THE LEAST WEIGHT SUBSEQUENCE PROBLEM

Shir Yerushalmi & Amir Harel

What is LWS?

- We define an instance of the **L**east **W**eight **S**ubsequence problem on [a, b] as follows:

  - We are given:
    - a real-valued weight function $W(i, j)$, defined for all $a \le i<j \le b$.

  - We want to find:
    - $i_0, i_1, \ldots, i_t \in \mathbb{N}$ s.t
    - $a= i_0< i_1<\ldots< i_t =b$
    - $\Sigma_{1\le s \le t} W(i_{s-1}, i_s)=W(i_0, i_1)+W(i_1,i_2)+ \ldots + W(i_{t-1}, i_t)$ Will be minimal.

Traditional Algorithm

Optimum Paragraph Formation

Optimum Paragraph Formation

Take a given text and **dynamically** cut the lines to a given length, without cutting words.

The Walt Disney Company has gotten the green light from Chinese authorities to build one of its theme parks in Shanghai.

"China is one of the most dynamic, exciting and important countries in the world, and this approval marks a very significant milestone for The Walt Disney Company in mainland China," said Robert A. Iger, Disney president and CEO, on Tuesday.

Optimum Paragraph Formation

SWITCH TO: | CNN U.S. | CNN ARABIC

Sign up | Log in

SEARCH

POWERED BY Google

Home | Video | World | U.S. | Africa | Asia | Europe | Latin America | Middle East | Business | World Sport | Entertainment | Tech | Travel | iReport

[-] Feedback

Disney theme park set for Shanghai

Mixx | Facebook | Twitter | Share | Email | Save | Print

November 5, 2009 -- Updated 0151 GMT (0951 HKT)

A young Chinese girl displays Disney products in Shanghai's town of Chuansha on March 7, 2008.

(CNN) -- The Walt Disney Company has gotten the green light from Chinese authorities to build one of its theme parks in Shanghai.

"China is one of the most dynamic, exciting and important countries in the world, and this approval marks a very significant milestone for The Walt Disney Company in mainland China," said Robert A. Iger, Disney president and CEO, on Tuesday.

Goal:

To form the text into a paragraph, minimizing the total penalty.

Optimum Paragraph Formation

Penalty:

There is an optimum length for a line: **lineopt.**

The penalty for a line
being too short or too long =
(length of current line – lineopt)^2

*Exception:
last line cannot be penalized
for being too short.

Optimum Paragraph Formation

Instead of this scroll:

The Walt Disney Company has gotten the green light from Chinese authorities to build one of its theme parks in Shanghai.

"China is one of the most dynamic, exciting and important countries in the world, and this approval marks a very significant milestone for The Walt Disney Company in mainland China," said Robert A. Iger, Disney president and CEO, on Tuesday.

Lets use a more simple example:

a the i and his her arm but

Optimum Paragraph Formation

We want to form a paragraph using this scroll:

a the i and his her arm but

- Minimize total penalty
- Let lineopt = 6

Optimum Paragraph Formation:

Solutions

Traditional Algorithm

a	the	i	and	his	her	arm	but

0 1 2 3 4 5 6 7 8

lineopt = 6

*remember:
penalty = (current line length – lineopt)^2

f(n) = minimum total penalty at index n, using previous n's in the computation

= f(i) + w(i,m) for all i's smaller than n

Goal: find the minimum f(8)

What is w(i,m)?

0 1 2 3 4 5 6 7 8

a	the	i	and	his	her	arm	but

w(i,m) =

penalty from i to m

(# of characters from i to m – lineopt)^2

We want to compute all the values of w(i,m) and store them into a table

w(i,m) Table

m

i	0	1	2	3	4	5	6	7	8
0	0	25	4	1	4	25	64	121	196
1	–	–	9	4	1	16	49	100	169
2	–	–	–	25	4	1	16	49	100
3	–	–	–	–	9	0	9	36	81
4	–	–	–	–	–	9	0	9	36
5	–	–	–	–	–	–	9	0	9
6	–	–	–	–	–	–	–	9	0
7	–	–	–	–	–	–	–	–	9
8	–	–	–	–	–	–	–	–	–

