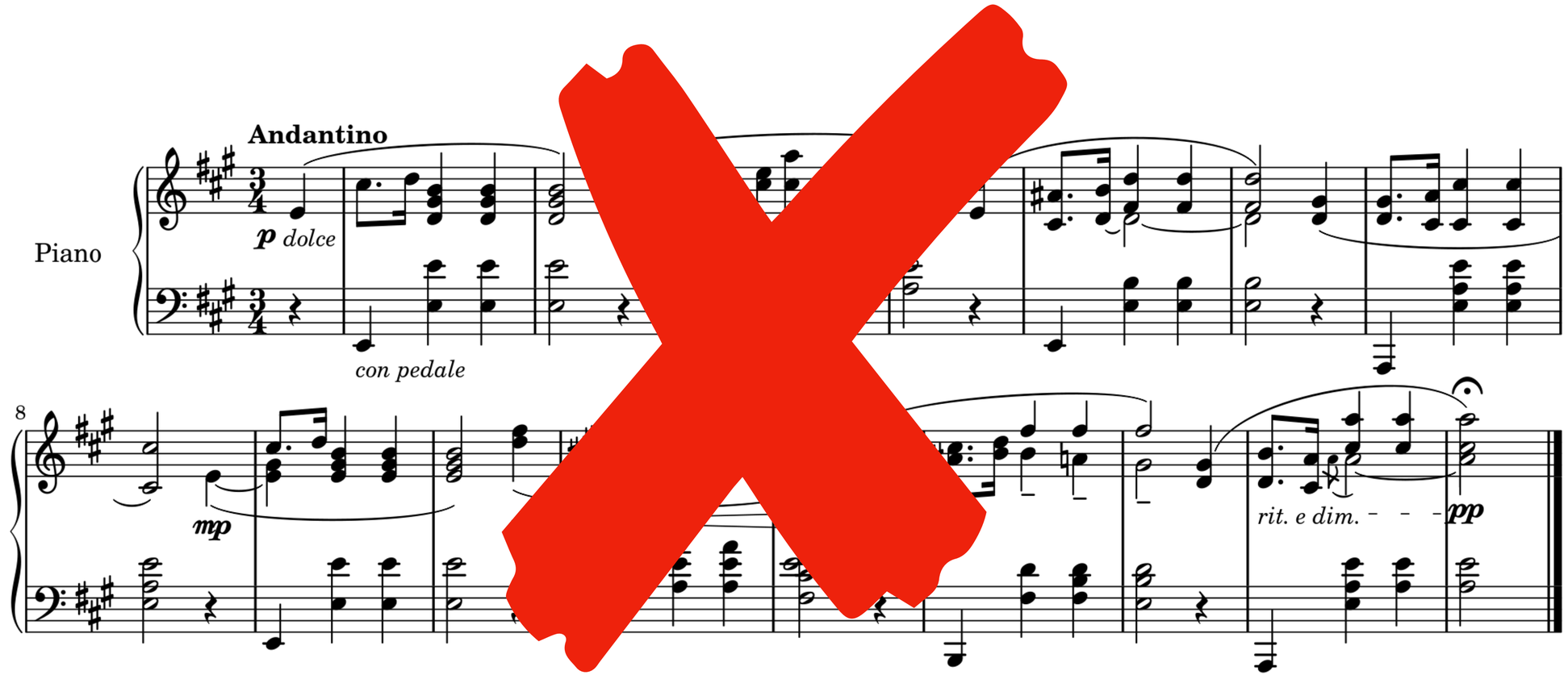
Keluarga Paduan Angklung SMA Negeri 3 Bandung
Esplanade Concert Hall, Singapore, 2011


