

Algoritmos de acompanhamento musical para performances polifônicas

Roberto Piassi Passos Bodo

Instituto de Matemática e Estatística
Universidade de São Paulo

24 de Setembro de 2013

"Real-Time Computer Accompaniment of Keyboard Performances"

International Computer Music Conference de 1985

Joshua J. Bloch e Roger B. Dannenberg

Carnegie Mellon University

O Problema

Acompanhamento musical:

O Problema

Acompanhamento musical:

- escutar a performance de um músico;

O Problema

Acompanhamento musical:

- escutar a performance de um músico;
- comparar os eventos da entrada com os eventos de uma partitura;

O Problema

Acompanhamento musical:

- escutar a performance de um músico;
- comparar os eventos da entrada com os eventos de uma partitura;
- inferir um andamento;

O Problema

Acompanhamento musical:

- escutar a performance de um músico;
- comparar os eventos da entrada com os eventos de uma partitura;
- inferir um andamento;
- tocar o acompanhamento apropriado.

"An On-Line Algorithm for Real-Time Accompaniment"

Roger B. Dannenberg

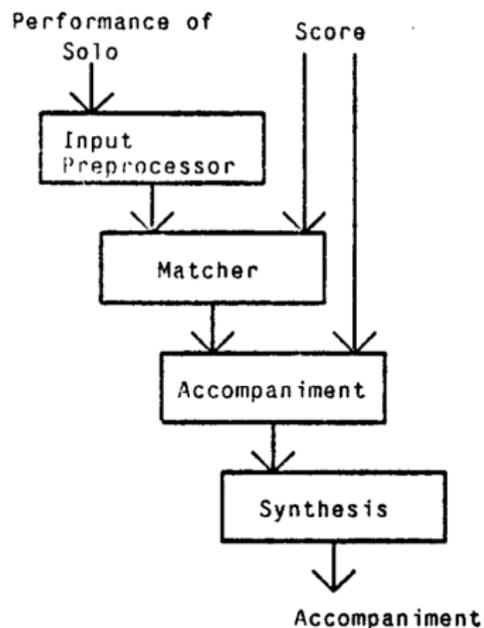
ICMC 1984

"The synthetic performer in the context of live performance"

Barry Vercoe

ICMC 1984

Estrutura de um sistema de acompanhamento



Casamento monofônico

Características dos eventos:

Casamento monofônico

Características dos eventos:

- contêm informação sobre a altura musical somente;

Casamento monofônico

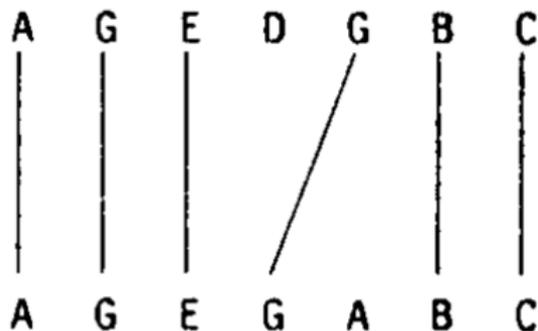
Características dos eventos:

- contêm informação sobre a altura musical somente;
- são trivialmente comparados;

Casamento monofônico

Características dos eventos:

- contêm informação sobre a altura musical somente;
- são trivialmente comparados;
- são totalmente ordenados.



Algoritmo de casamento monofônico

Queremos encontrar o *melhor casamento** entre a entrada do músico e a partitura.