1 2 3

the	i

ex: W(1,3) = (4 - 6) ^ 2 = 4

0 1 2 3 4 5 6 7 8

a	the	i	and	his	her	arm	but

to find f(8),
we need to first
compute f(i) for all
0<i<8`

F(1) =
min{
0 + w(0,1) = 0 + 25 = 25
}

F(2) =
min{
0 + w(0,2) = 0 + 4 = 4
F(1) + w(1,2) = 25 + 9 = 34
}

F(3) =
min{
0 + w(0,3) = 0 + 1 = 1
F(1) + w(1,3) = 25 + 4 = 29
F(2) + w(2,3) = 4 + 25 = 29
}

w(i,m) Table

m

i	0	1	2	3	4	5	6	7	8
0	0	25	4	1	4	25	64	121	196
1	-	-	9	4	1	16	49	100	169
2	-	-	-	25	4	1	16	49	100
3	-	-	-	-	9	0	9	36	81
4	-	-	-	-	-	9	0	9	36
5	-	-	-	-	-	-	9	0	9
6	-	-	-	-	-	-	-	9	0
7	-	-	-	-	-	-	-	-	9
8	-	-	-	-	-	-	-	-	-

0	1	2	3	4	5	6	7	8
a	the	i	and	his	her	arm	but	

Each time we find the minimum, we store the **index** of the minimum in an array called "best"

ex: best(1) = 0
best(2) = 0
best(3) = 0
best(4) = 0
best(5) = 3

w(i,m) Table

m

i	0	1	2	3	4	5	6	7	8
0	0	25	4	1	4	25	64	121	196
1	-	-	9	4	1	16	49	100	169
2	-	-	-	25	4	1	16	49	100
3	-	-	-	-	9	0	9	36	81
4	-	-	-	-	-	9	0	9	36
5	-	-	-	-	-	-	9	0	9
6	-	-	-	-	-	-	-	9	0
7	-	-	-	-	-	-	-	-	9
8	-	-	-	-	-	-	-	-	-

F(2) =
min{
0 + w(0,2) = 0 + 4 = 4
F(1) + w(1,2) = 25 + 9 = 34
}

F(3) =
min{
0 + w(0,3) = 0 + 1 = 1
F(1) + w(1,3) = 25 + 4 = 29
F(2) + w(2,3) = 4 + 25 = 29
}

F(4) =
min{
0 + w(0,4) = 0 + 4 = 4
F(1) + w(1,4) = 1 + 25 = 26
F(2) + w(2,4) = 4 + 4 = 8
F(3) + w(3,4) = 1 + 9 = 10
}

F(5) =
min{
0 + w(0,5) = 0 + 25 = 25
F(1) + w(1,5) = 25 + 16 = 41
F(2) + w(2,5) = 4 + 1 = 5
F(3) + w(3,5) = 1 + 0 = 1
F(4) + w(4,5) = 4 + 9 = 13
}

0	1	2	3	4	5	6	7	8

a	the	i	and	his	her	arm	but

best(1) = 0
best(2) = 0
best(3) = 0
best(4) = 0
best(5) = 3
best(6) = 4
best(7) = 5
best(8) = 6

F(6) =
min{
0 + w(0,6) = 0 + 64 = 64
F(1) + w(1,6) = 25 + 49 = 74
F(2) + w(2,6) = 4 + 16 = 20
F(3) + w(3,6) = 1 + 9 = 10
F(4) + w(4,6) = 4 + 0 = 4
F(5) + w(5,6) = 1 + 9 = 10
}

F(7) =
min{
0 + w(0,7) = 0 + 121 = 121
F(1) + w(1,7) = 25 + 100 = 125
F(2) + w(2,7) = 4 + 49 = 53
F(3) + w(3,7) = 1 + 36 = 37
F(4) + w(4,7) = 4 + 9 = 13
F(5) + w(5,7) = 1 + 0 = 1
F(6) + w(6,7) = 4 + 9 = 13
}