PROPERTY OF KPA 3



Angklung Sheet Music

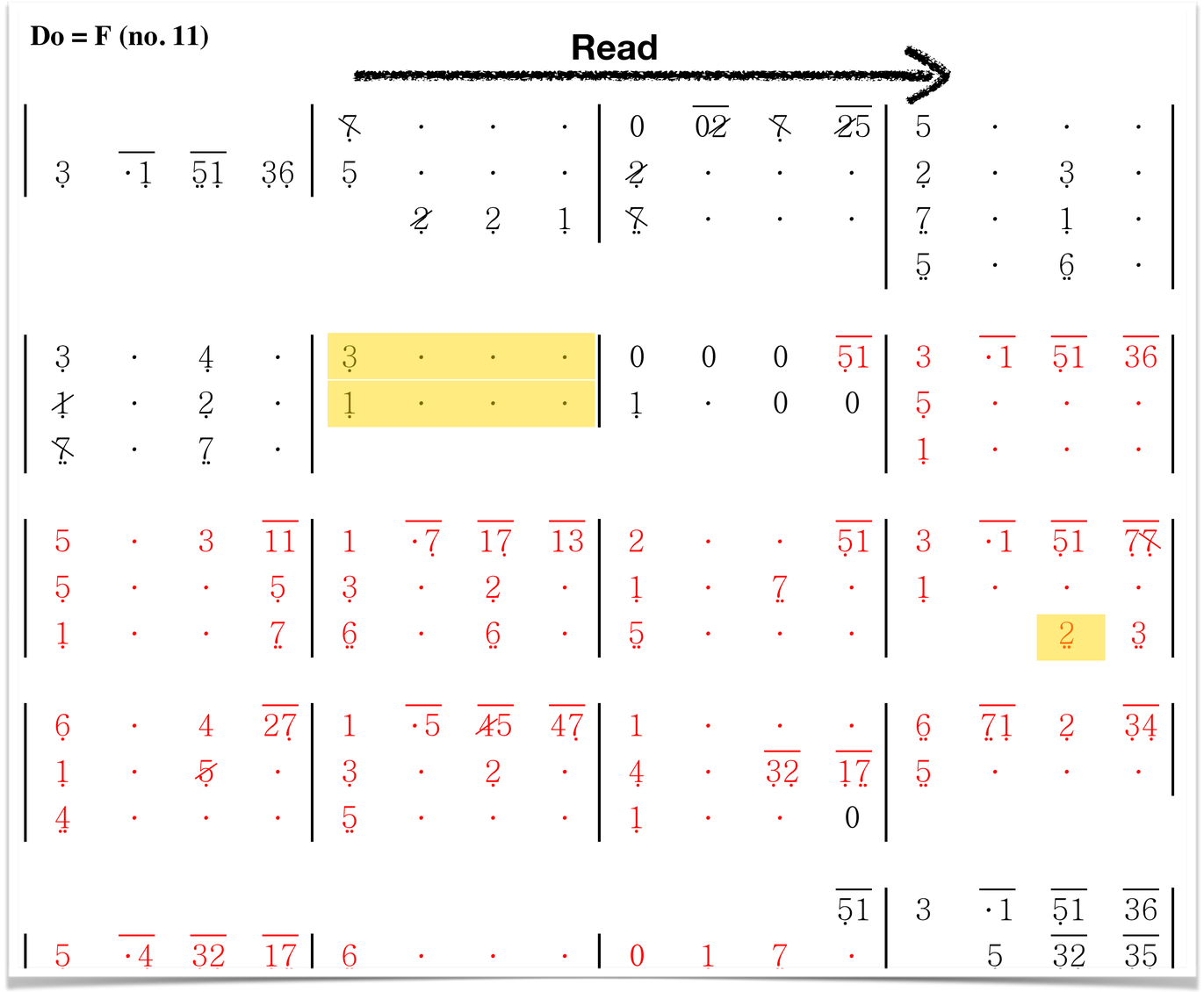
(or Music Score)


Andantino
Piano
p dolce
con pedale
8
mp
rit. e dim.
pp

Cipher Notation

or *Not Angka (lit. Number Notes)*

Do = F (no. 11)



