ACM Transactions on Information Systems (January 1996)

An Extension of Ukkonen's Enhanced Dynamic Programming ASM Algorithm

Hal Berghel
University of Arkansas
David Roach
Acxiom Corporation

Outline

- Introduction
  - Problem definition
  - Classic dp approach
- Observations
- Algorithms
- Analysis
- Conclusion

Introduction

- Problem definition
- Classic dp approach

Problem Definition

- **Approximate string matching ASM**
  - A class of techniques that associate strings of symbols with one another on the basis of some criterion of similarity. Positional, ordinal, material…
- **Edit distance**
  - the minimum number of basic editing operations that can transform one string into the other

Problem Definition and DP

- Edit operations
  - Deletion
  - Insertion
  - Substitution
  - Transposition

	λ	A	B	C	D	E
λ	0	1	2	3	4	5
X	1	1	2	3	4	5
A	2	1	2	3	4	5
X	3	2	2	3	4	5
E	4	3	3	3	4	4
D	5	4	4	4	3	4

Problem Definition and DP

- Edit operations
  - Deletion
  - Insertion
  - Substitution
  - Transposition

	λ	A	B	C	D	E
λ	**0**	1	2	3	4	5
X	**1**	**1**	2	3	4	5
A	2	**1**	**2**	3	4	5
X	3	2	**2**	**3**	4	5
E	4	3	3	**3**	4	4
D	5	4	4	4	**3**	**4**

Problem Definition and DP

- Edit operations
  - Deletion
  - Insertion
  - Substitution
  - Transposition

	λ	A	B	C	D	E
λ	**0**	1	2	3	4	5
X	**1**	**1**	2	3	4	5
A	2	**1**	**2**	3	4	5
X	3	2	**2**	**3**	4	5
E	4	3	3	**3**	4	4
D	5	4	4	4	**3**	**4**

Problem Definition and DP

- Edit operations
  - Deletion
  - Insertion
  - Substitution
  - Transposition

	λ	A	B	C	D	E
λ	**0**	1	2	3	4	5
X	**1**	**1**	2	3	4	5
A	2	**1**	**2**	3	4	5
X	3	2	**2**	**3**	4	5
E	4	3	3	**3**	4	4
D	5	4	4	4	**3**	**4**

Problem Definition and DP

- Edit operations
  - Deletion
  - Insertion
  - Substitution
  - Transposition

	λ	A	B	C	D	E
λ	**0**	1	2	3	4	5
X	**1**	**1**	2	3	4	5
A	2	**1**	**2**	3	4	5
X	3	2	**2**	**3**	4	5
E	4	3	3	**3**	4	4
D	5	4	4	4	**3**	**4**

Problem Definition and DP

- Edit operations
  - Deletion
  - Insertion
  - Substitution
  - Transposition

	λ	A	B	C	D	E
λ	**0**	1	2	3	4	5
X	**1**	**1**	2	3	4	5
A	2	**1**	**2**	3	4	5
X	3	2	**2**	**3**	4	5
E	4	3	3	**3**	4	4
D	5	4	4	4	**3**	**4**

Problem Definition and DP

- Edit operations
  - Deletion
  - Insertion
  - Substitution
  - Transposition
- 10/36 numbers actually contribute to edit path

	λ	A	B	C	D	E
λ	**0**	1	2	3	4	5
X	**1**	**1**	2	3	4	5
A	2	**1**	**2**	3	4	5
X	3	2	**2**	**3**	4	5
E	4	3	3	**3**	4	4
D	5	4	4	4	**3**	**4**

Observations

Observations

- Diagonal

	λ	A	B	C	D	E
λ						
X						
A						
X						
E						
D						

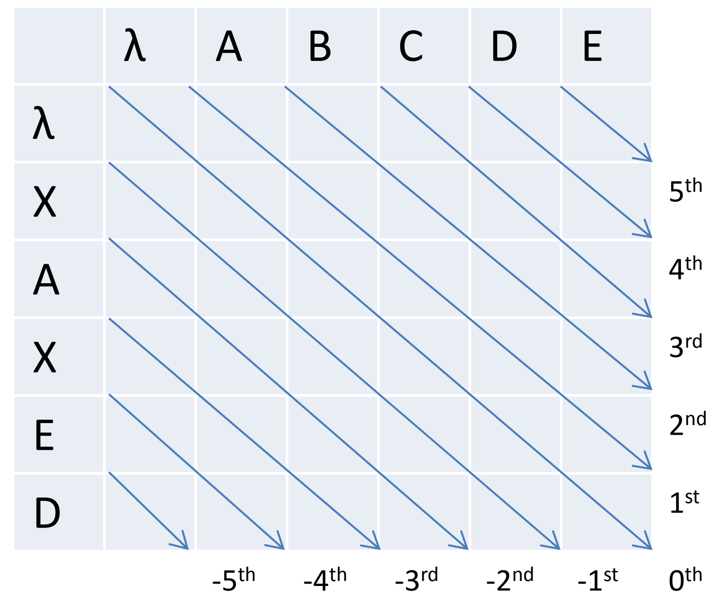
Observations

- Diagonal

	λ	A	B	C	D	E
λ						
X						
A						
X						
E						
D						

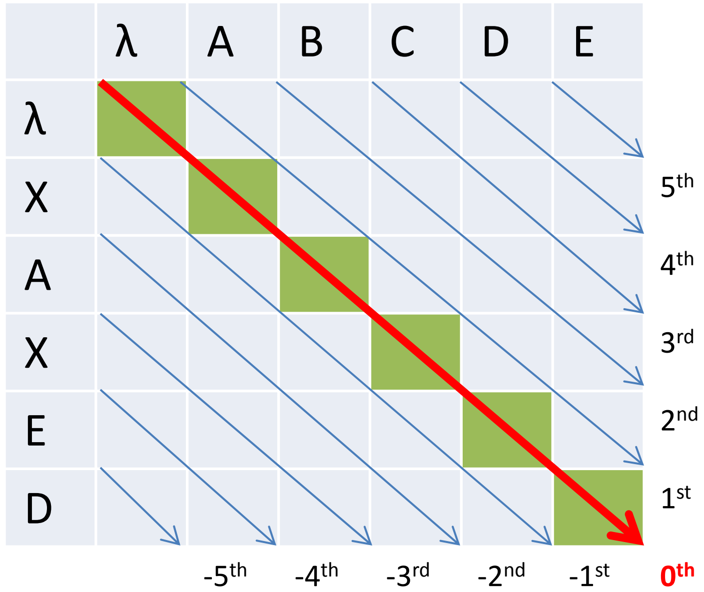
Observations

- Diagonal
  - k-th diagonal k = j-i



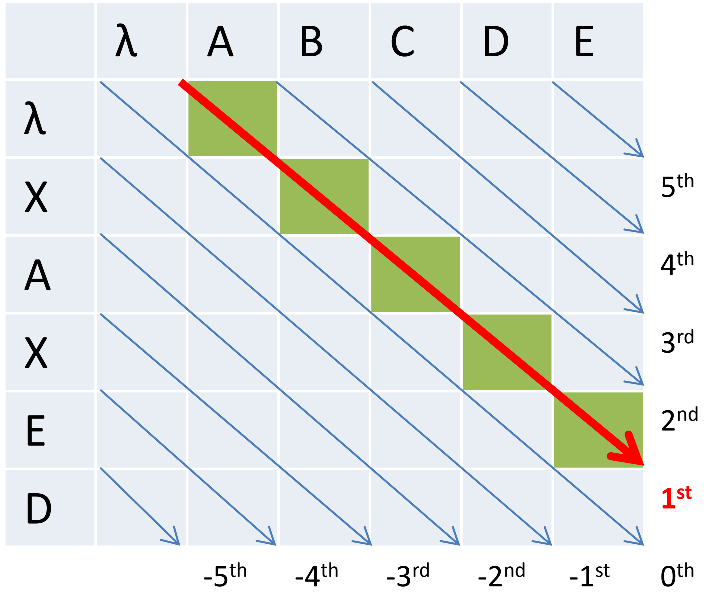
Observations

- Diagonal
  - k-th diagonal k = j-i



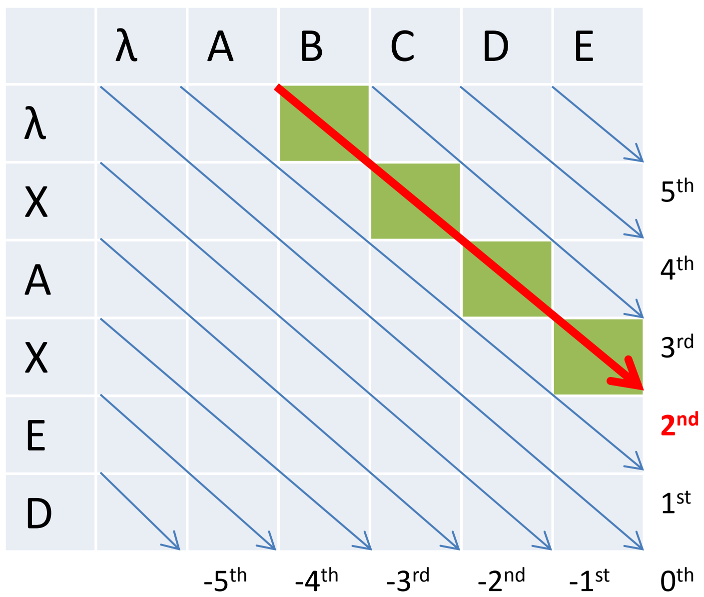
Observations

- Diagonal
  - k-th diagonal k = j-i



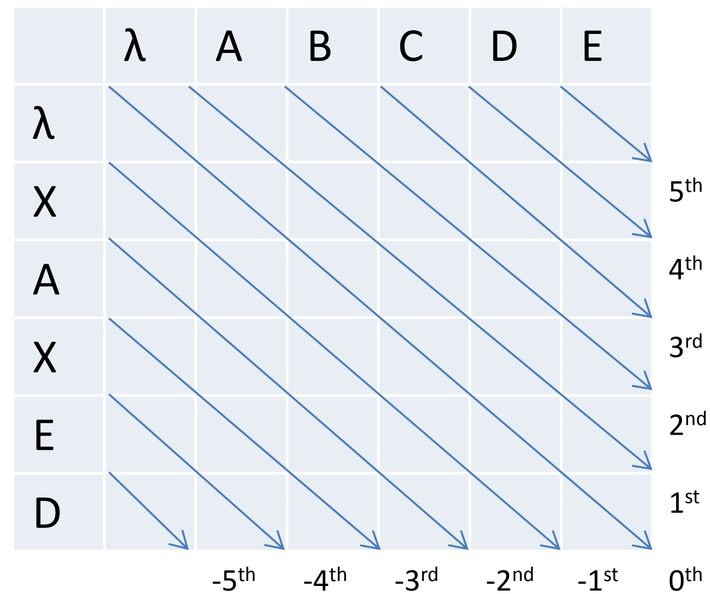
Observations

- Diagonal
  - k-th diagonal k = j-i



Observations

- Diagonal
  - k-th diagonal k = j-i
- Observations



Observations

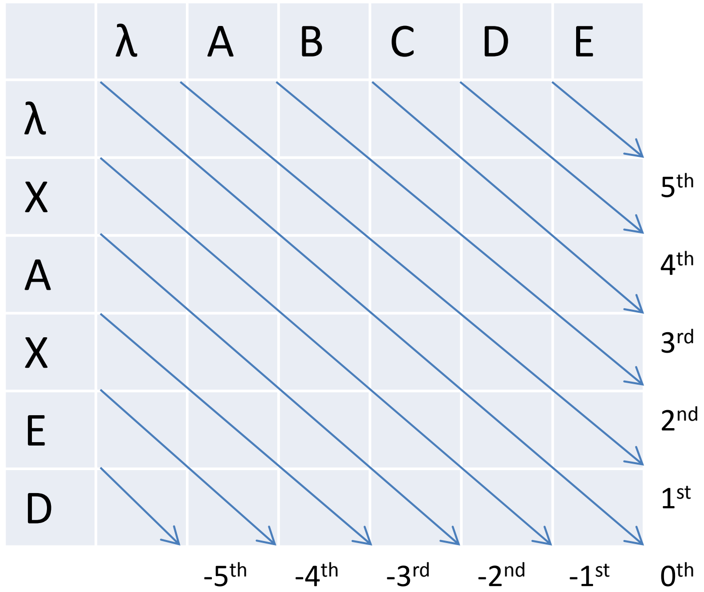
- Diagonal
  - k-th diagonal k = j-i
- Observations
  - Nondecreasing

