

Absolute Pitch

Jonghwan Park
2012006522

Yeonjun Choi
2015003818

CONTENTS

1. Motivation

2. Timetable

3. Roles

4. Algorithm

5. Simulation

6. Conclusion

1. Motivation

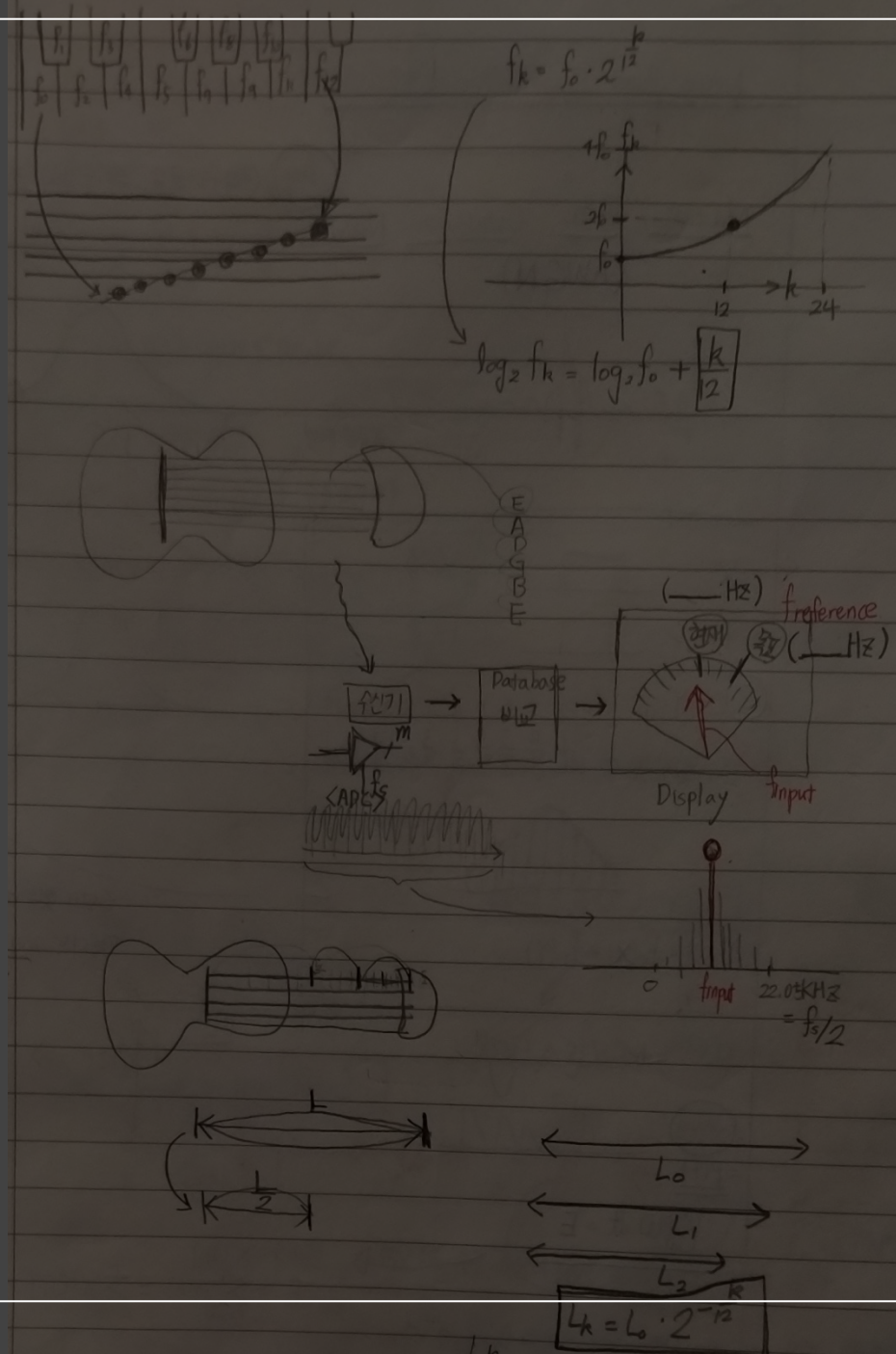
Absolute
Pitch

Absolute
Pitch

1. Motivation

Absolute Pitch

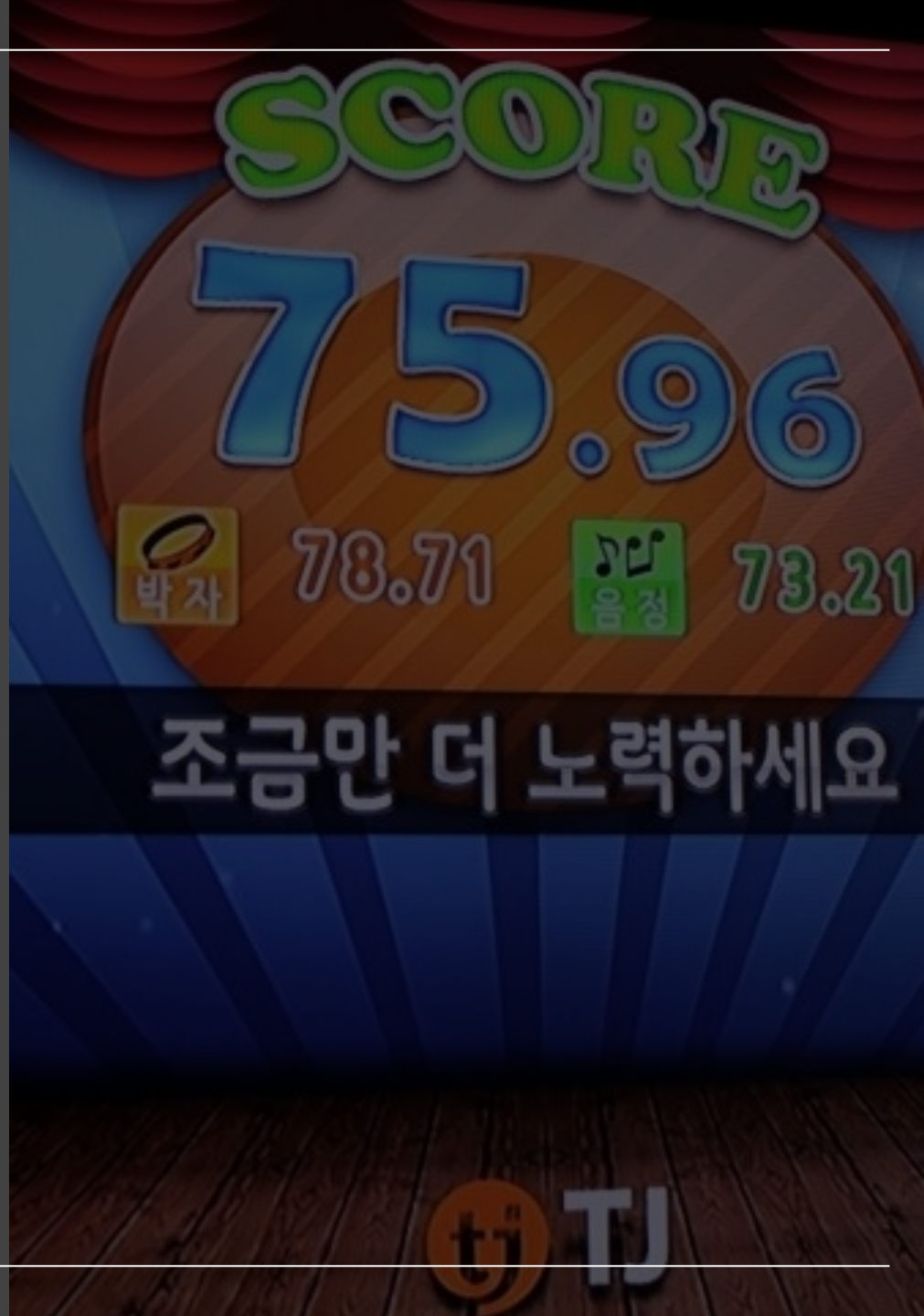
Guitar tuner principle
learned in Digital
Signal Processing class