1 2 3 4 5 6 7 i

do re mi fa so la si do

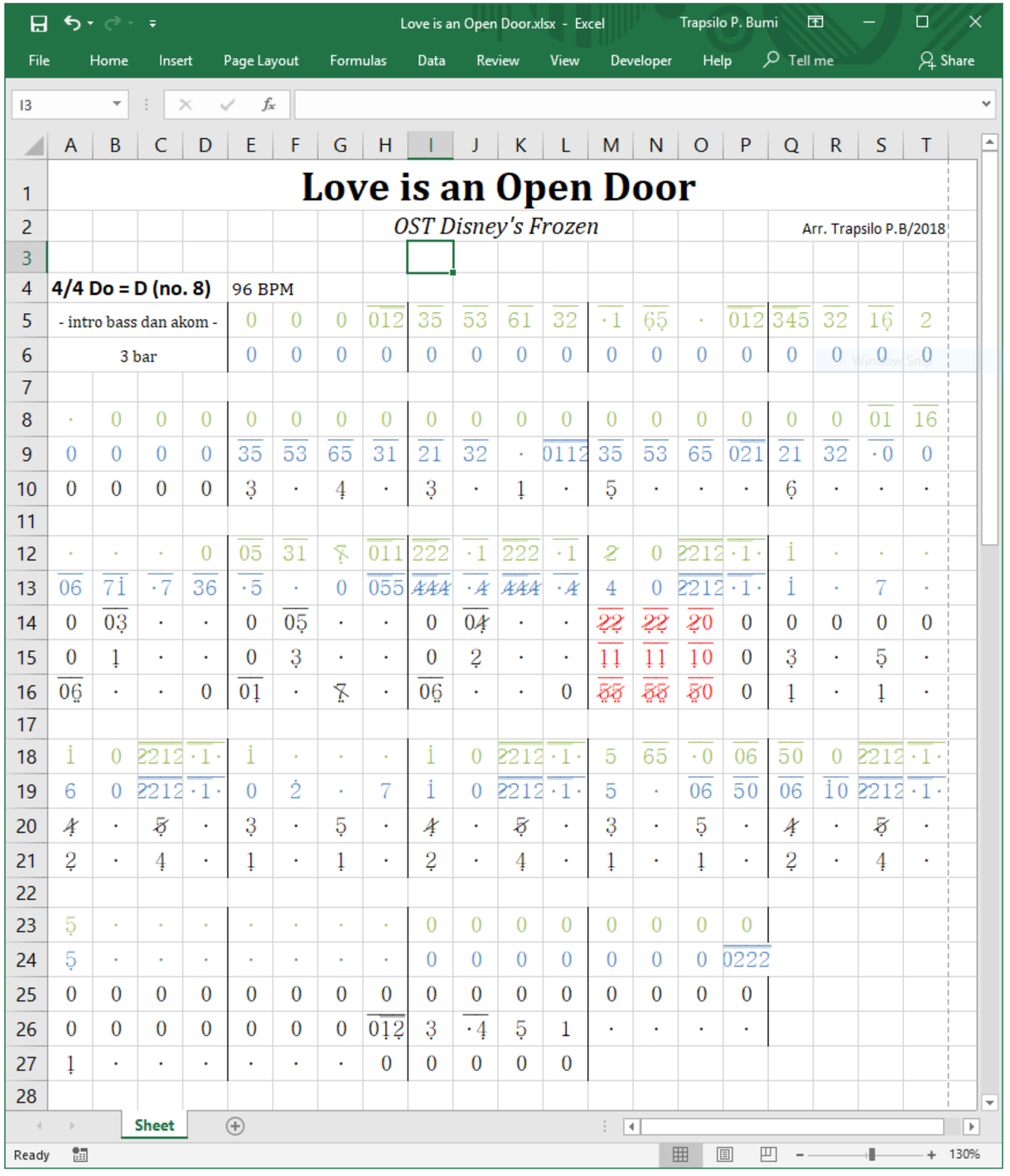
The Algorithm

Love is an Open Door

OST Disney's Frozen

Arr. Trapsilo P.B/2018

4/4 Do = D (no. 8)				96 BPM																
- intro bass dan akom -				0	0	0	012	35	53	61	32	·1	6̣5̣	·	012	345	32	16̣	2	
3 bar				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
·	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	01	16	
0	0	0	0	35	53	65	31	21	32	·	0112	35	53	65	021	21	32	·0	0	
0	0	0	0	3̣	·	4̣	·	3̣	·	1̣	·	5̣	·	·	·	6̣	·	·	·	
·	·	·	0	05	31	~~7̇~~	011	222	·1	222	·1	~~2~~	0	2212	·1·	i	·	·	·	
06	7i	·7	36	·5	·	0	055	~~444~~	~~·4~~	~~444~~	~~·4~~	4	0	2212	·1·	i	·	7	·	
0	03̣	·	·	0	05̣	·	·	0	~~04̣~~	·	·	~~22~~	~~22~~	~~20~~	0	0	0	0	0	
0	1̣	·	·	0	3̣	·	·	0	~~2̣~~	·	·	~~11~~	~~11~~	~~10~~	0	3̣	·	5̣	·	
06̣	·	·	0	01̣	·	~~7̣~~	·	06̣	·	·	0	~~66~~	~~66~~	~~60~~	0	1̣	·	1̣	·	
i	0	2212	·1·	i	·	·	·	i	0	2212	·1·	5	65	·0	06	50	0	2212	·1·	
6	0	2212	·1·	0	2̇	·	7	i	0	2212	·1·	5	·	06	50	06	i0	2212	·1·	
~~4̣~~	·	~~5̣~~	·	3̣	·	5̣	·	~~4̣~~	·	~~5̣~~	·	3̣	·	5̣	·	~~4̣~~	·	~~5̣~~	·	
2̣	·	4̣	·	1̣	·	1̣	·	2̣	·	4̣	·	1̣	·	1̣	·	2̣	·	4̣	·	
5̣	·	·	·	·	·	·	·	0	0	0	0	0	0	0	0					
5̣	·	·	·	·	·	·	·	0	0	0	0	0	0	0	0222					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0	0	0	0	0	0	0	012	3̣	·4	5̣	1	·	·	·	·					