Algoritmo de casamento monofônico

Queremos encontrar o *melhor casamento** entre a entrada do músico e a partitura.

performance:	A	G	E	D	G	B	C
				/			
best match:	A	G	E	G	B	C	
				/			
score:	A	G	E	G	A	B	C

Algoritmo de casamento monofônico

Queremos encontrar o *melhor casamento** entre a entrada do músico e a partitura.

performance:	A	G	E	D	G	B	C
				/			
best match:	A	G	E	G	B	C	
				/			
score:	A	G	E	G	A	B	C

**melhor casamento* = maior subsequência comum.

Algoritmo de casamento monofônico

Adaptações do algoritmo de programação dinâmica que calcula maior subsequência comum:

performance:

	A	G	E	D	G	B	C
score: A	1	1	1	1			
G	1	2	2	2			
E	1	2	3	3			
G	1	2	3	3			
A	1	2	3	3			
B	1	2	3	3			
C	1	2	3	3			

Algoritmo de casamento monofônico

Adaptações do algoritmo de programação dinâmica que calcula maior subsequência comum:

performance:

	A	G	E	D	G	B	C
score: A	1	1	1	1			
G	1	2	2	2			
E	1	2	3	3			
G	1	2	3	3			
A	1	2	3	3			
B	1	2	3	3			
C	1	2	3	3			

- cada linha corresponde a um evento da partitura;

Algoritmo de casamento monofônico

Adaptações do algoritmo de programação dinâmica que calcula maior subsequência comum:

		performance:						
		A	G	E	D	G	B	C
score:	A	1	1	1	1			
	G	1	2	2	2			
	E	1	2	3	3			
	G	1	2	3	3			
	A	1	2	3	3			
	B	1	2	3	3			
	C	1	2	3	3			

- cada linha corresponde a um evento da partitura;
- cada coluna corresponde a um evento da performance;

Algoritmo de casamento monofônico

Adaptações do algoritmo de programação dinâmica que calcula maior subsequência comum:

		performance:						
		A	G	E	D	G	B	C
score:	A	1	1	1	1			
	G	1	2	2	2			
	E	1	2	3	3			
	G	1	2	3	3			
	A	1	2	3	3			
	B	1	2	3	3			
	C	1	2	3	3			

- cada linha corresponde a um evento da partitura;
- cada coluna corresponde a um evento da performance;
- a cada evento detectado uma nova coluna é calculada.

Algoritmo de casamento monofônico

Valor da i -ésima linha e j -ésima coluna: a maior pontuação de qualquer associação até o momento atual.

Algoritmo de casamento monofônico

Valor da i -ésima linha e j -ésima coluna: a maior pontuação de qualquer associação até o momento atual.

Quais eventos foram associados para obter tal pontuação?

Algoritmo de casamento monofônico

Valor da i -ésima linha e j -ésima coluna: a maior pontuação de qualquer associação até o momento atual.

Quais eventos foram associados para obter tal pontuação?

performance:

	A	G	E	D	G	B	C
score:	A	<u>1</u>	1	1	1	1	1
G	1	<u>2</u>	2	2	2	2	2
E	1	2	<u>3</u>	3	3	3	3
G	1	2	3	3	<u>4</u>	4	4
A	1	2	3	3	4	<u>4</u>	4
B	1	2	3	3	4	<u>5</u>	5
C	1	2	3	3	4	5	<u>6</u>

Casamento polifônico

Características dos eventos:

Casamento polifônico

Características dos eventos:

- contêm informação temporal (além da altura musical);

Casamento polifônico

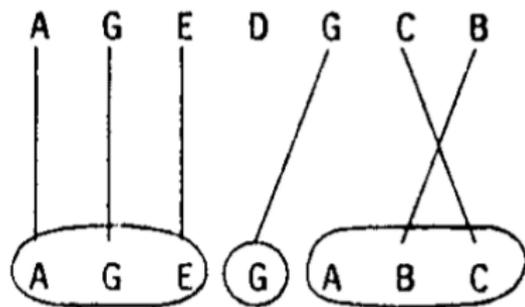
Características dos eventos:

- contêm informação temporal (além da altura musical);
- são mais difíceis de comparar;

Casamento polifônico

Características dos eventos:

- contêm informação temporal (além da altura musical);
- são mais difíceis de comparar;
- são parcialmente ordenados.