w(i,m) Table

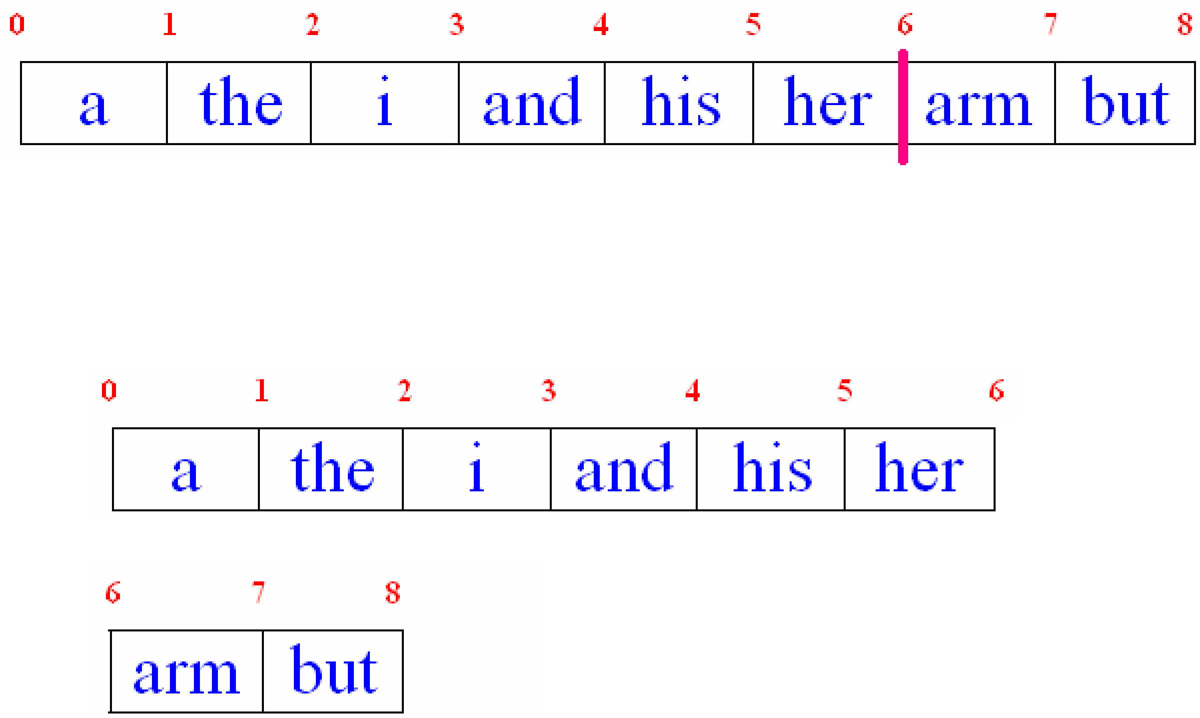
m

i	0	1	2	3	4	5	6	7	8
0	0	25	4	1	4	25	64	121	196
1	–	–	9	4	1	16	49	100	169
2	–	–	–	25	4	1	16	49	100
3	–	–	–	–	9	0	9	36	81
4	–	–	–	–	–	9	0	9	36
5	–	–	–	–	–	–	9	0	9
6	–	–	–	–	–	–	–	9	0
7	–	–	–	–	–	–	–	–	9
8	–	–	–	–	–	–	–	–	–

F(8) =
min{
0 + w(0,8) = 0 + 196 = 196
F(1) + w(1,8) = 25 + 169 = 194
F(2) + w(2,8) = 4 + 100 = 104
F(3) + w(3,8) = 1 + 81 = 82
F(4) + w(4,8) = 4 + 36 = 40
F(5) + w(5,8) = 1 + 9 = 10
F(6) + w(6,8) = 4 + 0 = 4
F(7) + w(7,8) = 1 + 9 = 10
}