1̣	·	·	·	·	·	·	0	0	0	0	0									


Love is an Open Door.xlsx - Excel
Trapsilo P. Bumi
File Home Insert Page Layout Formulas Data Review View Developer Help Tell me Share
I3
Love is an Open Door
OST Disney's Frozen
Arr. Trapsilo P.B/2018
4/4 Do = D (no. 8)
96 BPM
- intro bass dan akom -
3 bar
Sheet
Ready
130%

`openpyxl` - A Python library to read/write Excel 2010 xlsx/xlsm files

Author: Eric Gazoni, Charlie Clark

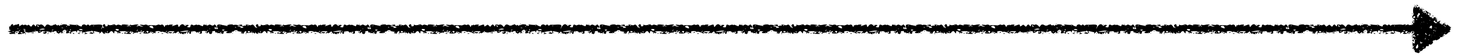
Source code: http://bitbucket.org/openpyxl/openpyxl/src

Issues: http://bitbucket.org/openpyxl/openpyxl/issues

Generated: Aug 17, 2018

Version: 2.5.5

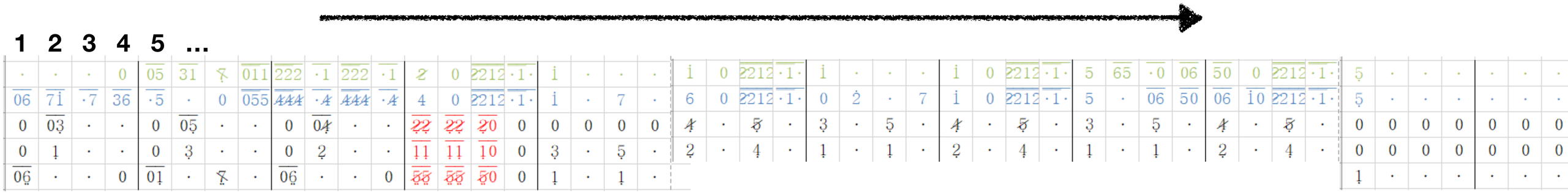
·	·	·	0	05	31	4	011	222	·1	222	·1	2	0	2212	·1·	i	·	·	·
06	7i	·7	36	·5	·	0	055	444	·4	444	·4	4	0	2212	·1·	i	·	7	·
0	03	·	·	0	05	·	·	0	04	·	·	22	22	20	0	0	0	0	0
0	1	·	·	0	3	·	·	0	2	·	·	11	11	10	0	3	·	5	·
06	·	·	0	01	·	4	·	06	·	·	0	66	66	60	0	1	·	1	·
i	0	2212	·1·	i	·	·	·	i	0	2212	·1·	5	65	·0	06	50	0	2212	·1·
6	0	2212	·1·	0	2	·	7	i	0	2212	·1·	5	·	06	50	06	i0	2212	·1·
4	·	6	·	3	·	5	·	4	·	6	·	3	·	5	·	4	·	6	·
2	·	4	·	1	·	1	·	2	·	4	·	1	·	1	·	2	·	4	·
5	·	·	·	·	·	·	·	0	0	0	0	0	0	0	0				
5	·	·	·	·	·	·	·	0	0	0	0	0	0	0	0222				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0	0	0	0	0	0	0	012	3	·4	5	1	·	·	·	·				
1	·	·	·	·	·	·	0	0	0	0	0								