Casamento polifônico

Partitura: sequência de conjuntos de símbolos não-ordenados.

$$CEG \left\{ \begin{array}{l} GEC \\ GCE \\ CEG \\ CGE \\ EGC \\ ECG \end{array} \right.$$

Casamento polifônico

Decisões de projeto de algoritmos:

Casamento polifônico

Decisões de projeto de algoritmos:

- 1 definir o significado de melhor associação entre performance e partitura;

Casamento polifônico

Decisões de projeto de algoritmos:

- 1 definir o significado de melhor associação entre performance e partitura;
- 2 decidir com confiança em qual ponto da partitura o músico está;

Casamento polifônico

Decisões de projeto de algoritmos:

- 1 definir o significado de melhor associação entre performance e partitura;
- 2 decidir com confiança em qual ponto da partitura o músico está;
- 3 agrupar as notas da performance em eventos compostos.

Associação entre performance e partitura

Definição de melhor associação?

Associação entre performance e partitura

Definição de melhor associação?

Aquela que maximiza uma função de pontuação.

Associação entre performance e partitura

Definição de melhor associação?

Aquela que maximiza uma função de pontuação.

Para que queremos encontrá-la?

Associação entre performance e partitura

Definição de melhor associação?

Aquela que maximiza uma função de pontuação.

Para que queremos encontrá-la?

Determinar o melhor prefixo da partitura para um dado prefixo da performance.

Associação entre performance e partitura

Definição de melhor associação?

Aquela que maximiza uma função de pontuação.

Para que queremos encontrá-la?

Determinar o melhor prefixo da partitura para um dado prefixo da performance.

Notação:

Associação entre performance e partitura

Definição de melhor associação?

Aquela que maximiza uma função de pontuação.

Para que queremos encontrá-la?

Determinar o melhor prefixo da partitura para um dado prefixo da performance.

Notação:

- $p[i]$ = i-ésimo evento da performance

Associação entre performance e partitura

Definição de melhor associação?

Aquela que maximiza uma função de pontuação.

Para que queremos encontrá-la?

Determinar o melhor prefixo da partitura para um dado prefixo da performance.

Notação:

- $p[i]$ = i -ésimo evento da performance
- $p[1:i]$ = prefixo da performance de tamanho i

Associação entre performance e partitura

Definição de melhor associação?

Aquela que maximiza uma função de pontuação.

Para que queremos encontrá-la?

Determinar o melhor prefixo da partitura para um dado prefixo da performance.

Notação:

- $p[i]$ = i -ésimo evento da performance
- $p[1:i]$ = prefixo da performance de tamanho i
- $s[j]$ = j -ésimo evento da partitura

Associação entre performance e partitura

Definição de melhor associação?

Aquela que maximiza uma função de pontuação.

Para que queremos encontrá-la?

Determinar o melhor prefixo da partitura para um dado prefixo da performance.

Notação:

- $p[i]$ = i -ésimo evento da performance
- $p[1:i]$ = prefixo da performance de tamanho i
- $s[j]$ = j -ésimo evento da partitura
- $s[1:j]$ = prefixo da partitura de tamanho j

Associação entre performance e partitura

Vamos definir a função *best-match-rating*($p[1:i]$, s).

Associação entre performance e partitura

Vamos definir a função *best-match-rating*($p[1:i]$, s).

Cada símbolo $p[i]$ é classificado como:

Associação entre performance e partitura

Vamos definir a função *best-match-rating*($p[1:i]$, s).

Cada símbolo $p[i]$ é classificado como:

- certo;

Associação entre performance e partitura

Vamos definir a função *best-match-rating*($p[1:i]$, s).

Cada símbolo $p[i]$ é classificado como:

- certo;
- errado;

Associação entre performance e partitura

Vamos definir a função *best-match-rating*($p[1:i]$, s).

Cada símbolo $p[i]$ é classificado como:

- certo;
- errado;
- adicional;

Associação entre performance e partitura

Vamos definir a função *best-match-rating*($p[1:i]$, s).