	λ	A	B	C	D	E	
λ	0	1	2	3	4	5	
X	1	1	2	3	4	5	5^{th}
A	2	1	2	3	4	5	4^{th}
X	3	2	2	3	4	5	3^{rd}
E	4	3	3	3	4	4	2^{nd}
D	5	4	4	4	3	4	1^{st}
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}

Observations

- Diagonal
  - k-th diagonal k = j-i
- Observations
  - Nondecreasing

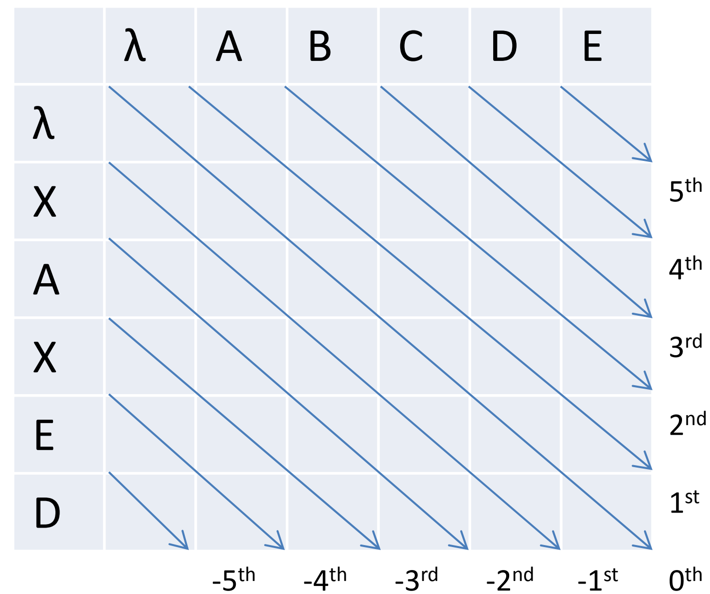
    $d(i,j) - 1 \leq d(i-1,j-1) \leq d(i,j)$



Observations

- Diagonal
  - k-th diagonal k = j-i
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost



Cost increases by 1 when edit path goes cross from a diagonal to an adjacent diagonal

Observations

- Diagonal
  - k-th diagonal k = j-i
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost

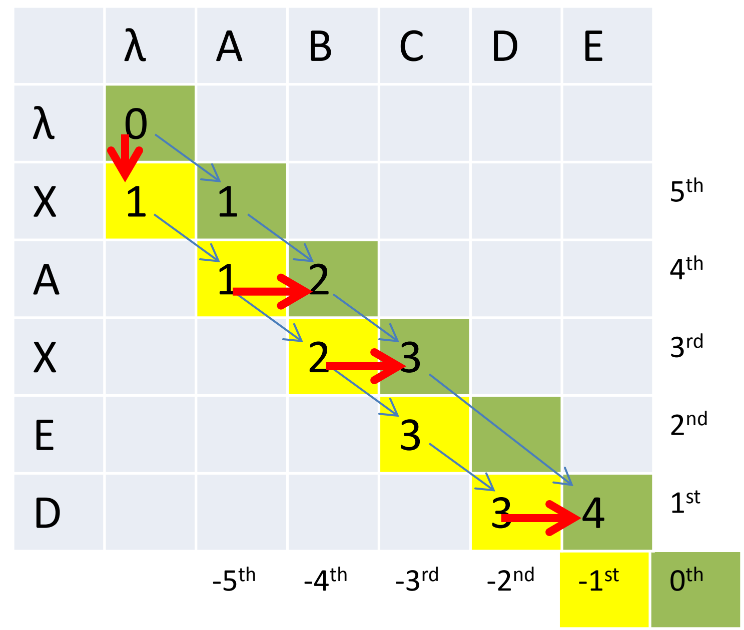
	λ	A	B	C	D	E	
λ	0						
X	1	1					5^{th}
A		1	2				4^{th}
X			2	3			3^{rd}
E				3			2^{nd}
D					3	4	1^{st}
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}

Cost increases by 1 when edit path goes cross from a diagonal to an adjacent diagonal

Observations

- Diagonal
  - k-th diagonal $k = j-i$
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost



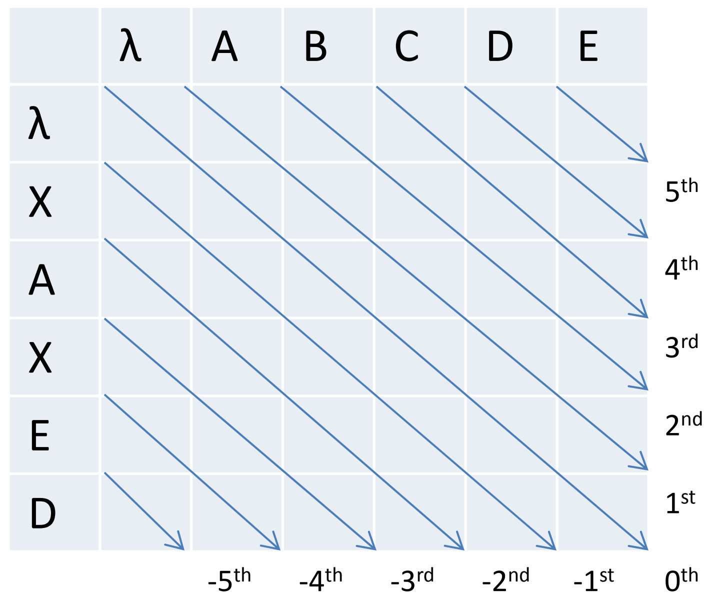
Cost increases by 1 when edit path goes cross from a diagonal to an adjacent diagonal

Observations

- Diagonal
  - k-th diagonal $k = j-i$
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost

    $d' \ge d + |k' - k|$

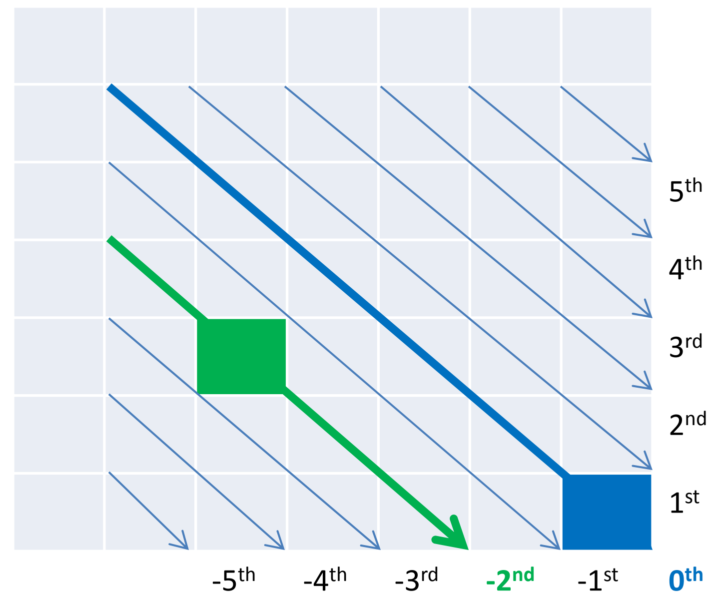


Observations

- Diagonal
  - k-th diagonal $k = j-i$
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost

    $d' \ge d + |k' - k|$



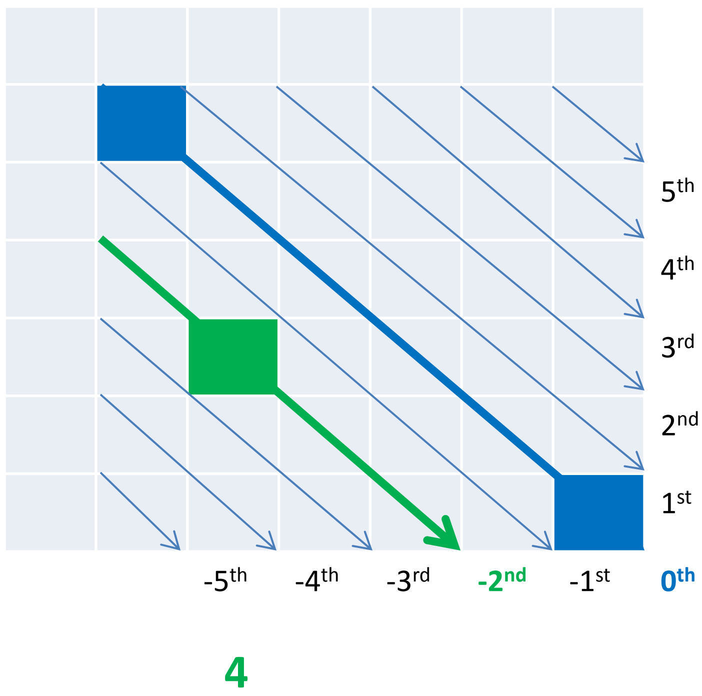
$$d \ge d + |0 - -2|$$
$$\ge d + 2$$

Observations

- Diagonal
  - k-th diagonal k = j-i
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost

    $d' \ge d + |k' - k|$

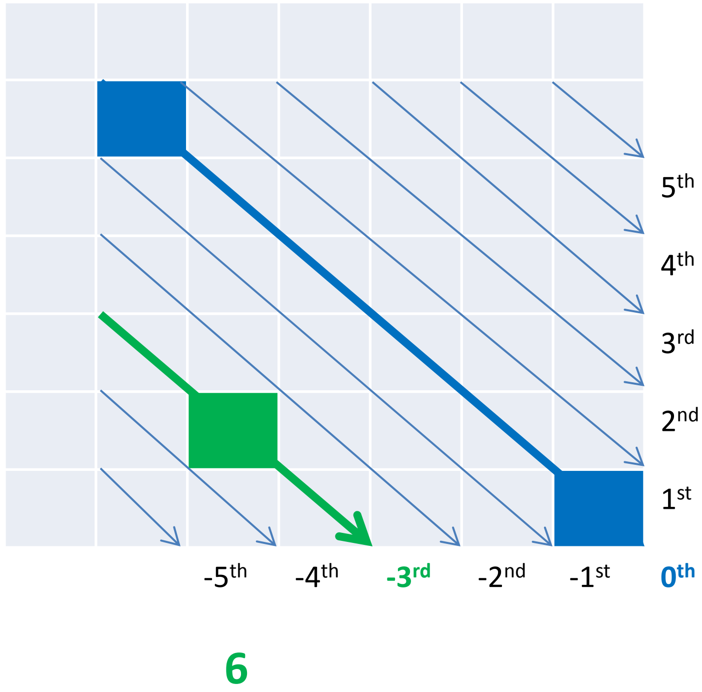


Observations

- Diagonal
  - k-th diagonal k = j-i
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost

    $d' \ge d + |k' - k|$

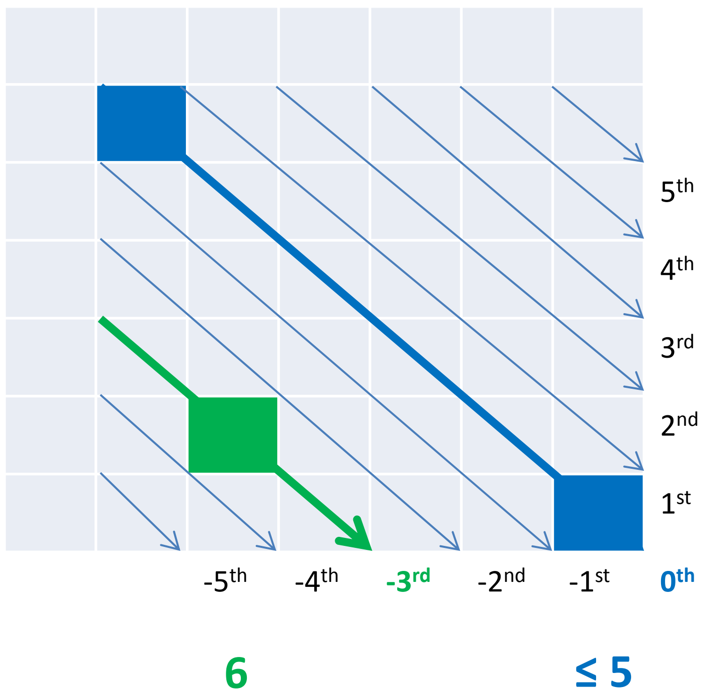


Observations

- Diagonal
  - k-th diagonal $k = j-i$
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost

    $d' \ge d + |k' - k|$

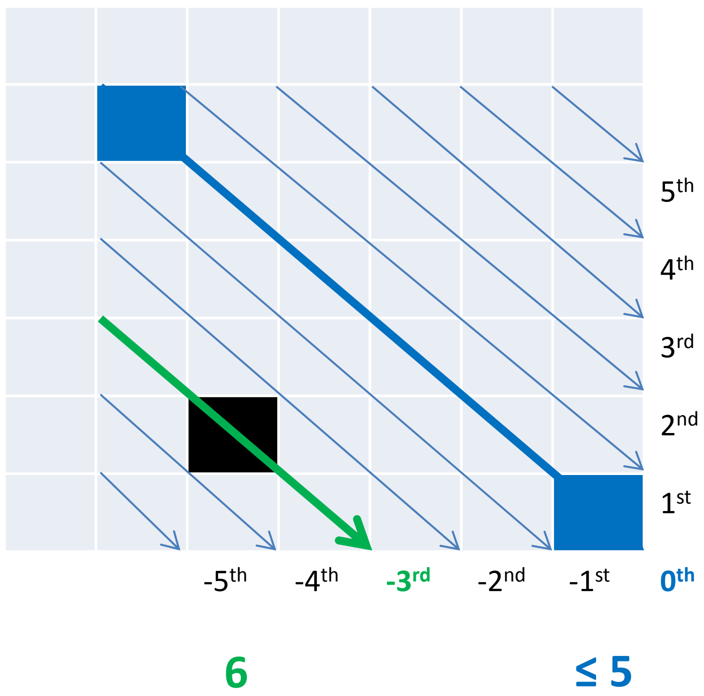


Observations

- Diagonal
  - k-th diagonal $k = j-i$
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost

    $d' \ge d + |k' - k|$

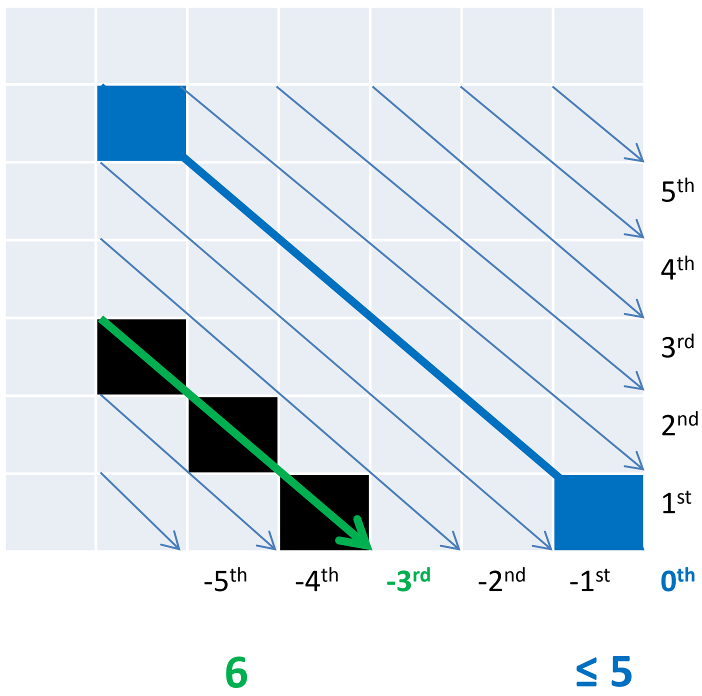


Observations

- Diagonal
  - k-th diagonal $k = j-i$
- Observations
  - Nondecreasing

    $d(i,j) - 1 \leq d(i-1,j-1) \leq d(i,j)$
  - Cross diagonal cost

    $d' \geq d + |k' - k|$

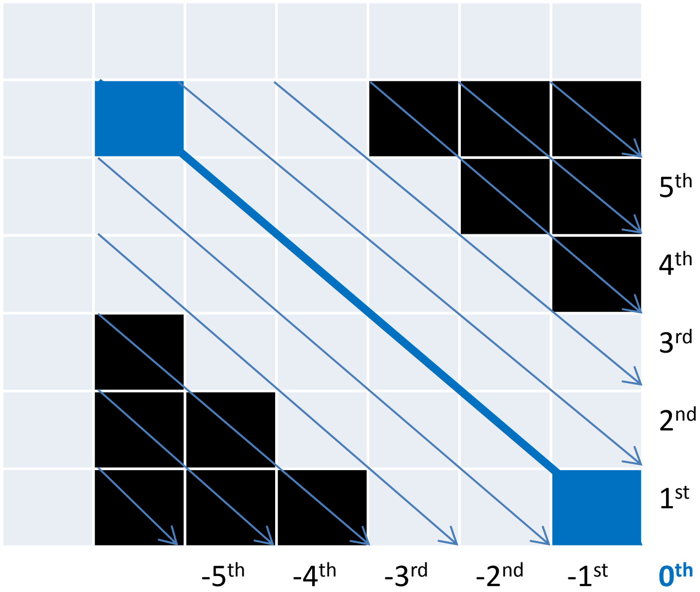


Observations

- Diagonal
  - k-th diagonal k = j-i
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost

    $d' \ge d + |k' - k|$

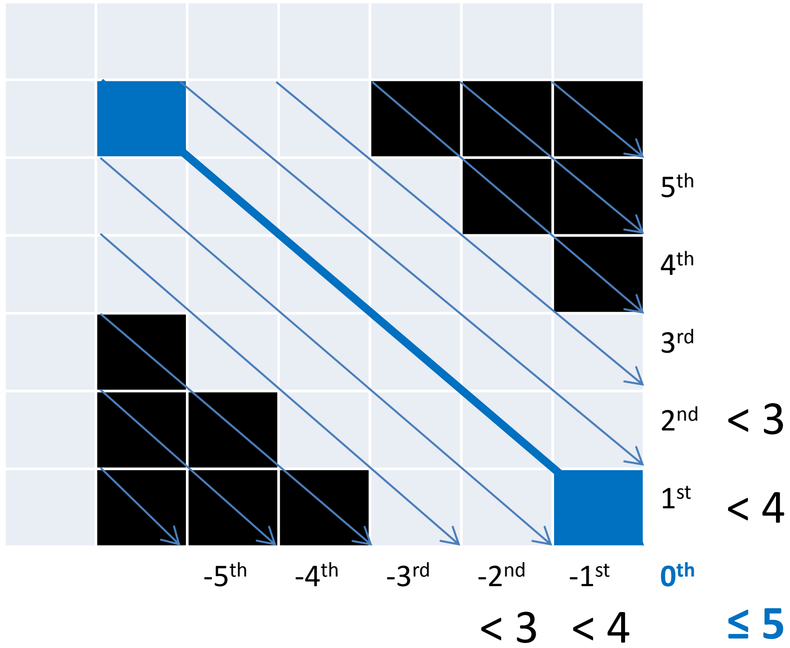


Observations

- Diagonal
  - k-th diagonal $k = j-i$
- Observations
  - Nondecreasing

    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost

    $d' \ge d + |k' - k|$

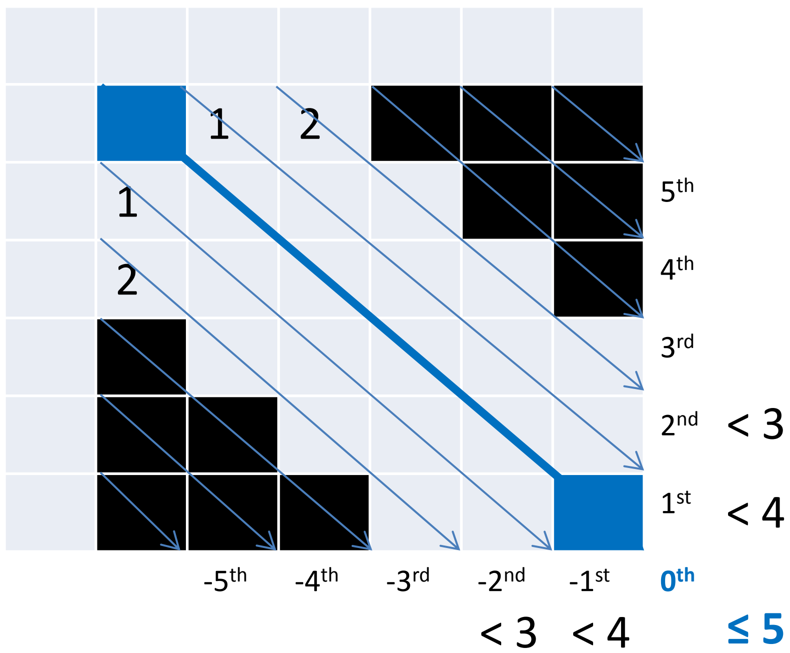


Observations

- Diagonal
  - k-th diagonal k = j-i
- Observations
  - Nondecreasing

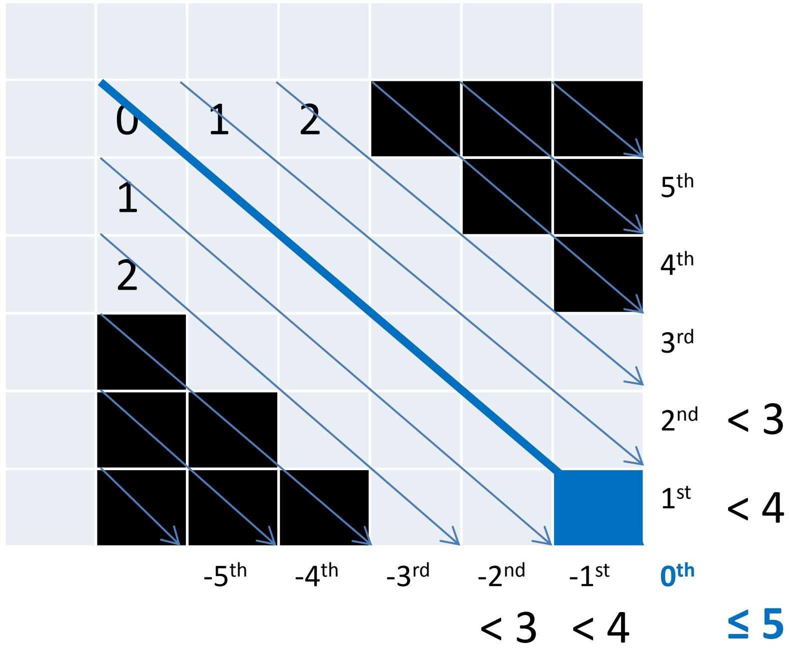
    $d(i,j) - 1 \leq d(i-1,j-1) \leq d(i,j)$
  - Cross diagonal cost