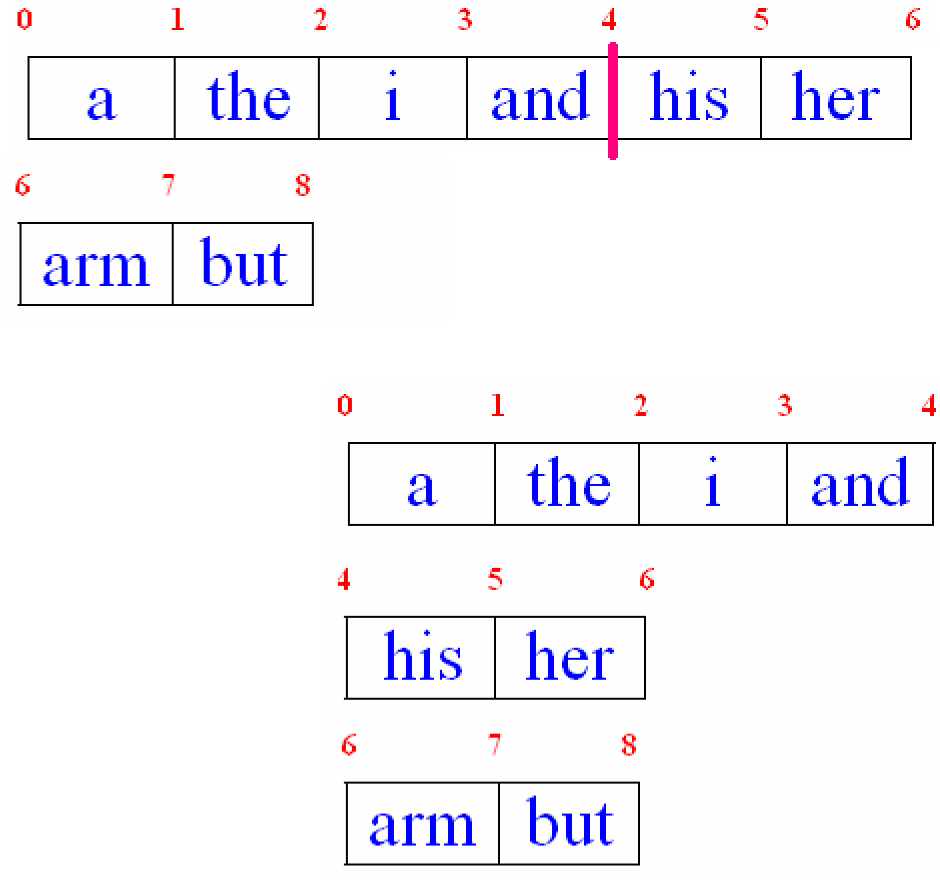
Traditional Algorithm

best(1) = 0
best(2) = 0
best(3) = 0
best(4) = 0
best(5) = 3
best(6) = 4
best(7) = 5
best(8) = 6



Traditional Algorithm

best(1) = 0
best(2) = 0
best(3) = 0
best(4) = 0
best(5) = 3
best(6) = 4
best(7) = 5
best(8) = 6