Cada símbolo $p[i]$ é classificado como:

- certo;
- errado;
- adicional;
- faltando.

Associação entre performance e partitura

Vamos definir a função *best-match-rating*($p[1:i]$, s).

Cada símbolo $p[i]$ é classificado como:

- certo;
- errado;
- adicional;
- faltando.

Classe de funções:

Associação entre performance e partitura

Vamos definir a função *best-match-rating*($p[1:i]$, s).

Cada símbolo $p[i]$ é classificado como:

- certo;
- errado;
- adicional;
- faltando.

Classe de funções:

= tamanho do prefixo da partitura - $C_{erradas}$ #notas erradas

- $C_{adicionais}$ #notas adicionais - $C_{faltando}$ #notas faltando

Em qual ponto da partitura o músico está

Primeira abordagem: assumir que a função *best-match-rating* tende a crescer com a evolução da performance e tende a diminuir com a execução de erros.

Em qual ponto da partitura o músico está

Primeira abordagem: assumir que a função *best-match-rating* tende a crescer com a evolução da performance e tende a diminuir com a execução de erros.

Quando reportamos um casamento?

Em qual ponto da partitura o músico está

Primeira abordagem: assumir que a função *best-match-rating* tende a crescer com a evolução da performance e tende a diminuir com a execução de erros.

Quando reportamos um casamento?

Quando *best-match-rating* é maior do que qualquer outro valor prévio.

Em qual ponto da partitura o músico está

Primeira abordagem: assumir que a função *best-match-rating* tende a crescer com a evolução da performance e tende a diminuir com a execução de erros.

Quando reportamos um casamento?

Quando *best-match-rating* é maior do que qualquer outro valor prévio.

Propriedade?

Em qual ponto da partitura o músico está

Primeira abordagem: assumir que a função *best-match-rating* tende a crescer com a evolução da performance e tende a diminuir com a execução de erros.

Quando reportamos um casamento?

Quando *best-match-rating* é maior do que qualquer outro valor prévio.

Propriedade?

Mais cautelosa após uma série de erros.

Em qual ponto da partitura o músico está

Segunda abordagem: se assegurar que a última nota tocada na performance é consistente com o local mais provável da partitura.

Em qual ponto da partitura o músico está

Segunda abordagem: se assegurar que a última nota tocada na performance é consistente com o local mais provável da partitura.

Quando reportamos um casamento?

Em qual ponto da partitura o músico está

Segunda abordagem: se assegurar que a última nota tocada na performance é consistente com o local mais provável da partitura.

Quando reportamos um casamento?

Quando $best-match-rating(p[1:i], s)$ é maior do que $best-match-rating(p[1:i-1], s)$.

Em qual ponto da partitura o músico está

Segunda abordagem: se assegurar que a última nota tocada na performance é consistente com o local mais provável da partitura.

Quando reportamos um casamento?

Quando $best-match-rating(p[1:i], s)$ é maior do que $best-match-rating(p[1:i-1], s)$.

Propriedades?

Em qual ponto da partitura o músico está

Segunda abordagem: se assegurar que a última nota tocada na performance é consistente com o local mais provável da partitura.

Quando reportamos um casamento?

Quando $best-match-rating(p[1:i], s)$ é maior do que $best-match-rating(p[1:i-1], s)$.

Propriedades?

Recupera confiança rapidamente após uma série de erros.

Algoritmo estático de casamento

Ideia básica: reduzir o problema do acompanhamento polifônico para o monofônico.

Algoritmo estático de casamento

Ideia básica: reduzir o problema do acompanhamento polifônico para o monofônico.

Implicação: transformar a partitura e a performance em sequências totalmente ordenadas de eventos compostos.

Algoritmo estático de casamento

Ideia básica: reduzir o problema do acompanhamento polifônico para o monofônico.