    $d' \geq d + |k' - k|$



Observations

- Diagonal
  - k-th diagonal $k = j-i$
- Observations
  - Nondecreasing
    $d(i,j) - 1 \le d(i-1,j-1) \le d(i,j)$
  - Cross diagonal cost
    $d' \ge d + |k' - k|$



Algorithms

Algorithms

- Basic Idea
  - Way of storing

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2	3			5^{th}	
A	2	1	2	3	4		4^{th}	
X		2	2	3	4	5	3^{rd}	
E			3	3	4	4	2^{nd}	< 3
D				4	3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2	3			5th	
A	2	1	2	3	4		4th	
X		2	2	3	4	5	3rd	
E			3	3	4	4	2nd	< 3
D				4	3	4	1st	< 4
		-5th	-4th	-3rd	-2nd	-1st	0th	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2	3			5^{th}	
A	2	1	2	3	4		4^{th}	
X		2	2	3	4	5	3^{rd}	
E			3	3	4	4	2^{nd}	< 3
D				4	3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2				5^{th}	
A	2	1	2	3			4^{th}	
X		2	2	3	4		3^{rd}	
E			3	3	4	4	2^{nd}	< 3
D				4	3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2				5th	
A	2	1	2	3			4th	
X		2	2	3	4		3rd	
E			3	3	4	4	2nd	< 3
D				4	3	4	1st	< 4
		-5th	-4th	-3rd	-2nd	-1st	0th	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2				5^{th}	
A	2	1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E			3	3	4		2^{nd}	< 3
D				4	3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2				5^{th}	
A	2	1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E			3	3	4		2^{nd}	< 3
D				4	3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2				5th	
A	2	1	2	3			4th	
X		2	2	3			3rd	
E			3	3			2nd	< 3
D				4	3	4	1st	< 4
		-5th	-4th	-3rd	-2nd	-1st	0th	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2				5^{th}	
A	2	1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E			3	3			2^{nd}	< 3
D				4	3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A	2	1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E			3				2^{nd}	< 3
D				4	3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

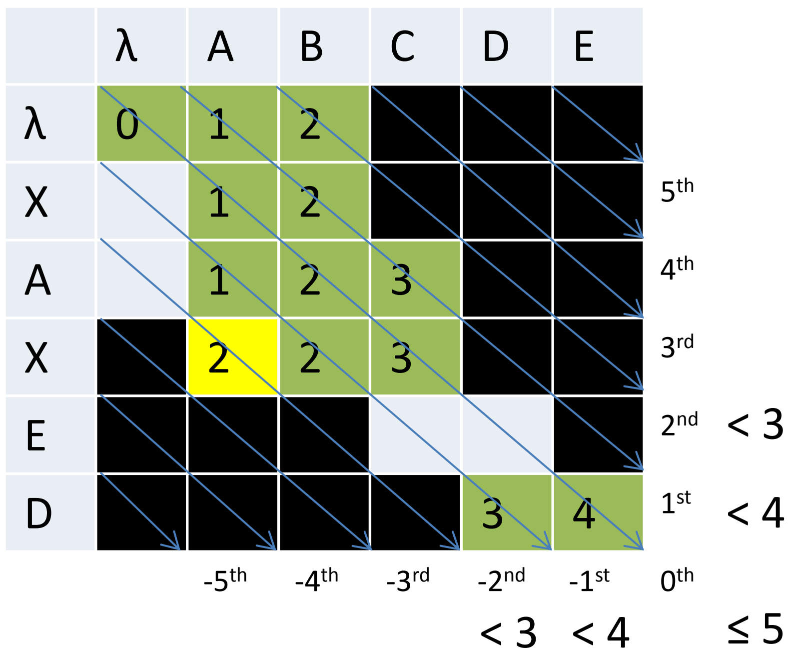
    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A	2	1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E			3				2^{nd}	< 3
D				4	3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.



Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5th	
A		1	2	3			4th	
X		2	2	3			3rd	
E							2nd	< 3
D					3	4	1st	< 4
		-5th	-4th	-3rd	-2nd	-1st	0th	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.
  - Calculate f(k,d) in dp style

    f(n-m,d) = m

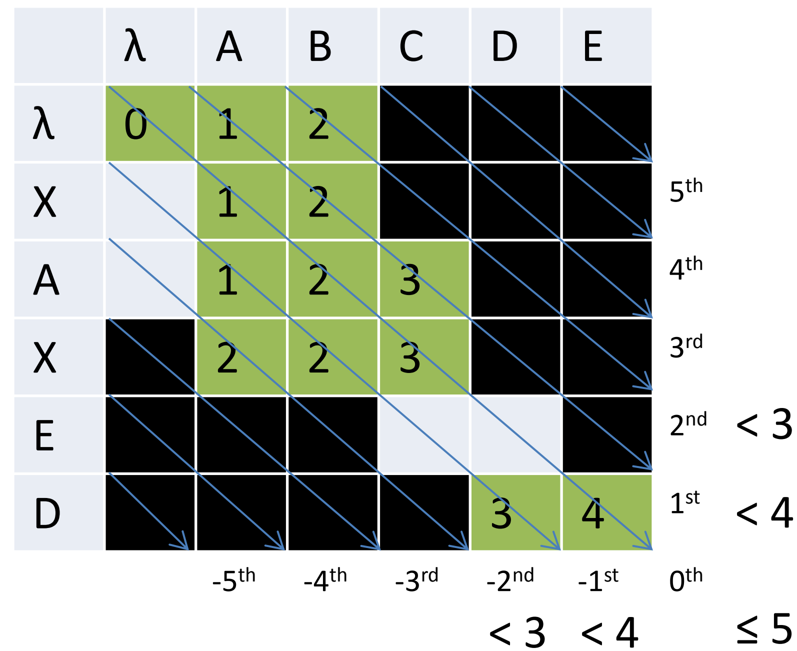
	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A		1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

Algorithms

- Basic Idea
  - Way of storing

    f(k,d) the row of the last appearance of a value d on the k-th diagonal, which tells where the value increases.
  - Calculate f(k,d) in dp style

    f(n-m,d) = m



f(0,4) = 5

Algorithms

- f(k,d)

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A		1	**2**	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

f(0,2) = {?

Algorithms

- f(k,d)

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A		1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

$f(0,2) = \{f(0,1) + 1$

Algorithms

- f(k,d)

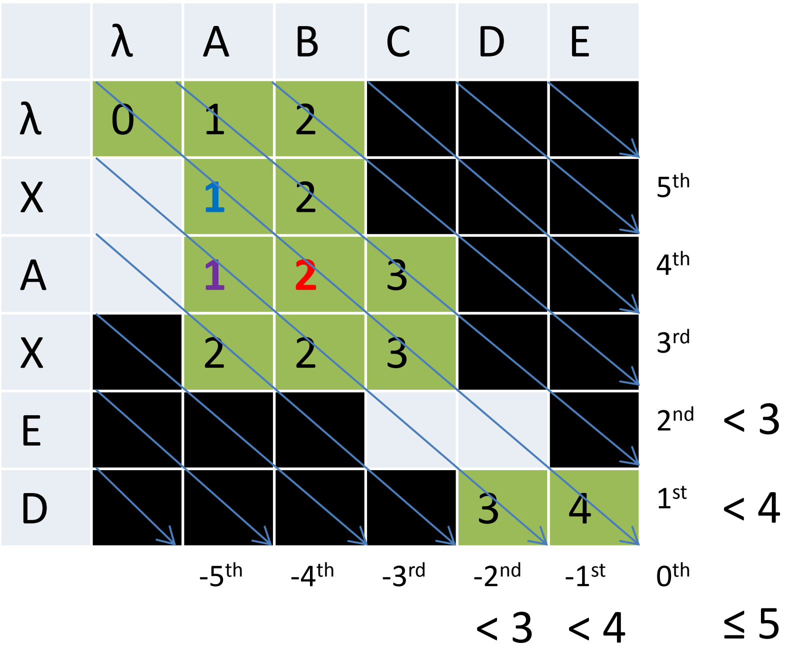
	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A		1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

$$f(0,2) = \begin{cases} f(0,1) + 1 \\ f(-1,1) \end{cases}$$

Algorithms

- f(k,d)

$$i=\max\begin{cases} f(k,d-1)+1 \\ f(k-1,d-1) \end{cases}$$



$$f(0,2) = \begin{cases} f(0,1) + 1 \\ f(-1,1) \end{cases}$$

Algorithms

- f(k,d)

$$i=\max\begin{cases} f(k,d-1)+1 \\ f(k-1,d-1) \\ \end{cases}$$

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A		1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

f(0,4) = {?

Algorithms

- f(k,d)

$$i=\max\begin{cases} f(k, d-1) + 1 \\ f(k-1, d-1) \end{cases}$$

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5th	
A		1	2	3			4th	
X		2	2	3			3rd	
E							2nd	< 3
D					3	4	1st	< 4
		-5th	-4th	-3rd	-2nd	-1st	0th	
					< 3	< 4		≤ 5

f(0,4) = {f(−1,3)

Algorithms

- f(k,d)

$$i = \max\begin{cases} f(k, d-1) + 1/2 \\ f(k-1, d-1) \end{cases}$$

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5th	
A		1	2	3			4th	
X		2	2	**3**			3rd	
E							2nd	< 3
D					**3**	**4**	1st	< 4
		-5th	-4th	-3rd	-2nd	-1st	0th	
					< 3	< 4		≤ 5

$$f(0,4) = \begin{cases} f(-1,3) \\ f(0,3) + 2 \end{cases}$$

Algorithms

- f(k,d)

$$\text{i=max}\begin{cases} f(k, d-1) + 1/2 \\ f(k-1, d-1) \\ \end{cases}$$

	λ	A	B	C	D	E		
λ	0	1	2					
X	**1**	1	2				5th	
A		1	2	3			4th	
X		2	2	3			3rd	
E							2nd	< 3
D					3	4	1st	< 4
		-5th	-4th	-3rd	-2nd	-1st	0th	
					< 3	< 4		≤ 5

f(-1,1) = {?

Algorithms

- f(k,d)

$$
i=\max\begin{cases} f(k,d-1)+1/2 \\ f(k-1,d-1) \\ (k+1,d-1)+1 \end{cases}
$$

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2				5^{th}	
A		1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

f(-1,1) = {f(0,0) + 1

Algorithms

- f(k,d)

$$
i=\max\begin{cases} f(k, d-1) + 1/2 \\ f(k-1, d-1) \\ f(k+1, d-1) + 1 \end{cases}
$$

	λ	A	B	C	D	E		
λ	0	1	2					
X	**1**	1	2				5^{th}	
A		1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

f(-1,1) = 1 ?

