Lottery

First AC: Mariusz Trela, Poland (24:13)
#AC = 15

problem author: Juliusz Straszyński

Lottery

Let's focus on one query.

Computing the distance naively is $O(L)$.

There are $O(n^2)$ pairs of intervals.

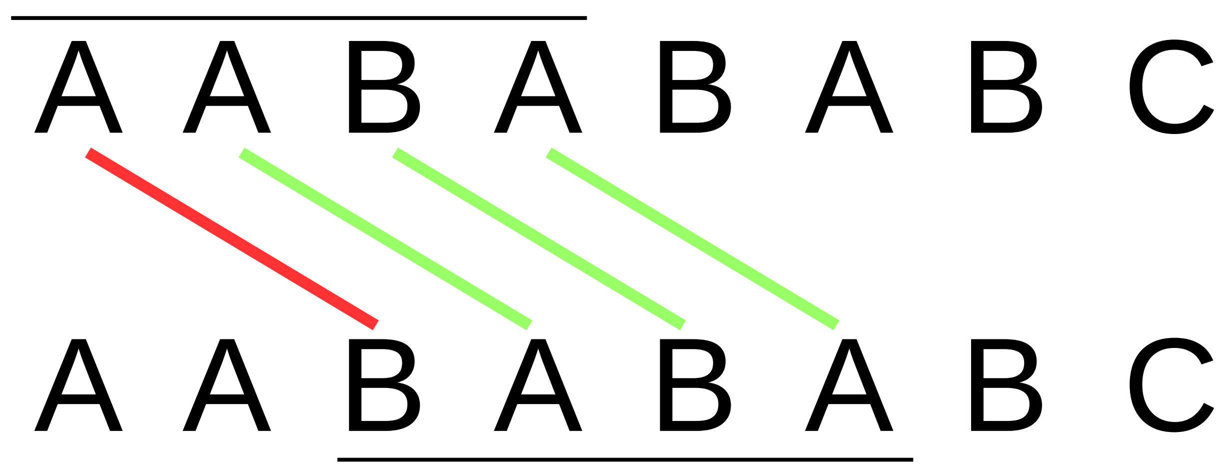
So, the brute force is $O(n^2 \times L) = O(n^3)$.

Lottery

A A B A B A B C

distance(1, 3) = 1

Lottery



distance(1, 3) = 1

Lottery

A A B A B A B C

A A B A B A B C

distance(2, 4) = 0

Lottery

$$A \quad A \quad \overline{B \quad A \quad B \quad A} \quad B \quad C$$

$$A \quad A \quad B \quad A \quad \underline{B \quad A \quad B \quad C}$$

distance(3, 5) = 1

Lottery

O(n) for every shift.

$O(n^2)$ in total to find all the distances.

How to avoid the q factor in answering queries?

We can’t store the $O(n^2)$ array!

Lottery

Queries: 2, 4, 7

0 → 2
1 → 2
2 → 2
3 → 4
4 → 4
5 → 7
6 → 7
7 → 7

Lottery

Time complexity: $O(n^2)$
Memory complexity: $O(nq)$

Cloud Computing

First AC: Costin-Andrei Oncescu, Romania (31:07)
#AC = 16

problem author: Karol Pokorski

Cloud Computing

Easiest possible version

$F_i = 1$, $f_i = 1$

$C_i = 1$, $c_i = 1$

$n = 1$ (one machine)

(consider the most profitable order)

Cloud Computing

Standard version

$F_i = 1, f_i = 1$

$\cancel{C_i = 1, c_i = 1}$

$n = 1$ (one machine)

$O(m \times c_1)$

dp[cores] – the largest profit to have so many cores

Cloud Computing

Double version

$F_i = 1$, $f_i = 1$

~~$C_i = 1$, $c_i = 1$~~

~~n = 1 (one machine)~~

two knapsacks

$O(n \times (n \times C) + m \times (m \times C))$

Cloud Computing

Double version

$F_i \leq f_i$ ← works too

~~$C_i = 1, c_i = 1$~~

~~n = 1 (one machine)~~

two knapsacks

$O(n \times (n \times C) + m \times (m \times C))$

Cloud Computing

One knapsack with modified items, e.g.:

- a task with weight 5 and value 20
- a machine with weight -7 and value -15

We must end with total weight 0 or smaller.

$O((n + m) \times (n \times C))$

Cloud Computing

Sort by f_i, F_i decreasingly.

Then just guarantee that the total weight is 0 or smaller **at every moment of time**.

$O((n + m) \times (n \times C))$

Cloud Computing

The alternative knapsack

$V_i = 1, v_i = 1$

dp[cores] → dp[money]

$O((n + m) \times n)$

Global Warming

First AC: Kacper Kluk, Poland (26:04)
#AC = 27

problem author: Kamil Dębowski

Global Warming

$$0, 3, \overset{+3}{\overline{1, 5, 2, 4, 6}}, 0, 7 \rightarrow \mathbf{0}, \mathbf{3}, \overline{\mathbf{4}, 8, \mathbf{5}, \mathbf{7}, \mathbf{9}}, 0, 7$$

$$0, 3, \overset{+3}{\overline{1, 5, 2, 4, 6, 0, 7}} \rightarrow \mathbf{0}, \mathbf{3}, \overline{\mathbf{4}, 8, \mathbf{5}, \mathbf{7}, \mathbf{9}, 0, \mathbf{10}}$$

suffix can only improve the answer!

Global Warming

$$1, \overset{-2}{\overline{5, 7}}, 6, 9 \rightarrow 1, \overline{3, 5}, 6, 9$$

$$\overset{-2}{\overline{1, 5, 7}}, 6, 9 \rightarrow \overline{-1, 3, 5}, 6, 9$$

$$1, 5, 7, \overset{+2}{\overline{6, 9}} \rightarrow 1, 5, 7, \overline{8, 11}$$

It’s enough to consider suffixes!

d = x

Global Warming

Modifying the standard LIS algorithm.

30, 60, 10, | 50, ...

1 – 10
2 – 60

Global Warming

Modifying the standard LIS algorithm.

30, 60, 10, 50, | ...

1 – 10
2 – ~~60~~ 50

Global Warming

The LDS from the right.

..., 50, 45, 75, 80

1 – 45
2 – 50
3 – 50

Global Warming

The LDS from the right.

30, 60, 10, **50**, 45, 75, 80

1 – 10
2 – 60

Global Warming

The LDS from the right.

30, 60, 10, | **50**, 45, 75, 80

1 – 10 ← longest ending with
2 – 60 a number 49 or smaller

Global Warming

The LDS from the right.

30, 60, 10, | **50**, 45, 75, 80

1 – 10
2 – 60 ← ok if x ≥ 11

Global Warming

When first going from left, just before “50”
remember the length of LIS ending with
a number 50+x-1 or smaller.

30, 60, 10, 50, ...

1 – 10
2 – 60

Thank you for your attention.

Good luck on Thursday!