Implicação: transformar a partitura e a performance em sequências totalmente ordenadas de eventos compostos.

Evento composto = conjunto de notas tocadas simultaneamente.

Algoritmo estático de casamento

Na partitura: aparentemente trivial.



Algoritmo estático de casamento

Na partitura: aparentemente trivial.



Na performance: nem tanto.

1, **160**, *Note_on_c*, 1, 60, 127

1, **172**, *Note_on_c*, 1, 64, 127

1, **185**, *Note_on_c*, 1, 67, 127

Algoritmo estático de casamento

- Valor máximo observado em notas de um mesmo acorde: 90ms

Algoritmo estático de casamento

- Valor máximo observado em notas de um mesmo acorde: 90ms
- Limite superior de tempo entre notas consecutivas: 125ms

Algoritmo estático de casamento

- Valor máximo observado em notas de um mesmo acorde: 90ms
- Limite superior de tempo entre notas consecutivas: 125ms
Semicolcheia em 120bpm

Algoritmo estático de casamento

- Valor máximo observado em notas de um mesmo acorde: 90ms
- Limite superior de tempo entre notas consecutivas: 125ms
Semicolcheia em 120bpm
⇒ 8 notas por segundo

Algoritmo estático de casamento

- Valor máximo observado em notas de um mesmo acorde: 90ms
- Limite superior de tempo entre notas consecutivas: 125ms
Semicolcheia em 120bpm
 - ⇒ 8 notas por segundo
 - ⇒ 125ms

Algoritmo estático de casamento

- Valor máximo observado em notas de um mesmo acorde: 90ms
- Limite superior de tempo entre notas consecutivas: 125ms
Semicolcheia em 120bpm
⇒ 8 notas por segundo
⇒ 125ms
- Valor arbitrariamente escolhido para o algoritmo: 100ms (**epsilon**)

Algoritmo estático de casamento

Algoritmo

Para cada nota recebida da performance:

Algoritmo estático de casamento

Algoritmo

Para cada nota recebida da performance:

- se o intervalo de tempo em relação à nota anterior for menor do que **epsilon**, adicionamos no mesmo evento composto;

Algoritmo estático de casamento

Algoritmo

Para cada nota recebida da performance:

- se o intervalo de tempo em relação à nota anterior for menor do que **epsilon**, adicionamos no mesmo evento composto;
- caso contrário, criamos um novo evento composto.

Algoritmo estático de casamento

O algoritmo não é livre de erros.

Algoritmo estático de casamento

O algoritmo não é livre de erros.

Principal problema:

Algoritmo estático de casamento

O algoritmo não é livre de erros.

Principal problema:

- acorde em arpejo;

Algoritmo estático de casamento

O algoritmo não é livre de erros.

Principal problema:

- acorde em arpejo;
O intervalo entre notas é facilmente maior que **epsilon**.

Algoritmo estático de casamento

O algoritmo não é livre de erros.

Principal problema:

- acorde em arpejo;
O intervalo entre notas é facilmente maior que **epsilon**.

Solução: teste **epsilon-fraction**.

$$C \underbrace{\quad E}_{1/4} \underbrace{\quad A\#}_{3/4}$$

Algoritmo estático de casamento

Quando comparar (isto é, calcular a função de pontuação entre) eventos compostos em construção da performance com os eventos da partitura?

Algoritmo estático de casamento

Quando comparar (isto é, calcular a função de pontuação entre) eventos compostos em construção da performance com os eventos da partitura?

Caso monofônico:

Algoritmo estático de casamento

Quando comparar (isto é, calcular a função de pontuação entre) eventos compostos em construção da performance com os eventos da partitura?

Caso monofônico:

- calculamos cada vez que um novo evento solo é processado.

Algoritmo estático de casamento

Quando comparar (isto é, calcular a função de pontuação entre) eventos compostos em construção da performance com os eventos da partitura?

Caso monofônico:

- calculamos cada vez que um novo evento solo é processado.