Algorithms

- f(k,d)

$$i=\max\begin{cases} f(k,d-1)+1/2 \\ f(k-1,d-1) \\ f(k+1,d-1)+1 \end{cases}$$

	λ	A	B	C	D	E		
λ	0	1	2					
X	1	1	2				5^{th}	
A		1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

f(-1,1) = 1 ?

Algorithms

- f(k,d)

$$i=\max\begin{cases} f(k,d-1)+1/2 \\ f(k-1,d-1) \\ f(k+1,d-1)+1 \end{cases}$$

while($A_{i+1}=B_{i+1+k}$)

    i++;

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A		1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

f(-1,1) = 2 ?

Algorithms

- f(k,d)

$$i=\max\begin{cases} f(k,d-1)+1/2 \\ f(k-1,d-1) \\ f(k+1,d-1)+1 \end{cases}$$

while($A_{i+1}=B_{i+1+k}$)

i++;

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A		1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3	4	1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

f(-1,1) = 2 ?

Algorithms

- f(k,d)

$$i=\max\begin{cases} f(k,d-1)+1/2 \\ f(k-1,d-1) \\ f(k+1,d-1)+1 \end{cases}$$

while($A_{i+1}=B_{i+1+k}$)

    i++;

f(k,d) = i

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5th	
A		1	2	3			4th	
X		2	2	3			3rd	
E							2nd	< 3
D					3	4	1st	< 4
		-5th	-4th	-3rd	-2nd	-1st	0th	
					< 3	< 4		≤ 5

f(-1,1) = 2

Algorithms

- f(k,d)

$$i=\max\begin{cases} f(k,d-1)+1/2 \\ f(k-1,d-1) \\ f(k+1,d-1)+1 \end{cases}$$

while($A_{i+1}=B_{i+1+k}$)

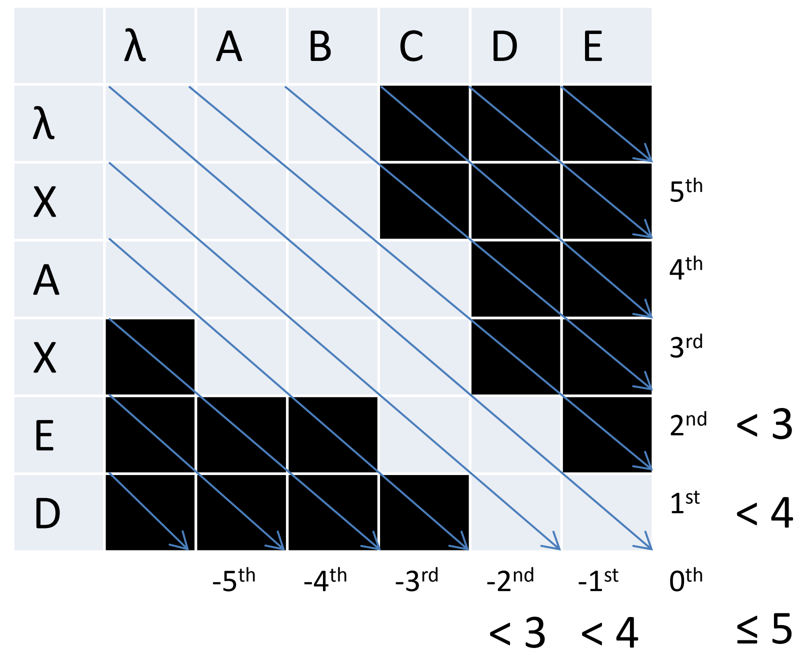
i++;

f(k,d) = i

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5th	
A		1	2	3			4th	
X		2	2	3			3rd	
E							2nd	< 3
D					3	4	1st	< 4
		-5th	-4th	-3rd	-2nd	-1st	0th	
					< 3	< 4		≤ 5

Algorithms

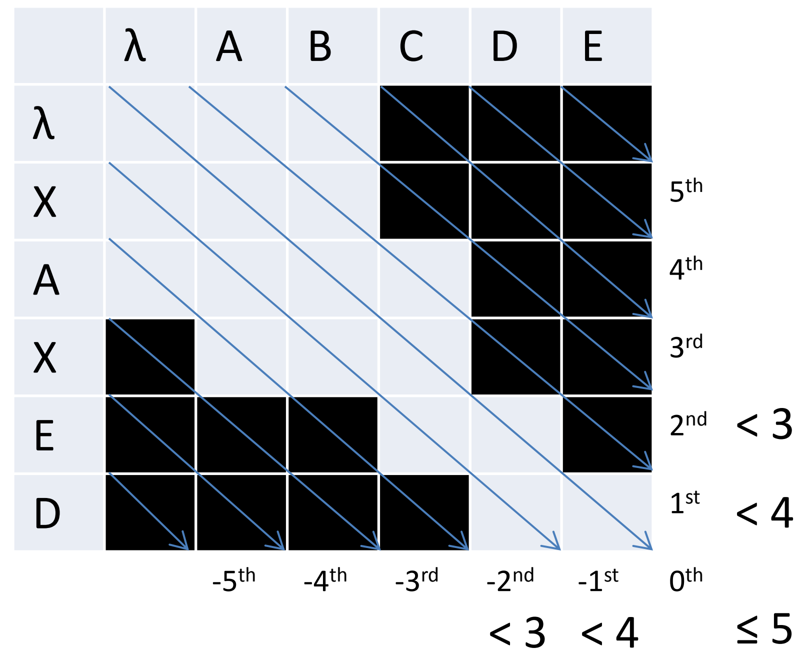
- dp



Algorithms

- dp

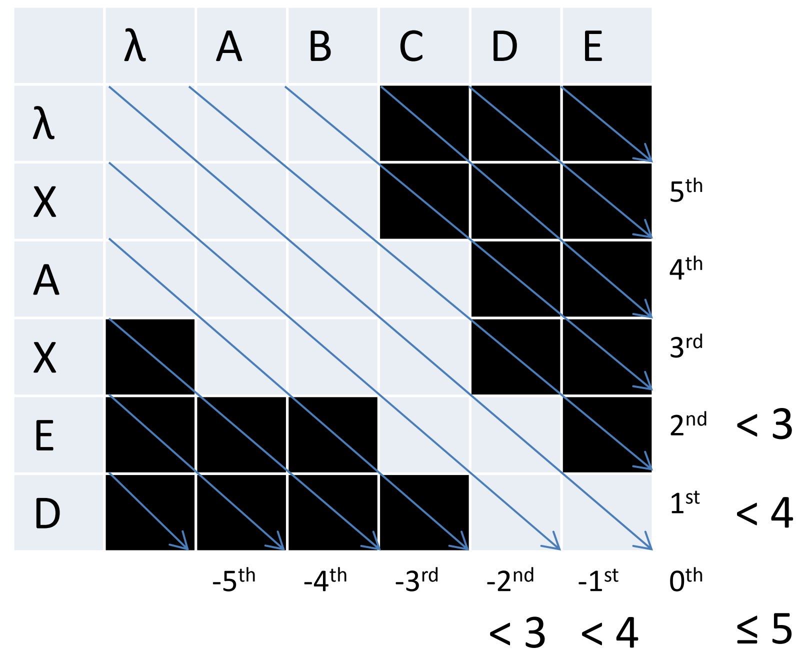
$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$



Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

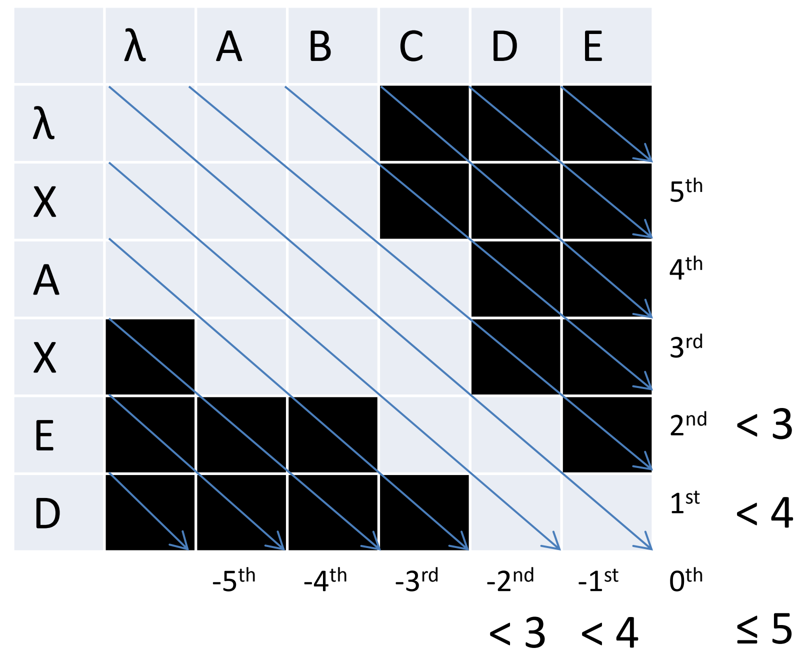


+2
+1
0
-1
-2

Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$



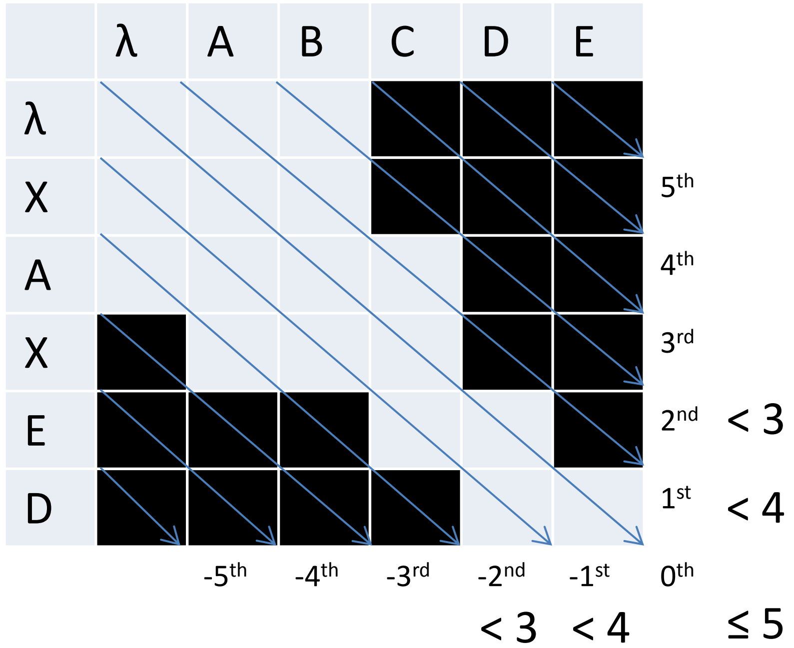
+2					
+1					
0					
-1					
-2					
	d=0	d=1	d=2	d=3	d=4

Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

+2	f(+2,0)	f(+2,1)	f(+2,2)	f(+2,3)	f(+2,4)
+1	f(+1,0)	f(+1,1)	f(+1,2)	f(+1,3)	f(+1,4)
0	f(0,0)	f(0,1)	f(0,2)	f(0,3)	f(0,4)
-1	f(-1,0)	f(-1,1)	f(-1,2)	f(-1,3)	f(-1,4)
-2	f(-2,0)	f(-2,1)	f(-2,2)	f(-2,3)	f(-2,4)
	d=0	d=1	d=2	d=3	d=4