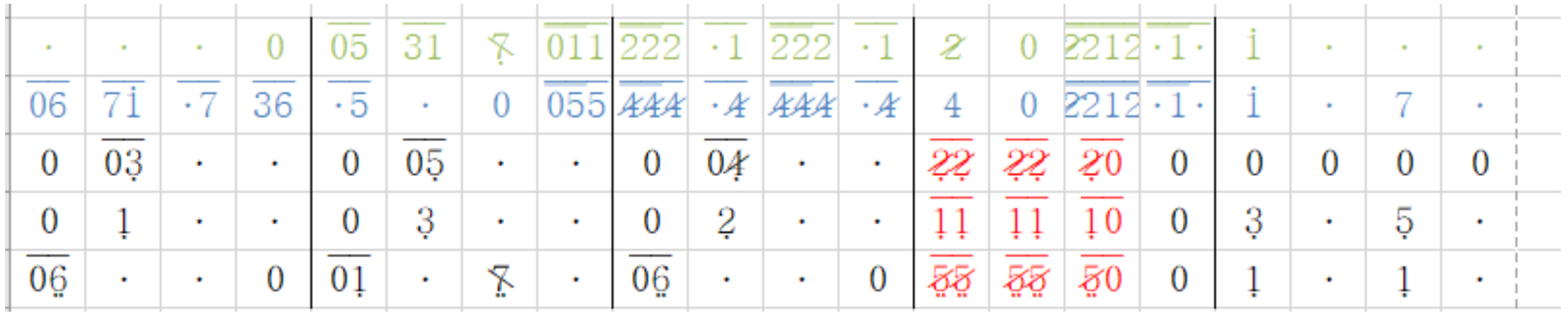
·	·	·	0	05	31	4	011	222	·1	222	·1	2	0	2212	·1·	i	·	·	·
06	7i	·7	36	·5	·	0	055	444	·4	444	·4	4	0	2212	·1·	i	·	7	·
0	03	·	·	0	05	·	·	0	04	·	·	22	22	20	0	0	0	0	0
0	1	·	·	0	3	·	·	0	2	·	·	11	11	10	0	3	·	5	·
06	·	·	0	01	·	4	·	06	·	·	0	55	55	50	0	1	·	1	·

i	0	2212	·1·	i	·	·	·	i	0	2212	·1·	5	65	·0	06	50	0	2212	·1·
6	0	2212	·1·	0	2	·	7	i	0	2212	·1·	5	·	06	50	06	i0	2212	·1·
4	·	5	·	3	·	5	·	4	·	5	·	3	·	5	·	4	·	5	·
2	·	4	·	1	·	1	·	2	·	4	·	1	·	1	·	2	·	4	·

5	·	·	·	·	·	·	·	0	0	0	0	0	0	0	0
5	·	·	·	·	·	·	·	0	0	0	0	0	0	0	0222
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	012	3	·4	5	1	·	·	·	·
1	·	·	·	·	·	·	0	0	0	0	0				




```
wb = openpyxl.load_workbook(file_name)
score = []  # music score, List[List[notes]]
for row in wb.active.rows:
  is_empty_row = True
  for cell in row:
    if re.match(NOTE_VALID_CHARS, cell.value):
      is_empty_row = False
      new_row.append(cell.value)
  if is_empty_row:
    internal_row = 0
  score[internal_row] += new_row
  internal_row += 1
```