Caso polifônico:

Algoritmo estático de casamento

Quando comparar (isto é, calcular a função de pontuação entre) eventos compostos em construção da performance com os eventos da partitura?

Caso monofônico:

- calculamos cada vez que um novo evento solo é processado.

Caso polifônico:

- calculamos quando um novo evento solo é processado;

Algoritmo estático de casamento

Quando comparar (isto é, calcular a função de pontuação entre) eventos compostos em construção da performance com os eventos da partitura?

Caso monofônico:

- calculamos cada vez que um novo evento solo é processado.

Caso polifônico:

- calculamos quando um novo evento solo é processado;
- atualizamos quando agrupamos uma nova nota.

Algoritmo estático de casamento

Qual função de pontuação deve ser utilizada?

Algoritmo estático de casamento

Qual função de pontuação deve ser utilizada?

cevt-match-rating

Algoritmo estático de casamento

Qual função de pontuação deve ser utilizada?

cevt-match-rating =

(#notas da performance presentes no evento composto da partitura -
#notas da performance ausentes no evento composto da partitura) /
#notas da performance

Algoritmo estático de casamento

Quando reportamos um casamento entre eventos compostos?

Algoritmo estático de casamento

Quando reportamos um casamento entre eventos compostos?

cevt-match

Algoritmo estático de casamento

Quando reportamos um casamento entre eventos compostos?

$$cevt-match = \begin{cases} \text{true, se } cevt-match-rating > n \\ \text{false, c.c.} \end{cases}$$

Algoritmo estático de casamento

Quando reportamos um casamento entre eventos compostos?

$$cevt-match = \begin{cases} \text{true, se } cevt-match-rating > n \\ \text{false, c.c.} \end{cases}$$

n é um valor limite escolhido empiricamente.

Algoritmo estático de casamento

Quando reportamos um casamento entre eventos compostos?

$$cevt-match = \begin{cases} \text{true, se } cevt-match-rating > n \\ \text{false, c.c.} \end{cases}$$

n é um valor limite escolhido empiricamente.

$n = 0.5$ (1 nota errada em um acorde de 4 notas).

Algoritmo dinâmico de casamento

Algoritmo dinâmico de casamento

- similar ao algoritmo monofônico de matching;

Algoritmo dinâmico de casamento

- similar ao algoritmo monofônico de matching;
- sequência de símbolos vs. sequência de conjuntos de símbolos;

Algoritmo dinâmico de casamento

- similar ao algoritmo monofônico de matching;
- sequência de símbolos vs. sequência de conjuntos de símbolos;
- maximizar uma função de pontuação.

Algoritmo dinâmico de casamento

- similar ao algoritmo monofônico de matching;
- sequência de símbolos vs. sequência de conjuntos de símbolos;
- maximizar uma função de pontuação.

		Performed Events			
		a_1	a_2	a_3	...
Score Sets	A_1				
	A_2				
	A_3				
	⋮				⋮

Algoritmo dinâmico de casamento

Em cada posição da matriz:

$\text{matrix}[r, c].\text{value}$ = valor da função de pontuação para a melhor associação até o instante

$\text{matrix}[r, c].\text{used}$ = conjunto de símbolos utilizados para obter essa pontuação

Algoritmo dinâmico de casamento

Algoritmo dinâmico de casamento

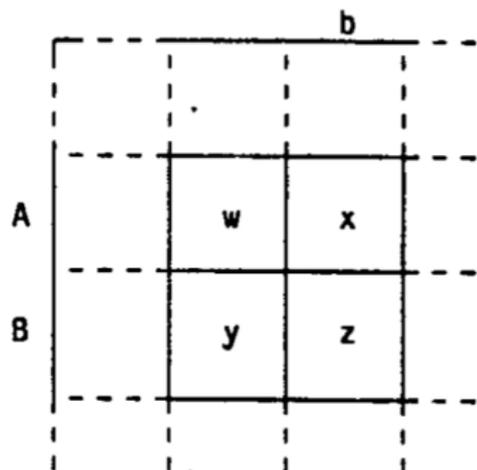
- cada coluna é calculada quando uma nota é recebida;