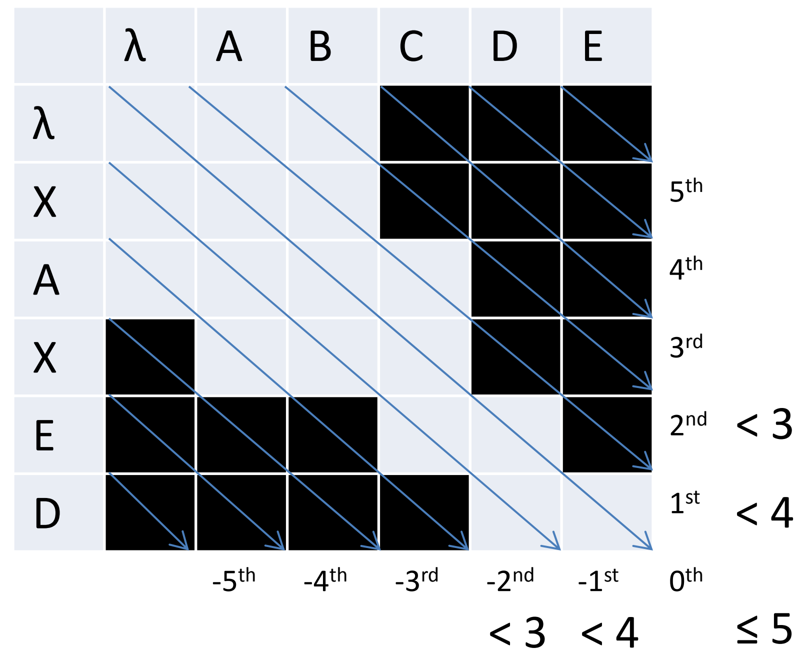


Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

+2	f(+2,0)	f(+2,1)			
+1	f(+1,0)				
0					
-1	f(-1,0)				
-2	f(-2,0)	f(-2,1)			
	d=0	d=1	d=2	d=3	d=4

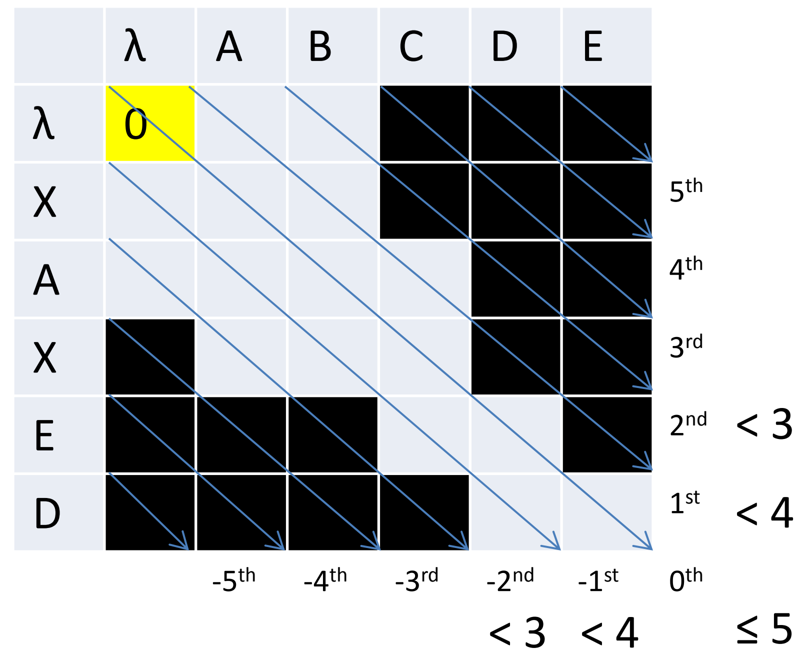


Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

+2	f(+2,0)	f(+2,1)			
+1	f(+1,0)				
0	f(0,0)				
-1	f(-1,0)				
-2	f(-2,0)	f(-2,1)			
	d=0	d=1	d=2	d=3	d=4

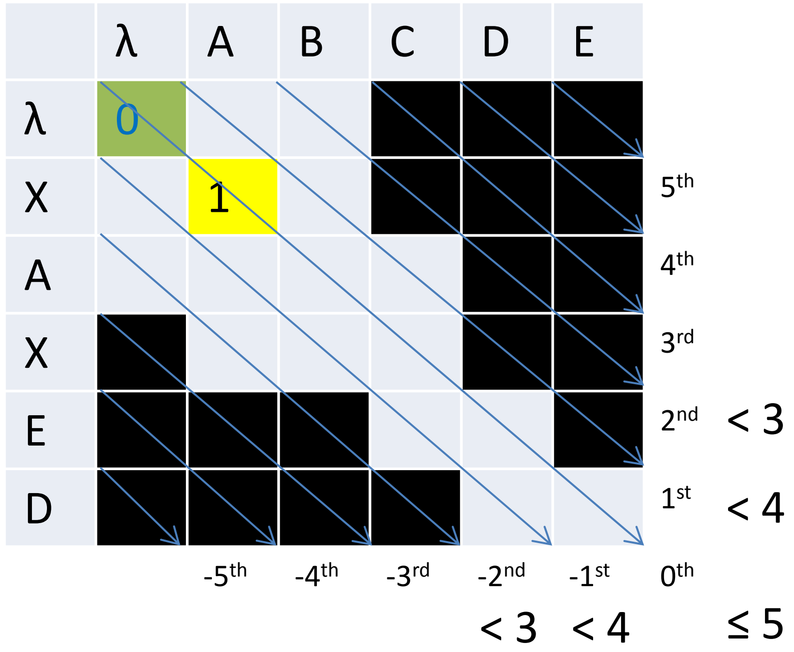


Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

	d=0	d=1	d=2	d=3	d=4
+2	f(+2,0)	f(+2,1)			
+1	f(+1,0)				
0	f(0,0)	f(0,1)			
-1	f(-1,0)				
-2	f(-2,0)	f(-2,1)			

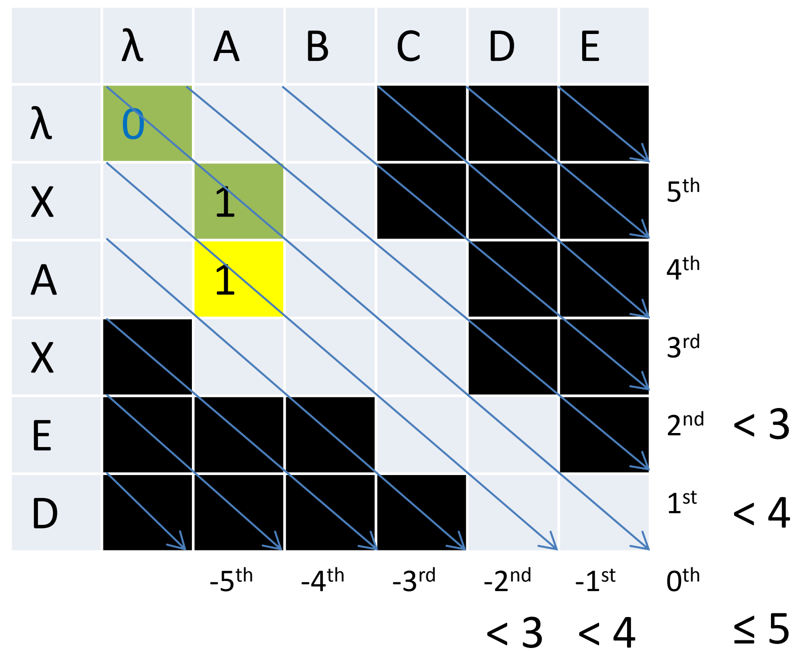


Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

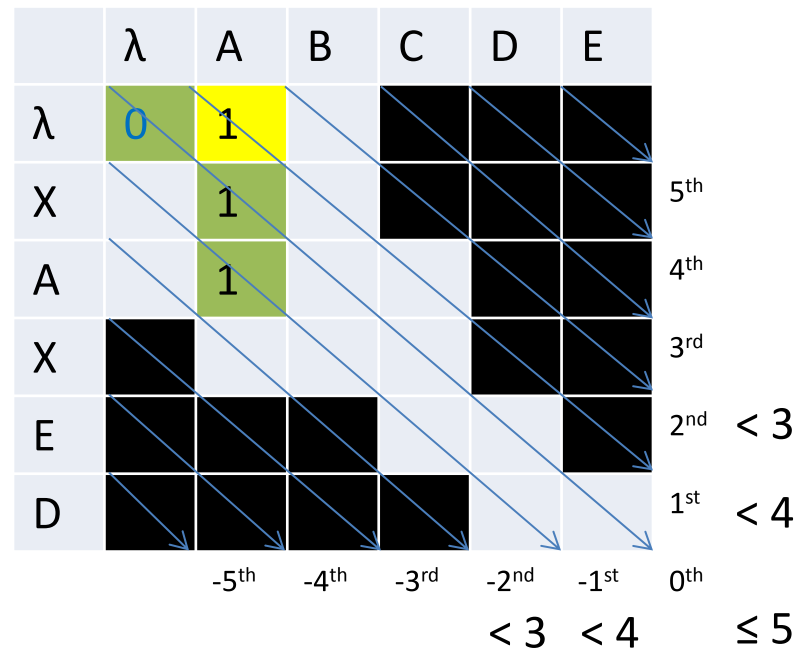
+2	f(+2,0)	f(+2,1)			
+1	f(+1,0)				
0	f(0,0)	f(0,1)			
-1	f(-1,0)	f(-1,1)			
-2	f(-2,0)	f(-2,1)			
	d=0	d=1	d=2	d=3	d=4



Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$



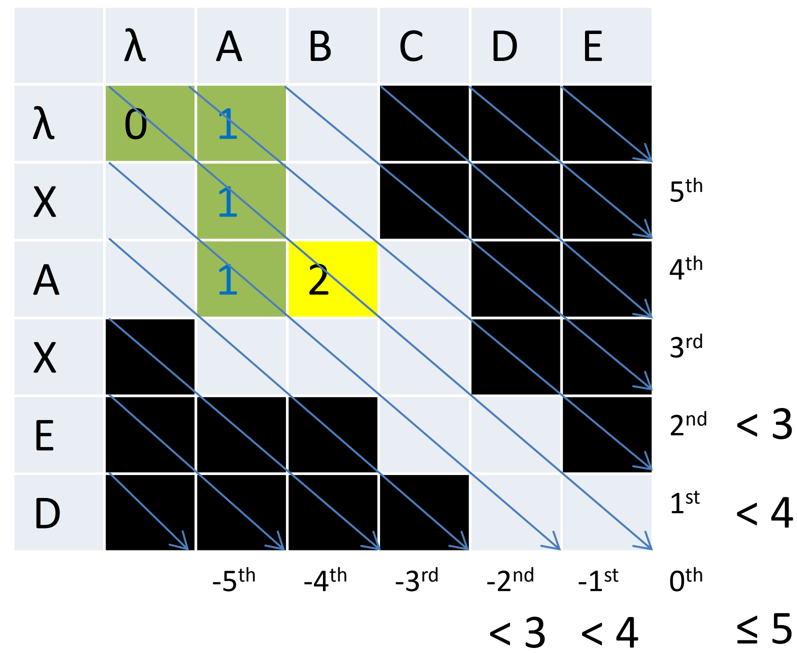
+2	f(+2,0)	f(+2,1)			
+1	f(+1,0)	f(+1,1)			
0	f(0,0)	f(0,1)			
-1	f(-1,0)	f(-1,1)			
-2	f(-2,0)	f(-2,1)			
	d=0	d=1	d=2	d=3	d=4

Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

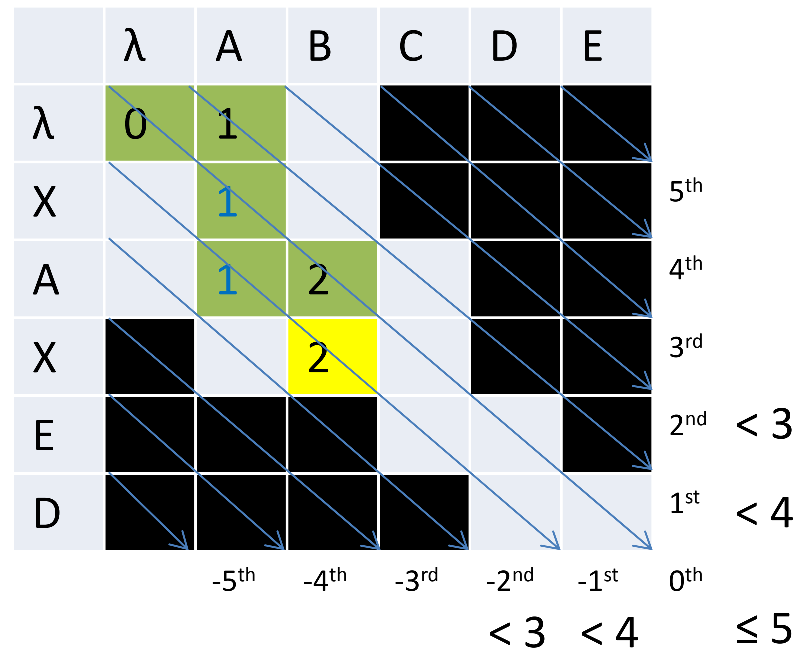
+2	f(+2,0)	f(+2,1)			
+1	f(+1,0)	f(+1,1)			
0	f(0,0)	f(0,1)	f(0,2)		
-1	f(-1,0)	f(-1,1)			
-2	f(-2,0)	f(-2,1)			
	d=0	d=1	d=2	d=3	d=4



Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$



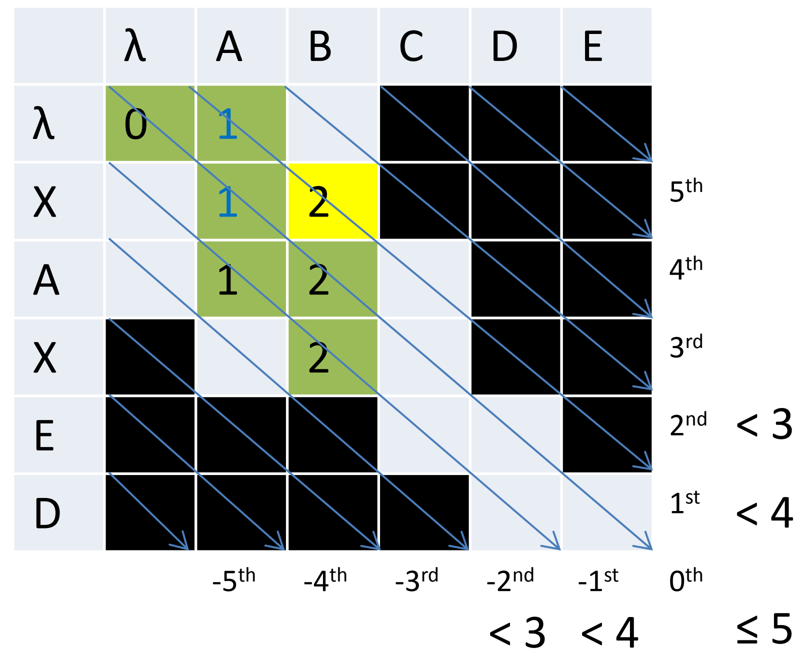
+2	f(+2,0)	f(+2,1)			
+1	f(+1,0)	f(+1,1)			
0	f(0,0)	f(0,1)	f(0,2)		
-1	f(-1,0)	f(-1,1)	f(-1,2)		
-2	f(-2,0)	f(-2,1)			
	d=0	d=1	d=2	d=3	d=4

Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

+2	f(+2,0)	f(+2,1)			
+1	f(+1,0)	f(+1,1)	f(+1,2)		
0	f(0,0)	f(0,1)	f(0,2)		
-1	f(-1,0)	f(-1,1)	f(-1,2)		
-2	f(-2,0)	f(-2,1)			
	d=0	d=1	d=2	d=3	d=4

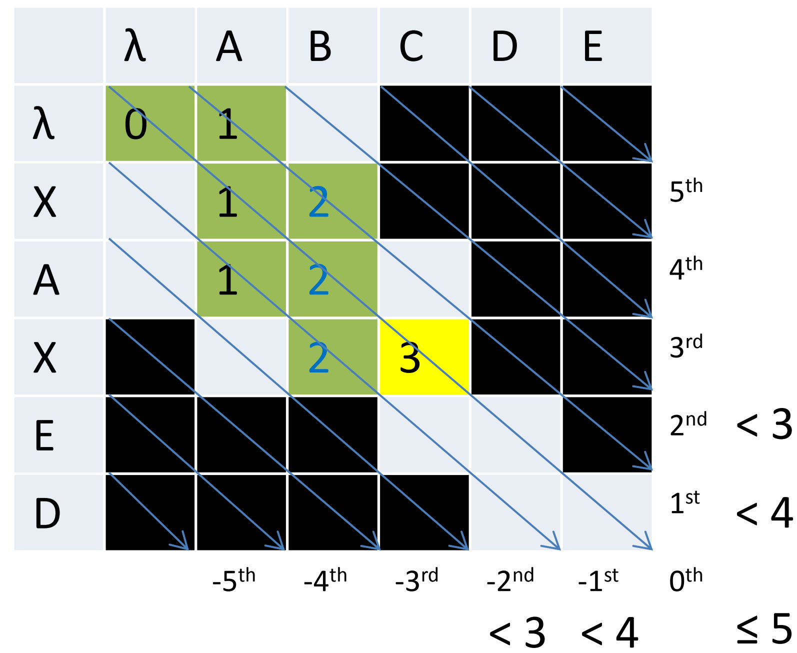


Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

+2	f(+2,0)	f(+2,1)			
+1	f(+1,0)	f(+1,1)	f(+1,2)		
0	f(0,0)	f(0,1)	f(0,2)	f(0,3)	
-1	f(-1,0)	f(-1,1)	f(-1,2)		
-2	f(-2,0)	f(-2,1)			
	d=0	d=1	d=2	d=3	d=4



Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

	λ	A	B	C	D	E		
λ	0	1						
X		1	2				5^{th}	
A		1	2				4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D							1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

+2	f(+2,0)	f(+2,1)			
+1	f(+1,0)	f(+1,1)	f(+1,2)		
0	f(0,0)	f(0,1)	f(0,2)	f(0,3)	
-1	f(-1,0)	f(-1,1)	f(-1,2)		
-2	f(-2,0)	f(-2,1)	f(-2,2)		
	d=0	d=1	d=2	d=3	d=4

Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

	λ	A	B	C	D	E	
λ	0	1	2				
X		1	2				5^{th}
A		1	2				4^{th}
X		2	2	3			3^{rd}
E							2^{nd} < 3
D							1^{st} < 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}
					< 3	< 4	≤ 5

+2	f(+2,0)	f(+2,1)	f(+2,2)		
+1	f(+1,0)	f(+1,1)	f(+1,2)		
0	f(0,0)	f(0,1)	f(0,2)	f(0,3)	
-1	f(-1,0)	f(-1,1)	f(-1,2)		
-2	f(-2,0)	f(-2,1)	f(-2,2)		
	d=0	d=1	d=2	d=3	d=4

Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

+2	f(+2,0)	f(+2,1)	f(+2,2)			
+1	f(+1,0)	f(+1,1)	f(+1,2)			
0	f(0,0)	f(0,1)	f(0,2)	f(0,3)		
-1	f(-1,0)	f(-1,1)	f(-1,2)	f(-1,3)		
-2	f(-2,0)	f(-2,1)	f(-2,2)			
	d=0	d=1	d=2	d=3	d=4	

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A		1	2				4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3		1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5^{th}	
A		1	2	3			4^{th}	
X		2	2	3			3^{rd}	
E							2^{nd}	< 3
D					3		1^{st}	< 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd}	-1^{st}	0^{th}	
					< 3	< 4		≤ 5

+2	f(+2,0)	f(+2,1)	f(+2,2)		
+1	f(+1,0)	f(+1,1)	f(+1,2)	f(+1,3)	
0	f(0,0)	f(0,1)	f(0,2)	f(0,3)	
-1	f(-1,0)	f(-1,1)	f(-1,2)	f(-1,3)	
-2	f(-2,0)	f(-2,1)	f(-2,2)		
	d=0	d=1	d=2	d=3	d=4

Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

	λ	A	B	C	D	E	
λ	0	1	2				
X		1	2				5^{th}
A		1	2	3			4^{th}
X		2	2	3			3^{rd}
E							2^{nd} < 3
D					3	4	1^{st} < 4
		-5^{th}	-4^{th}	-3^{rd}	-2^{nd} < 3	-1^{st} < 4	0^{th} ≤ 5

+2	f(+2,0)	f(+2,1)	f(+2,2)		
+1	f(+1,0)	f(+1,1)	f(+1,2)	f(+1,3)	
0	f(0,0)	f(0,1)	f(0,2)	f(0,3)	f(0,4)
-1	f(-1,0)	f(-1,1)	f(-1,2)	f(-1,3)	
-2	f(-2,0)	f(-2,1)	f(-2,2)		
	d=0	d=1	d=2	d=3	d=4

Algorithms

- dp

$$f(k,d):\begin{cases} f(k,d-1) \\ f(k-1,d-1) \\ f(k+1,d-1) \end{cases}$$

+2	f(+2,0)	f(+2,1)	f(+2,2)		
+1	f(+1,0)	f(+1,1)	f(+1,2)	f(+1,3)	
0	f(0,0)	f(0,1)	f(0,2)	f(0,3)	**f(0,4)**
-1	f(-1,0)	f(-1,1)	f(-1,2)	f(-1,3)	
-2	f(-2,0)	f(-2,1)	f(-2,2)		
	d=0	d=1	d=2	d=3	d=4

	λ	A	B	C	D	E		
λ	0	1	2					
X		1	2				5th	
A		1	2	3			4th	
X		2	2	3			3rd	
E							2nd	< 3
D					3	**4**	1st	< 4
		-5th	-4th	-3rd	-2nd	-1st	0th	
					< 3	< 4		≤ 5

Analysis

- $f(k,d)$ entries $\Rightarrow$ space and time complexity

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5
3	-∞	-∞	-∞	-1			
2	-∞	-∞	-1				
1	-∞	-1					
0	-1						
-1	-∞	0					
-2	-∞	-∞	1				
-3	-∞	-∞	-∞	2			

	λ	F	G	H	I	J
λ						
A						
B						
C						
D						
E						

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5
3	-∞	-∞	-∞	-1			
2	-∞	-∞	-1	0	1		
1	-∞	-1	0	1	2	3	
0	-1	0	1	2	3	4	5
-1	-∞	0	1	2	3	4	
-2	-∞	-∞	1	2	3		
-3	-∞	-∞	-∞	2			

	λ	F	G	H	I	J
λ	0	1	2			
A	1	1	2	3		
B	2	2	2	3		
C		3	3	3	4	
D				4	4	
E						5

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5
3	-∞	-∞	-∞	-1			
2	-∞	-∞	-1	0	1		
1	-∞	-1	0	1	2	3	
0	-1	0	1	2	3	4	5
-1	-∞	0	1	2	3	4	
-2	-∞	-∞	1	2	3		
-3	-∞	-∞	-∞	2			