1. Motivation

Absolute Pitch

Wanting to train our
vocal ability







2. Timetable

Absolute
Pitch

Absolute
Pitch

2. Timetable

3 ~ 4 月	5 ~ 6 月	7 月	8 ~ 10 月	10 月
				
Brainstorming	Review feasibility	Final idea	Develop	Debugging and UI

3. Roles

Absolute
Pitch

Absolute
Pitch

3. Roles

Absolute Pitch

Individually developing
the same functions and
merging the optimal
codes

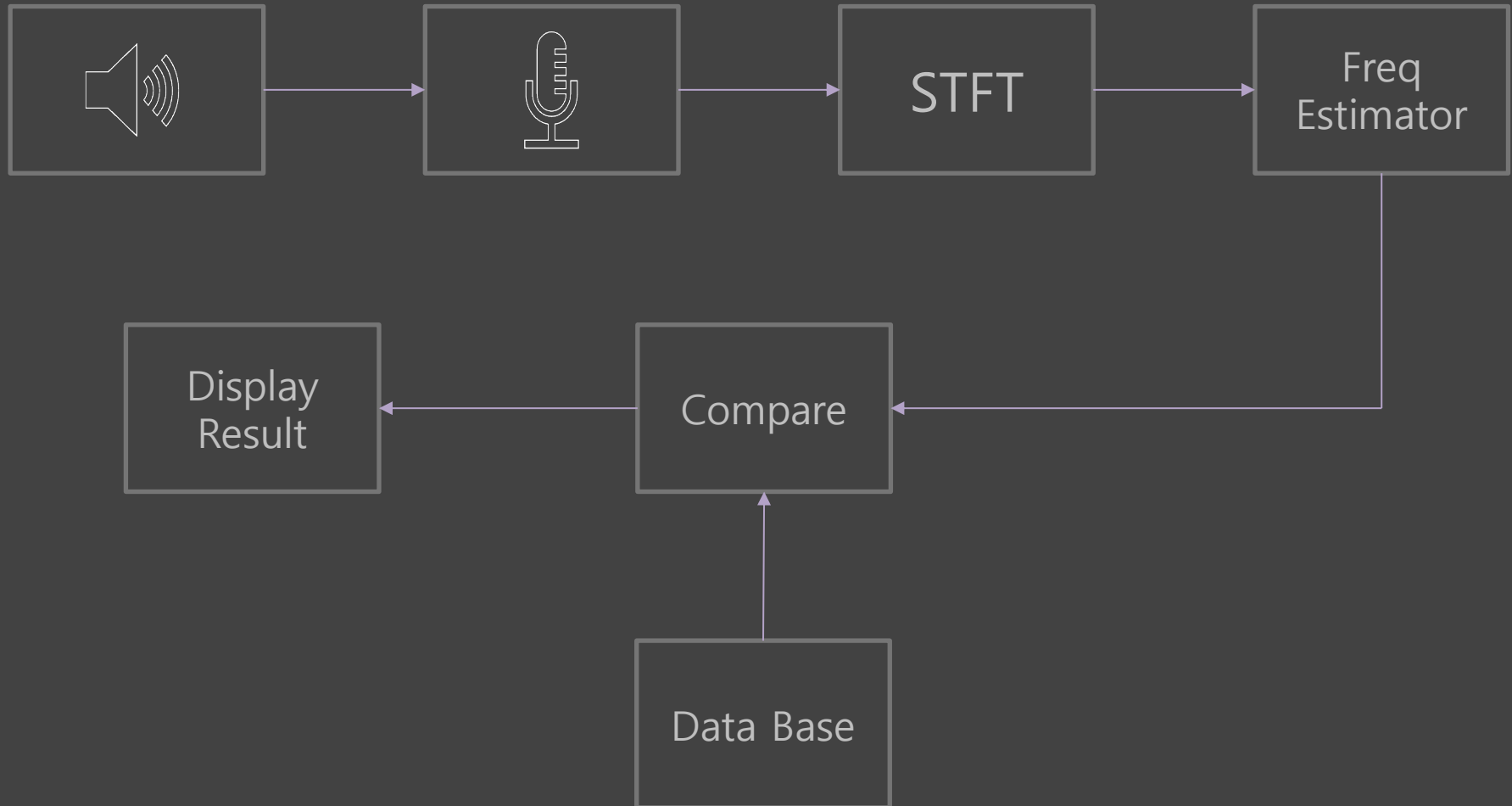


4. Algorithm

Absolute
Pitch

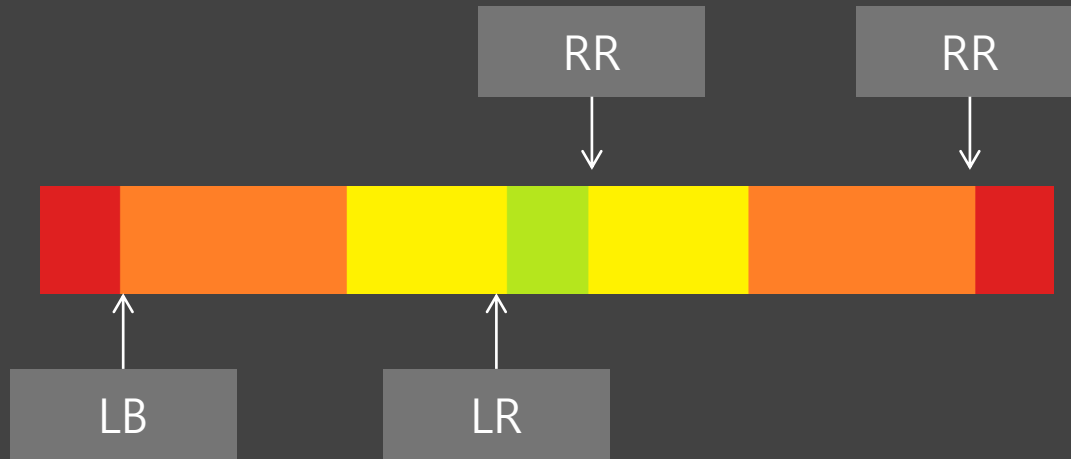
Absolute
Pitch

4. Algorithm – Pitch Training



4. Algorithm – Pitch Training: Display Result

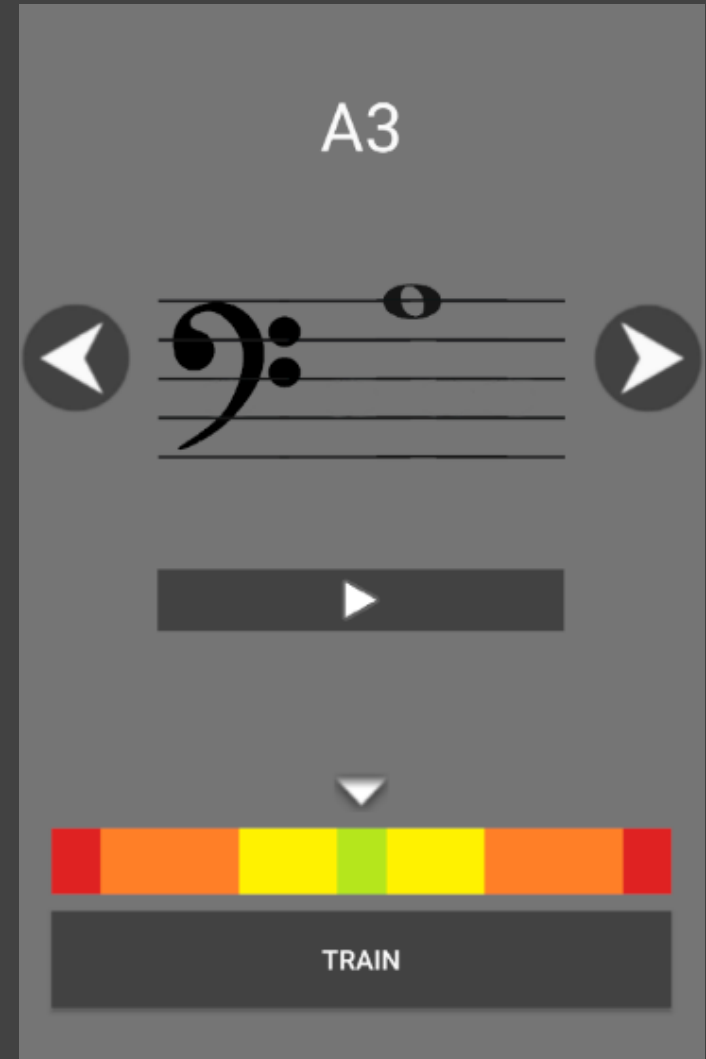
Ex. Receiving A3 (220Hz)