Collision Table

	1	2	3
2	13	-	-
3	40	2	-
4	8	38	27

Play Time Information

1	2	3	4
100	48	76	45

```python
def calculate_collision_table(score):
    angklung_per_column = {}
    all_angklung = set()
    for row in score:
        for col, beat in enumerate(row):
            if col not in angklung_per_column:
                angklung_per_column[col] = set()
            for m in re.finditer("<[0-9A-Gg#]+>", beat):
                angklung = m[0].strip("<>")
                angklung_per_column[col].add(angklung)
                all_angklung.add(angklung)

    collision_table = {}
    while len(all_angklung) > 0:
        a = all_angklung.pop()
        for b in all_angklung:
            collision_table[pair(a, b)] = 0

    for i, col in angklung_per_column.items():
        processing = col.copy()
        while len(processing) > 0:
            a = processing.pop()
            for b in list(processing) + list(angklung_per_column.get(i+1, set()) - {a}):
                collision_table[pair(a, b)] += 1

    # normalize
    max_collision = (len(score) * 2) - 1
    norm_collision_table = {}
    for a_pair in collision_table:
        norm_collision_table[a_pair] = collision_table[a_pair] / max_collision

    return norm_collision_table
```

```
def calculate_play_time(*partiturs):
    play_time = {}
    for row in score:
        joined_row = ''.join(row)
        for m in re.finditer(r'(-|-=)?(<[A-Gg0-9#]{1,2}>)((?:(?:-|-=)?\.+)*)', joined_row):
            if m[1] == '-':
                duration = 0.5
            elif m[1] == '-=':
                duration = 0.25
            else:
                duration = 1
            for m2 in re.finditer(r'(-|-=)?\.', m[3]):
                if m2[1] == '-':
                    duration += 0.5
                elif m2[1] == '-=':
                    duration += 0.25
                else:
                    duration += 1
            angklung = m[2].strip('<>')
            try:
                play_time[angklung] += duration
            except KeyError:
                play_time[angklung] = duration

    # normalize
    total_len_partiturs = sum(len(partitur) for partitur in partiturs)
    norm_play_time = {}
    for no_angklung in play_time:
        norm_play_time[no_angklung] = play_time[no_angklung] / total_len_partiturs

    return norm_play_time
```

Then, we optimize the distribution based on:

1. **Minimum** number of **collisions**
2. **Maximum** amount of **play time**
3. Good balance of size of Angklung

Remember linear programming? Finding optimum value?

```python
def generate_distribution(play_time, collision_table, num_players, num_each_angklung):
    angklung_to_distribute = []
    for a, j in num_each_angklung.items():
        angklung_to_distribute.extend([a for _ in range(j)])
    random.shuffle(angklung_to_distribute)

    distribution = {i: [] for i in range(num_players)}
    while len(angklung_to_distribute) > 0:
        angklung_candidate = angklung_to_distribute.pop()
        candidate_values = {}
        for player_index, player_distribution in distribution.items():
            if angklung_candidate in player_distribution:
                continue

            weight = 0
            for players_angklung in player_distribution:
                weight -= play_time[players_angklung]
                weight -= collision_table[pair(angklung_candidate, players_angklung)]
                if angklung_candidate in LOW_ANGKLUNG and \
                   players_angklung in LOW_ANGKLUNG:
                    weight -= 1
                weight -= 1 / abs(no_angklung_to_numeral(angklung_candidate)
                                  - no_angklung_to_numeral(players_angklung))
            weight -= len(player_distribution) * 3
            candidate_values[player_index] = weight
        if len(candidate_values) == 0:
            raise Exception("Too few players")
        candidate_values_sorted = sorted([(v, i) for i, v in candidate_values.items()],
                                         key=lambda t: t[0], reverse=True)
        distribution[candidate_values_sorted[0][1]].append(angklung_candidate)

    distribution = {i: sorted(angklungs, key=no_angklung_to_numeral)
                    for i, angklungs in distribution.items()}
    return distribution
```

- Many possibilities for future improvements of other aspects of Angklung with technology
- Lots of details skipped: key signatures, conversion of relative to absolute notes, etc.

Thank you!

P.S. Come work in Japan:

https://hennge.com/global/gip

Global Internship Program


Unlock your Potential.
HENNGE
GLOBAL
INTERNSHIP
PROGRAM