Algoritmo dinâmico de casamento

- cada coluna é calculada quando uma nota é recebida;
- $matrix[r, c]$ é calculado a partir de
 - ▶ $matrix[r, c - 1]$;
 - ▶ $matrix[r - 1, c]$;
 - ▶ $matrix[r - 1, c - 1]$;

Algoritmo dinâmico de casamento

- cada coluna é calculada quando uma nota é recebida;
- $\text{matrix}[r, c]$ é calculado a partir de
 - ▶ $\text{matrix}[r, c - 1]$;
 - ▶ $\text{matrix}[r - 1, c]$;
 - ▶ $\text{matrix}[r - 1, c - 1]$;



Algoritmo dinâmico de casamento

```
1)  $z.value := y.value$ ;  
2)  $z.used := y.used$ ;  
3) if  $b$  in  $B$  then  
4)   if  $b$  not in  $y.used$  then  
5)      $z.value := y.value + 1$ ;  
6)      $z.used := y.used \cup \{ b \}$   
7)   endif;  
8)    $d := w.value + 1 - \#(A - w.used)$ ;  
9)   if  $d \geq z.value$  then  
10)     $z.value := d$ ;  
11)     $z.used := \{ b \}$   
12)   endif  
13) endif;  
14)  $v := x.value - \#(A - x.used)$ ;  
15) if  $v \geq z.value$  then  
16)    $z.value := v$ ;  
17)    $z.used := \{ \}$ ;  
18) endif
```

Algoritmo dinâmico de casamento

```
1)  $z.value := y.value;$   
2)  $z.used := y.used;$   
3) if  $b$  in  $B$  then  
4)   if  $b$  not in  $y.used$  then  
5)      $z.value := y.value + 1;$   
6)      $z.used := y.used \cup \{ b \}$   
7)   endif;  
8)    $d := w.value + 1 - \#(A - w.used);$   
9)   if  $d \geq z.value$  then  
10)     $z.value := d;$   
11)     $z.used := \{ b \}$   
12)   endif  
13) endif;  
14)  $v := x.value - \#(A - x.used);$   
15) if  $v \geq z.value$  then  
16)    $z.value := v;$   
17)    $z.used := \{ \};$   
18) endif
```

Algoritmo dinâmico de casamento

$$1^{\text{a}} \text{ atribuição: } \begin{cases} z.\text{value} = y.\text{value} \\ z.\text{used} = y.\text{used} \end{cases}$$

Algoritmo dinâmico de casamento

$$1^{\text{a}} \text{ atribuição: } \begin{cases} z.value = y.value \\ z.used = y.used \end{cases}$$

- b não casa com nada;

Algoritmo dinâmico de casamento

$$1^{\text{a}} \text{ atribuição: } \begin{cases} z.value = y.value \\ z.used = y.used \end{cases}$$

- b não casa com nada;
- as notas antes de b são casadas assim como em y.

Algoritmo dinâmico de casamento

$$2^{\text{a}} \text{ atribuição: } \begin{cases} z.\text{value} = y.\text{value} + 1 \\ z.\text{used} = y.\text{used} \cup \{b\} \end{cases}$$

Algoritmo dinâmico de casamento

$$2^{\text{a}} \text{ atribuição: } \begin{cases} z.\text{value} = y.\text{value} + 1 \\ z.\text{used} = y.\text{used} \cup \{b\} \end{cases}$$

- b pertence a B ;

Algoritmo dinâmico de casamento

$$2^{\text{a}} \text{ atribuição: } \begin{cases} z.\text{value} = y.\text{value} + 1 \\ z.\text{used} = y.\text{used} \cup \{b\} \end{cases}$$

- b pertence a B ;
- b não pertence ao conjunto das notas já utilizadas;

Algoritmo dinâmico de casamento

$$2^{\text{a}} \text{ atribuição: } \begin{cases} z.\text{value} = y.\text{value} + 1 \\ z.\text{used} = y.\text{used} \cup \{b\} \end{cases}$$

- b pertence a B ;
- b não pertence ao conjunto das notas já utilizadas;
- as notas são casadas assim como em y .