RR: $220 \times (1.05)$, RB: A4 frequency

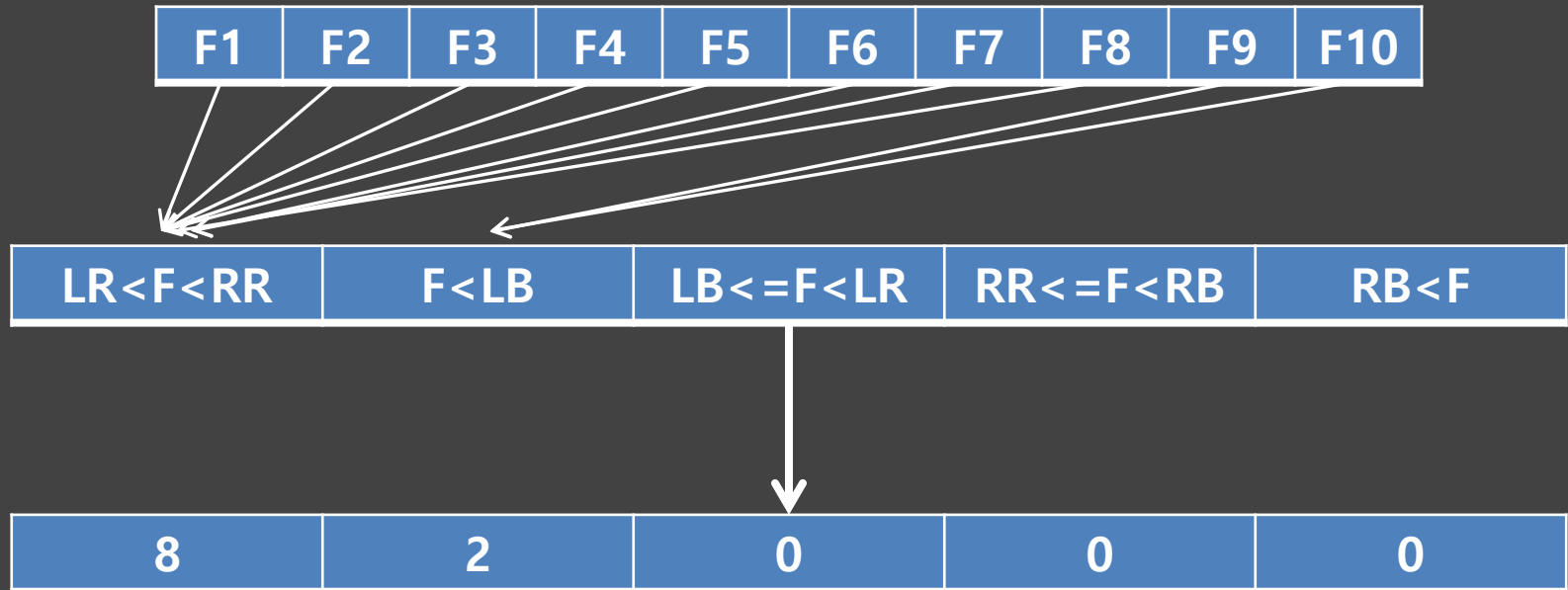
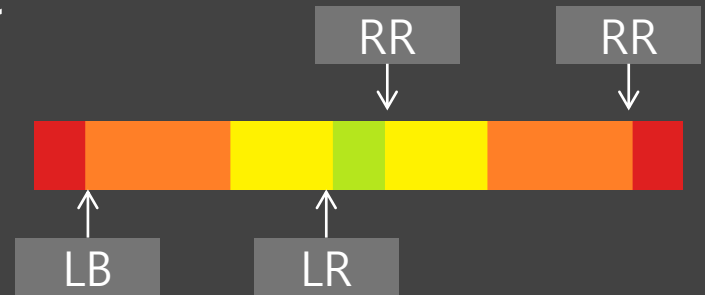
LR: $220 \times (0.95)$, LB: A2 frequency

$[LR < F < RR, F < LB, LB \leq F < LR, RR \leq F < RB, RB < F]$



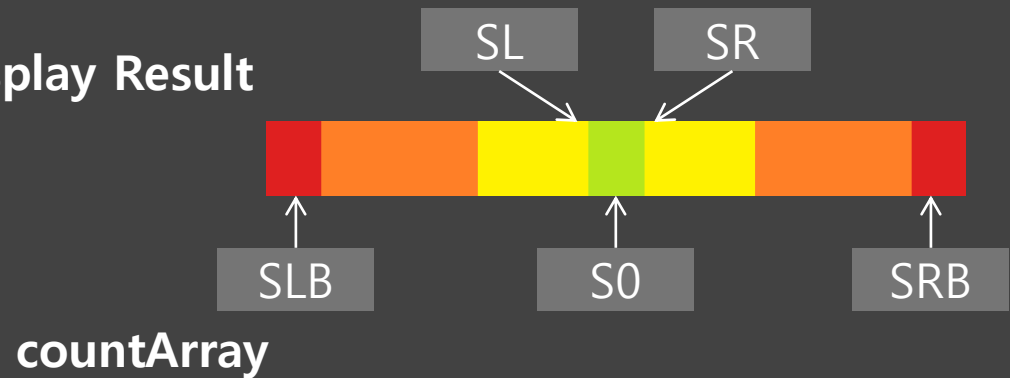
4. Algorithm – Pitch Training: Display Result

Ex. Receiving A3 (220Hz)



4. Algorithm – Pitch Training: Display Result

Ex. Receiving A3 (220Hz)



LR < F < RR [0]

F < LB [1]