	λ	F	G	H	I	J
λ	0	1	2			
A	1	1	2	3		
B	2	2	2	3		
C		3	3	3	4	
D				4	4	
E						5

D = 5

K = 0

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5
3	-∞	-∞	-∞	-1			
2	-∞	-∞	-1	0	1		
1	-∞	-1	0	1	2	3	
0	-1	0	1	2	3	4	5
-1	-∞	0	1	2	3	4	
-2	-∞	-∞	1	2	3		
-3	-∞	-∞	-∞	2			

M

f(k,**0**) upto f(k,**D**)

	λ	F	G	H	I	J
λ	0	1	2			
A	1	1	2	3		
B	2	2	2	3		
C		3	3	3	4	
D				4	4	
E						5

M = D + 1 **D** = 5

K = 0

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5	
3	-∞	-∞	-∞	-1				
2	-∞	-∞	-1	0	1			
1	-∞	-1	0	1	2	3		M-2
0	-1	0	1	2	3	4	5	M
-1	-∞	0	1	2	3	4		M-2
-2	-∞	-∞	1	2	3			
-3	-∞	-∞	-∞	2				

f(k,**1**) upto f(k,**D-1**)

	λ	F	G	H	I	J
λ	0	1	2			
A	1	1	2	3		
B	2	2	2	3		
C		3	3	3	4	
D				4	4	
E						5

M = D + 1

D = 5

K = 0

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5	
3	-∞	-∞	-∞	-1				
2	-∞	-∞	-1	0	1			M-2-2
1	-∞	-1	0	1	2	3		M-2
0	-1	0	1	2	3	4	5	M
-1	-∞	0	1	2	3	4		M-2
-2	-∞	-∞	1	2	3			M-2-2
-3	-∞	-∞	-∞	2				

f(k,**2**) upto f(k,**D-2**)

	λ	F	G	H	I	J
λ	0	1	2			
A	1	1	2	3		
B	2	2	2	3		
C		3	3	3	4	
D				4	4	
E						5

M = D + 1

D = 5

K = 0

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5	
3	-∞	-∞	-∞	-1				
2	-∞	-∞	-1	0	1			M-2-2
1	-∞	-1	0	1	2	3		M-2
0	-1	0	1	2	3	4	5	M
-1	-∞	0	1	2	3	4		M-2
-2	-∞	-∞	1	2	3			M-2-2
-3	-∞	-∞	-∞	2				

2,4,⋯,M-2,M,M-2,⋯4,2

	λ	F	G	H	I	J
λ	0	1	2			
A	1	1	2	3		
B	2	2	2	3		
C		3	3	3	4	
D				4	4	
E						5

M = D + 1 **D** = 5

K = 0

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5	
3	-∞	-∞	-∞	-1				
2	-∞	-∞	-1	0	1			M-2-2
1	-∞	-1	0	1	2	3		M-2
0	-1	0	1	2	3	4	5	M
-1	-∞	0	1	2	3	4		M-2
-2	-∞	-∞	1	2	3			M-2-2
-3	-∞	-∞	-∞	2				

1,3,⋯,M-2,M,M-2,⋯3,1

	λ	F	G	H	I	J
λ	0	1	2			
A	1	1	2	3		
B	2	2	2	3		
C		3	3	3	4	
D				4	4	
E						5

M = D + 1 **D** = 4

K = 0

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5	
3	-∞	-∞	-∞	-1				
2	-∞	-∞	-1	0	1			M-2-2
1	-∞	-1	0	1	2	3		M-2
0	-1	0	1	2	3	4	5	M
-1	-∞	0	1	2	3	4		M-2
-2	-∞	-∞	1	2	3			M-2-2
-3	-∞	-∞	-∞	2				

	λ	F	G	H	I	J
λ	0	1	2			
A	1	1	2	3		
B	2	2	2	3		
C		3	3	3	4	
D				4	4	
E						5

What if k ≠ 0 ?

M = D + 1

D = 4

K = 0

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5
3	-∞	-∞	-∞	-1			
2	-∞	-∞	-1				
1	-∞	-1					
0	-1						
-1	-∞	0					
-2	-∞	-∞	1				
-3	-∞	-∞	-∞	2			

What if k ≠ 0 ?

	λ	G	A	R	V	E	Y
λ							
A							
V							
E							
R							
Y							

M =

D =

K = 1

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5
3	-∞	-∞	-∞	-1			
2	-∞	-∞	-1	3			
1	-∞	-1	1	2	5		
0	-1	0	1	2			
-1	-∞	0	1				
-2	-∞	-∞	1				
-3	-∞	-∞	-∞	2			

What if k ≠ 0 ?

	λ	G	A	R	V	E	Y
λ	0						
A	1	1	1				
V			2	2			
E						2	
R							
Y							3

M =

D = 3

K = 1

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5
3	-∞	-∞	-∞	-1			
2	-∞	-∞	-1	3			
1	-∞	-1	1	2	5		
0	-1	0	1	2			
-1	-∞	0	1				
-2	-∞	-∞	1				
-3	-∞	-∞	-∞	2			

M

f(k,**K**) upto f(k,**D**)

	λ	G	A	R	V	E	Y
λ	0						
A	1	1	1				
V			2	2			
E						2	
R							
Y							3

M = D – K + 1 **D** = 3

= 3 **K** = 1

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5
3	-∞	-∞	-∞	-1			
2	-∞	-∞	-1	3			
1	-∞	-1	1	2	5		
0	-1	0	1	2			
-1	-∞	0	1				
-2	-∞	-∞	1				
-3	-∞	-∞	-∞	2			

M

M

	λ	G	A	R	V	E	Y
λ	0						
A	1	1	1				
V			2	2			
E						2	
R							
Y							3

f(k,**k**) upto f(k,**D-|K-k|**)

+ |K-k| **-** |K-k|

M = D – K + 1 = 3

D = 3

K = 1

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5
3	-∞	-∞	-∞	-1			
2	-∞	-∞	-1	3			
1	-∞	-1	1	2	5		
0	-1	0	1	2			
-1	-∞	0	1				
-2	-∞	-∞	1				
-3	-∞	-∞	-∞	2			

M

M

$0^{th} - K^{th}$: $(K + 1)\,M$

	λ	G	A	R	V	E	Y
λ	0						
A	1	1	1				
V			2	2			
E						2	
R							
Y							3

M = D – K + 1 **D** = 3

= 3 **K** = 1

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5	
3	-∞	-∞	-∞	-1				
2	-∞	-∞	-1	3				M-2
1	-∞	-1	1	2	5			M
0	-1	0	1	2				M
-1	-∞	0	1					M-2
-2	-∞	-∞	1					
-3	-∞	-∞	-∞	2				

f(k,**k**) upto f(k,**D-|K-k|**)

	λ	G	A	R	V	E	Y
λ	0						
A	1	1	1				
V			2	2			
E						2	
R							
Y							3

M = D – K + 1 **D** = 3

= 3 **K** = 1

Analysis

- f(k,d) entries ⇒ space and time complexity

	-1	0	1	2	3	4	5	
3	-∞	-∞	-∞	-1				
2	-∞	-∞	-1	3				M-2
1	-∞	-1	1	2	5			M
0	-1	0	1	2				M
-1	-∞	0	1					M-2
-2	-∞	-∞	1					
-3	-∞	-∞	-∞	2				

1,3,⋯,M-2,(K+1)M,M-2,⋯3,1

	λ	G	A	R	V	E	Y
λ	0						
A	1	1	1				
V			2	2			
E						2	
R							
Y							3

M = D – K + 1 = 3

D = 3

K = 1

Analysis

- f(k,d) entries ⇒ space and time complexity
  - $\cdots$,M-2,(K+1)M,M-2,$\cdots$
  - M = D – K + 1

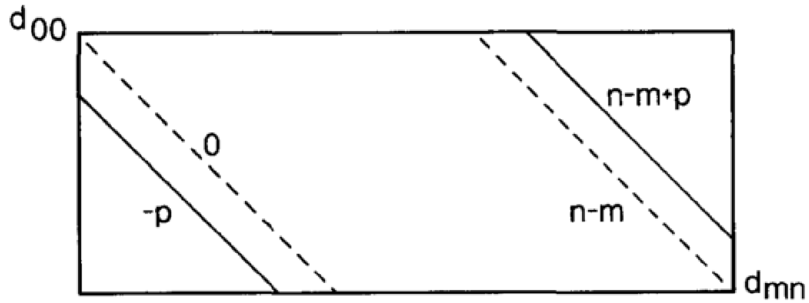
Conclusion

Conclusion

- Compare with Ukkonen's algorithm
  - f(k,d) entry
    - ≤ Ukkonen

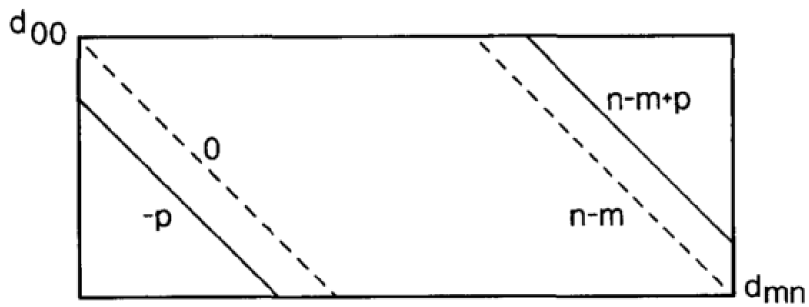
Conclusion

- Compare with Ukkonen's algorithm
  - f(k,d) entry
    - ≤ Ukkonen



Conclusion

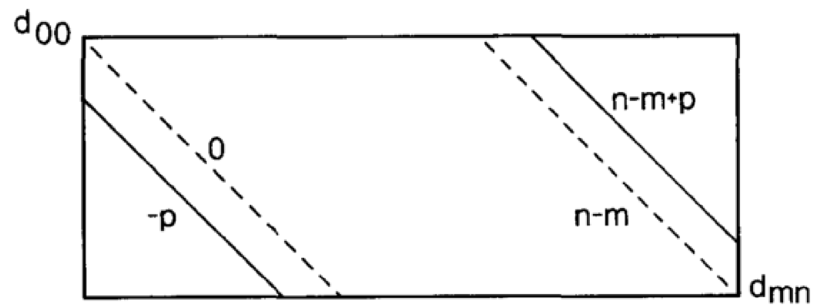
- Compare with Ukkonen's algorithm
  - f(k,d) entry
    - ≤ Ukkonen



		F	G	H	I	J
	0	1	2			
A	1	1	2	3		-
B	2	2	2	3		-
C		3	3	3	4	-
D				4	4	-
E						5

Conclusion

- Compare with Ukkonen's algorithm
  - f(k,d) entry
    - ≤ Ukkonen



		F	G	H	I	J
	0	1	2			
A	1	1	2	3		-
B	2	2	2	3		-
C		3	3	3	4	-
D				4	4	-
E						5

		G	A	R	V	E	Y
	0	-	-	-	-	-	-
A	1	1	1	-	-	-	-
V			2	2	-	-	-
E			-		-	2	-
R	-	-	-		-	-	-
Y	-	-	-	-	-	-	3

Conclusion

- Compare with Ukkonen's algorithm
  - f(k,d) entry
    - ≤ Ukkonen
  - Time efficiency
    - ≤ Ukkonen $O(t \cdot \min(m,n))$

Conclusion

- Compare with Ukkonen's algorithm
  - f(k,d) entry
    - ≤ Ukkonen
  - Time efficiency
    - ≤ Ukkonen $O(t \cdot \min(m,n))$
  - Space efficiency
    - (distance only) ≥ Ukkonen $O(\min(t,m,n))$
    - (path recoverable) ≤ Ukkonen $O(\min(t,m,n))$

- Thanks for listening
- Thanks for your patience

- Any question ?