Algoritmo dinâmico de casamento

$$3^{\text{a}} \text{ atribuição: } \begin{cases} z.value = w.value + 1 - \#(A - w.used) \\ z.used = \{b\} \end{cases}$$

Algoritmo dinâmico de casamento

$$3^{\text{a}} \text{ atribuição: } \begin{cases} z.\text{value} = w.\text{value} + 1 - \#(A - w.\text{used}) \\ z.\text{used} = \{b\} \end{cases}$$

- b pertence a B ;

Algoritmo dinâmico de casamento

$$3^{\text{a}} \text{ atribuição: } \begin{cases} z.\text{value} = w.\text{value} + 1 - \#(A - w.\text{used}) \\ z.\text{used} = \{b\} \end{cases}$$

- b pertence a B ;
- as notas são casadas assim como em w .

Algoritmo dinâmico de casamento

$$4^{\text{a}} \text{ atribuição: } \begin{cases} z.\text{value} = x.\text{value} - \#(A - x.\text{used}) \\ z.\text{used} = \emptyset \end{cases}$$

Algoritmo dinâmico de casamento

$$4^{\text{a}} \text{ atribuição: } \begin{cases} z.\text{value} = x.\text{value} - \#(A - x.\text{used}) \\ z.\text{used} = \emptyset \end{cases}$$

- b não casa com nada;

Algoritmo dinâmico de casamento

$$4^{\text{a}} \text{ atribuição: } \begin{cases} z.\text{value} = x.\text{value} - \#(A - x.\text{used}) \\ z.\text{used} = \emptyset \end{cases}$$

- b não casa com nada;
- as notas são casadas assim como em x.

Resultados experimentais

Resultados experimentais

- foram implementados 2 sistemas modularizados: estático e dinâmico;

Resultados experimentais

- foram implementados 2 sistemas modularizados: estático e dinâmico;
- não ficou claro qual dos dois é melhor;

Resultados experimentais

- foram implementados 2 sistemas modularizados: estático e dinâmico;
- não ficou claro qual dos dois é melhor;
- ambos ficaram lentos para performances mais complexas;

Resultados experimentais

- foram implementados 2 sistemas modularizados: estático e dinâmico;
- não ficou claro qual dos dois é melhor;
- ambos ficaram lentos para performances mais complexas;
- são adequados para performances ao vivo.

Resultados experimentais

Resultados experimentais

Algoritmo estático:

Resultados experimentais

Algoritmo estático:

- mais suscetível a erros quando o andamento é muito diferente do original;

Resultados experimentais

Algoritmo estático:

- mais suscetível a erros quando o andamento é muito diferente do original;
- lida com trinados e ornamentos.

Resultados experimentais

Algoritmo estático:

- mais suscetível a erros quando o andamento é muito diferente do original;
- lida com trinados e ornamentos.

Algoritmo dinâmico:

Resultados experimentais

Algoritmo estático:

- mais suscetível a erros quando o andamento é muito diferente do original;
- lida com trinados e ornamentos.

Algoritmo dinâmico:

- nem sempre modela o conceito de um bom casamento para um músico;

Resultados experimentais

Algoritmo estático:

- mais suscetível a erros quando o andamento é muito diferente do original;
- lida com trinados e ornamentos.

Algoritmo dinâmico:

- nem sempre modela o conceito de um bom casamento para um músico;
- precisa de ajuda do *InputProcessor* para tratar trinados e ornamentos.

That's all folks!

The End! =)