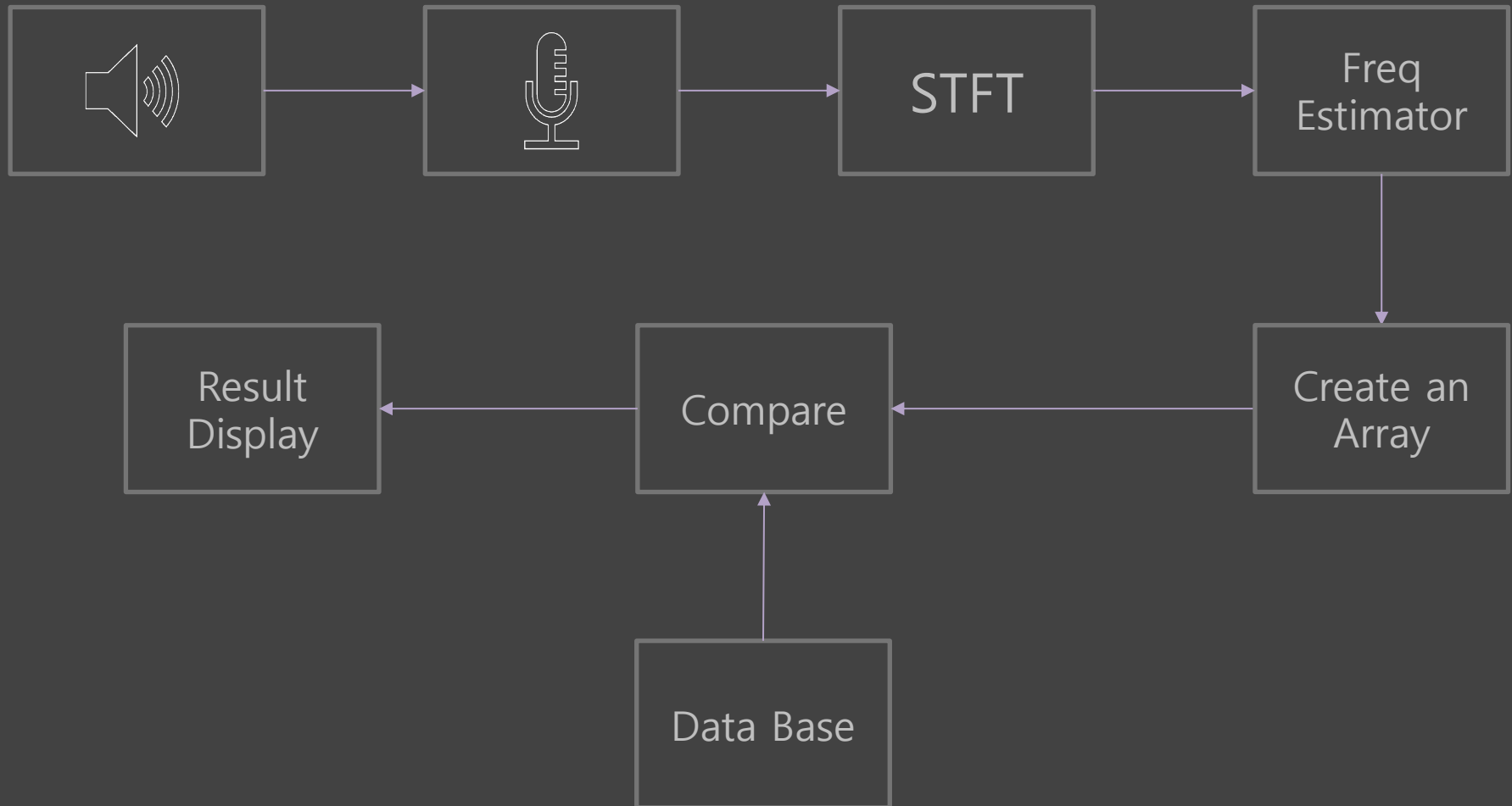
LB ≤ F < LR [2]

RR ≤ F < RB [3]

RB < F [4]

Cases	Arrow position on screen
countArray[0] is the biggest	S0
countArray[1] is the biggest	SLB
countArray[2] is the biggest	$SLB + (Avg - LB) * \{(SL - SLB) / (LR - LB)\}$
countArray[3] is the biggest	$SR + (Avg - RR) * \{(SRB - SR) / (RB - RR)\}$
countArray[4] is the biggest	SRB

4. Algorithm – Vocal Training



4. Algorithm – Vocal Training: Create an Array

Ex. 시간 T당 2개 혹은 3개의 Frequency를 받고,
총 3T의 시간에 Frequency를 받는 경우

Step 1. Frequency를 받는 배열의 최소 크기를 정한다.