Traditional Algorithm

best(1) = 0

best(2) = 0

best(3) = 0

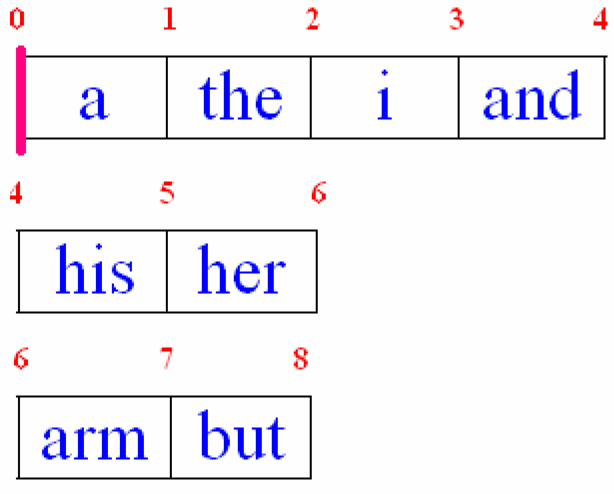
best(4) = 0

best(5) = 3

best(6) = 4

best(7) = 5

best(8) = 6

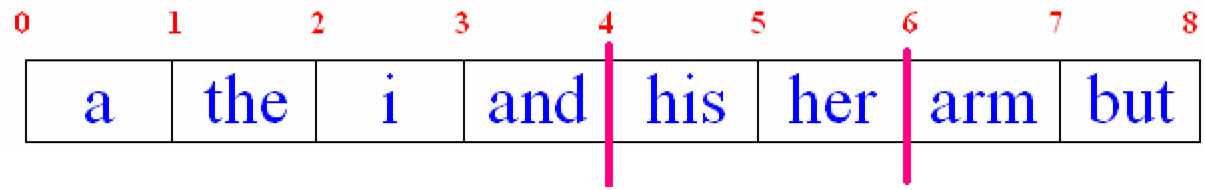


Traditional Algorithm

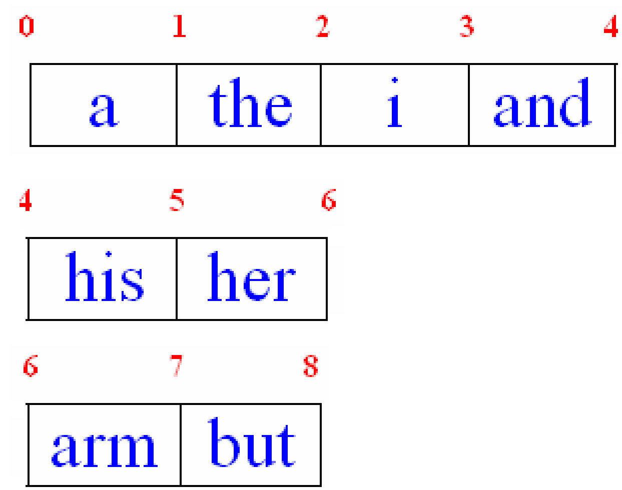
best(4) = 0

best(6) = 4

best(8) = 6



These are the optimal cuts
to minimize penalty
when lineopt = 6.



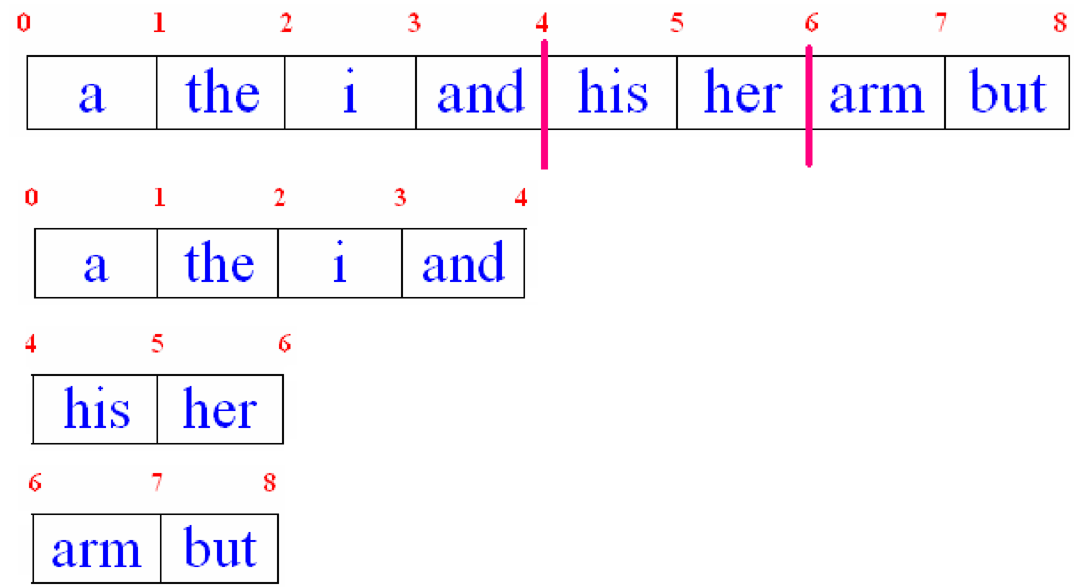
Traditional Algorithm

Optimal cuts
when lineopt = 6:

best(8) = **6**

best(6) = **4**

best(4) = **0**



F(8) =
min{

0	+	w(0,8)	=	0	+	196	=	196
F(1)	+	w(1,8)	=	25	+	169	=	194
F(2)	+	w(2,8)	=	4	+	100	=	104
F(3)	+	w(3,8)	=	1	+	81	=	82
F(4)	+	w(4,8)	=	4	+	36	=	40
F(5)	+	w(5,8)	=	1	+	9	=	10
F(6)	+	w(6,8)	=	4	+	0	=	4
F(7)	+	w(7,8)	=	1	+	9	=	10

}

Penalty = 4

The Traditional Algorithm

- f[0] ← 0
- for m from 1 to n do
- begin

    f [m]← W(0,m)

    best [m]← 0

    For i from 1 to m-1 do

        if (f [i] + W (i, m)) < f [m] then

        begin

            f [m] ← f [i] + W (i, m)

            best [m] ← i

        end
- end

Solution build algorithm

L ← (n)