시간 T당 2개 혹은 3개의 Frequency를 받으므로, 시간 T당 배정된 배열의 크기를 4(=S)로 한다. 따라서 이 경우 Frequency를 받는 배열의 최소 크기는 12로 한다.

Step 2. Timer Counter를 설정한다.

시간에 따라 배열을 받는 위치를 변화시키기 위한 도구이다.

4. 알고리즘 – Vocal Training: Create an Array

Ex. 시간 T당 2개 혹은 3개의 Frequency를 받고,
총 3T의 시간에 Frequency를 받는 경우

Step 3. Frequency를 받는다. (Array: 배열이 이름, $S=4$, $TS=12$)

0	1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	---	----	----

1. $0T \sim 1T$ 의 시간 동안 2개의 Frequency를 받음 ($TC = 0$)

이때 각 Frequency($F1_1$, $F1_2$)는 $\text{Array}[S*0+0]$, $\text{Array}[S*0+1]$ 에 들어간다.

2. $1T \sim 2T$ 의 시간 동안 3개의 Frequency를 받음 ($TC=1$)

이때 각 Frequency($F2_1$, $F2_2$, $F2_3$)는 $\text{Array}[S*1+0]$, $\text{Array}[S*1+1]$, $\text{Array}[S*1+1]$ 에 들어간다.

3. $2T \sim 3T$ 의 시간 동안 2개의 Frequency를 받음 ($TC=2$)

이때 각 Frequency($F3_1$, $F3_2$)는 $\text{Array}[S*2+0]$, $\text{Array}[S*2+1]$ 에 들어간다.

4. 알고리즘 – Vocal Training: Create an Array

Ex. 시간 T 당 2개 혹은 3개의 Frequency를 받고,
총 $3T$ 의 시간에 Frequency를 받는 경우

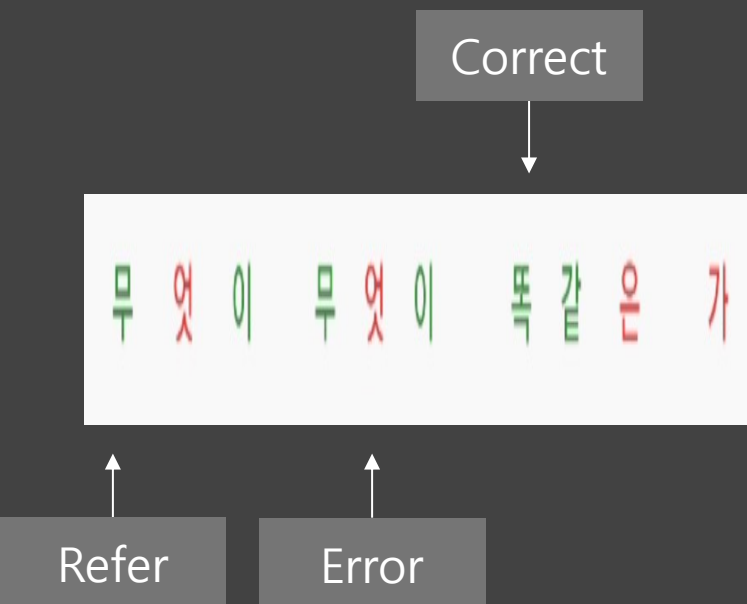
이를 통해 다음과 같은 형태의 Array를 얻을 수 있다.

0~ T 의 음	T ~ $2T$ 의 음	$2T$ ~ $3T$ 의 음
------------	----------------	-----------------

이와 같이 구한 Array에서 $F1_1$, $F1_2$ 을 통해 $0 \sim T$ 에 해당하는 음을, $F2_1$, $F2_2$, $F2_3$ 를 통해 $T \sim 2T$ 에 해당하는 음을, $F3_1$, $F3_2$ 를 통해 $2T \sim 3T$ 에 해당하는 음을 잡아낼 수 있다.

따라서 이와 같이 Frequency의 배열을 만들면 각 박자에 해당하는 음을 확인하기 쉬워지고, 이후 어플리케이션에 저장된 원곡과 비교 과정 또한 용이해진다.