m ← n

while m > 0 do

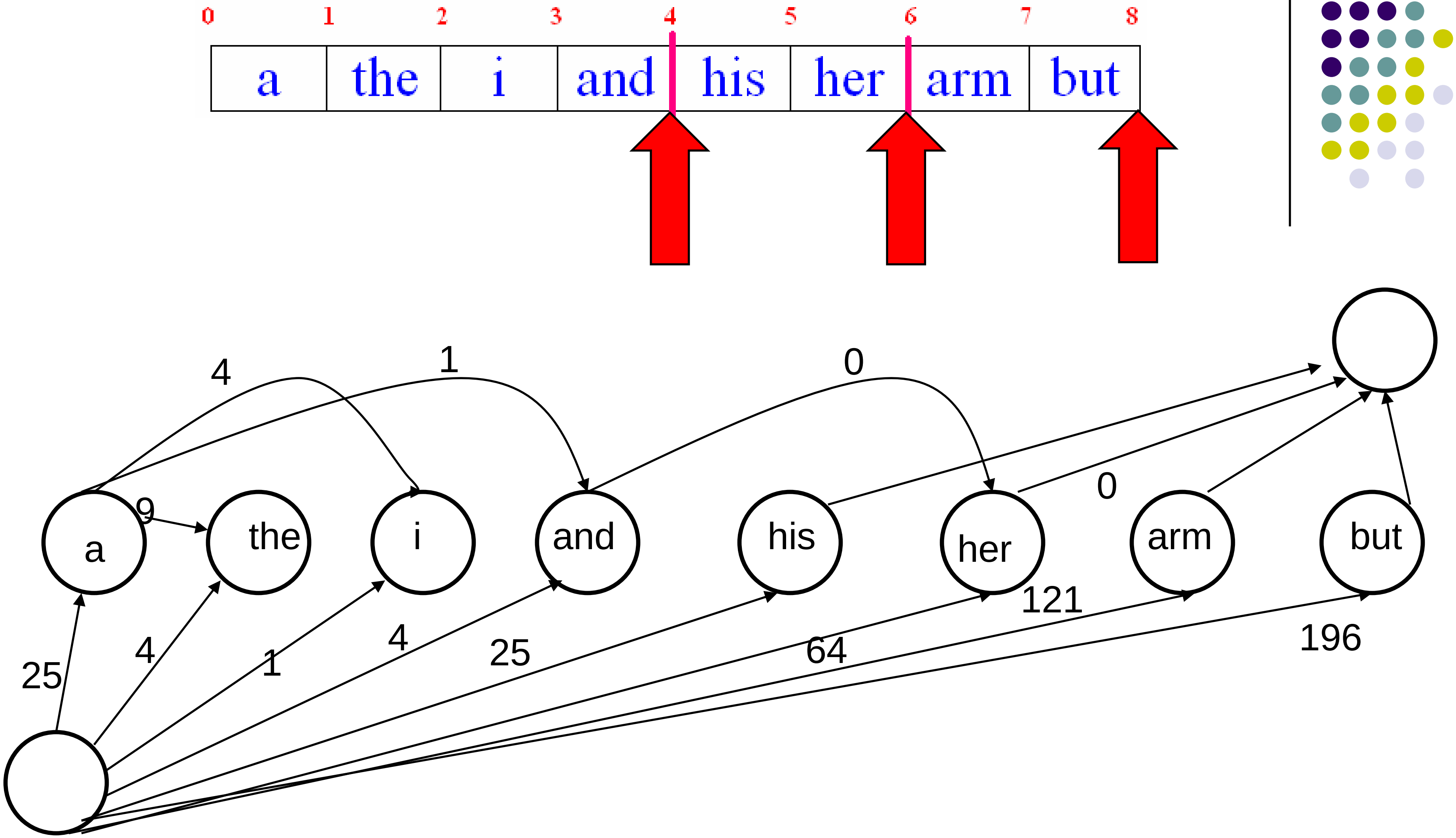
begin

  m ←best [m]

  prepend m to L

end

return (f [n] , L)



0 1 2 3 4 5 6 7 8
a the i and his her arm but
4
1
0
9
0
a the i and his her arm but
25
4
1
4
25
64
121
196

Time & Memory Complexity

- The Algorithm visits $n^2/2$ cells in the table $W(i,j)$ s.t $i<j$ → $O(n^2)$

  and then calculate $f(i)$ for every $0 \leq i \leq n$ – for $f(1)$ one action, for $f(2)$ two actions, … for $f(n)$ n actions ➔ $1+2+\ldots+n = n^2 = O(n^2)$.

  In total - time complexity is $O(n^2)$.

- The Algorithm stores nxn table → $O(n^2)$

  The size of the f and the best arrays are $O(n)$ each. In total - memory complexity is $O(n^2)$.