4. 알고리즘 – Vocal Training: Compare& Display Result



똑같아요

윤석중 요

1 2 4 1 2 4 5 5 5 4

무엇이 무엇이 똑같은가

START

Refer

: 첫번째 받는 음을 Reference로 지정

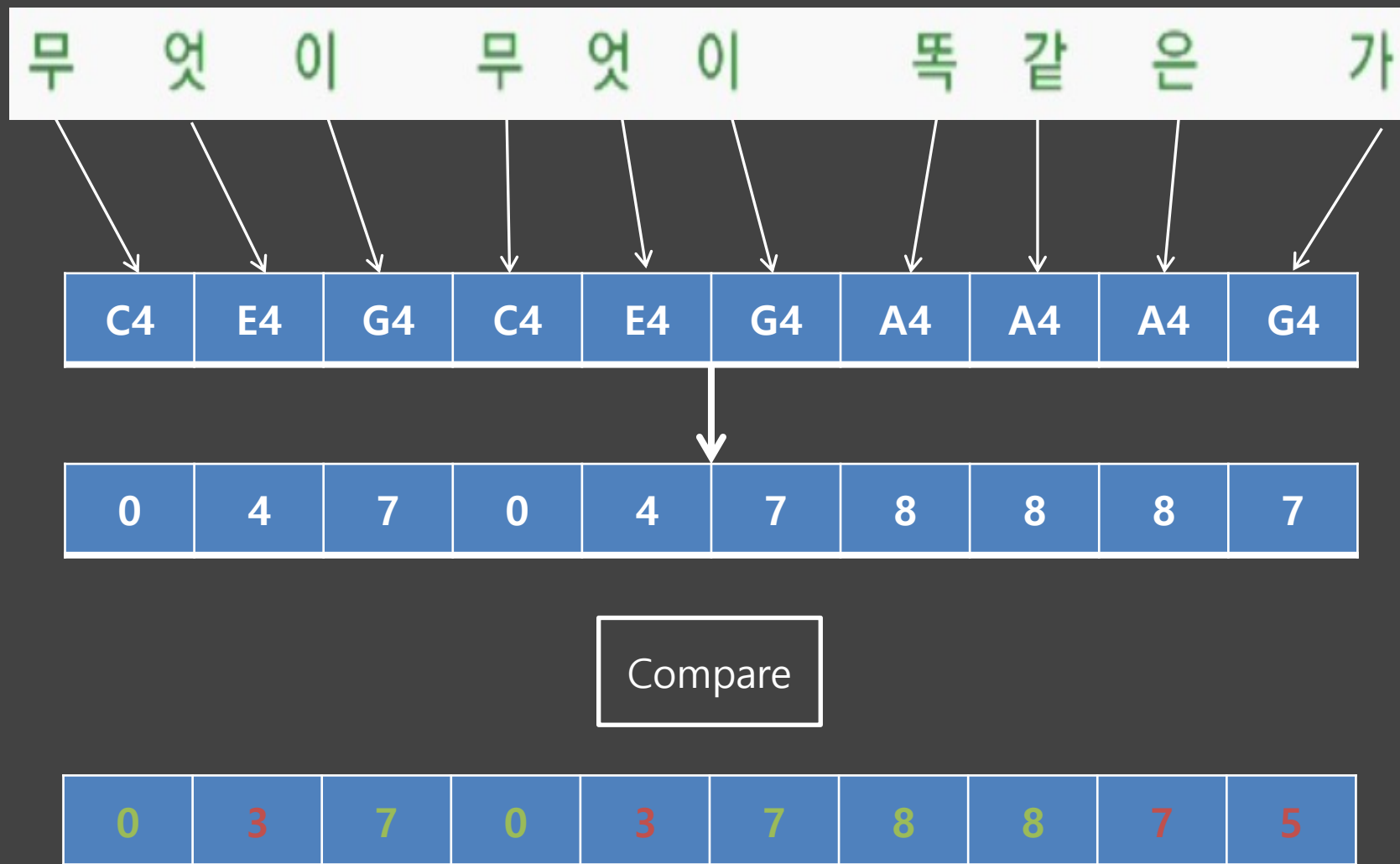
Error

: 저장된 데이터 음과 받은 음의 주파수를 비교 시 틀린 경우

Correct

: 저장된 데이터 음과 받은 음의 주파수를 비교 시 맞는 경우

4. 알고리즘 – Vocal Training: Compare& Display Result



5. Demo

Absolute
Pitch

Absolute
Pitch

6. Conclusion

Absolute
Pitch

Absolute
Pitch

6. Conclusion – Objective

Absolute Pitch

Be able to practice an
absolute pitch via
Absolute Pitch app



6. Conclusion – Objective

Absolute Pitch

Be able to practice
accurate intervals of
songs via **Absolute Pitch**
app



Thank You



CREA!