HKOI 2016/17
JQ3 – Fibonacci Word

Alex Tung

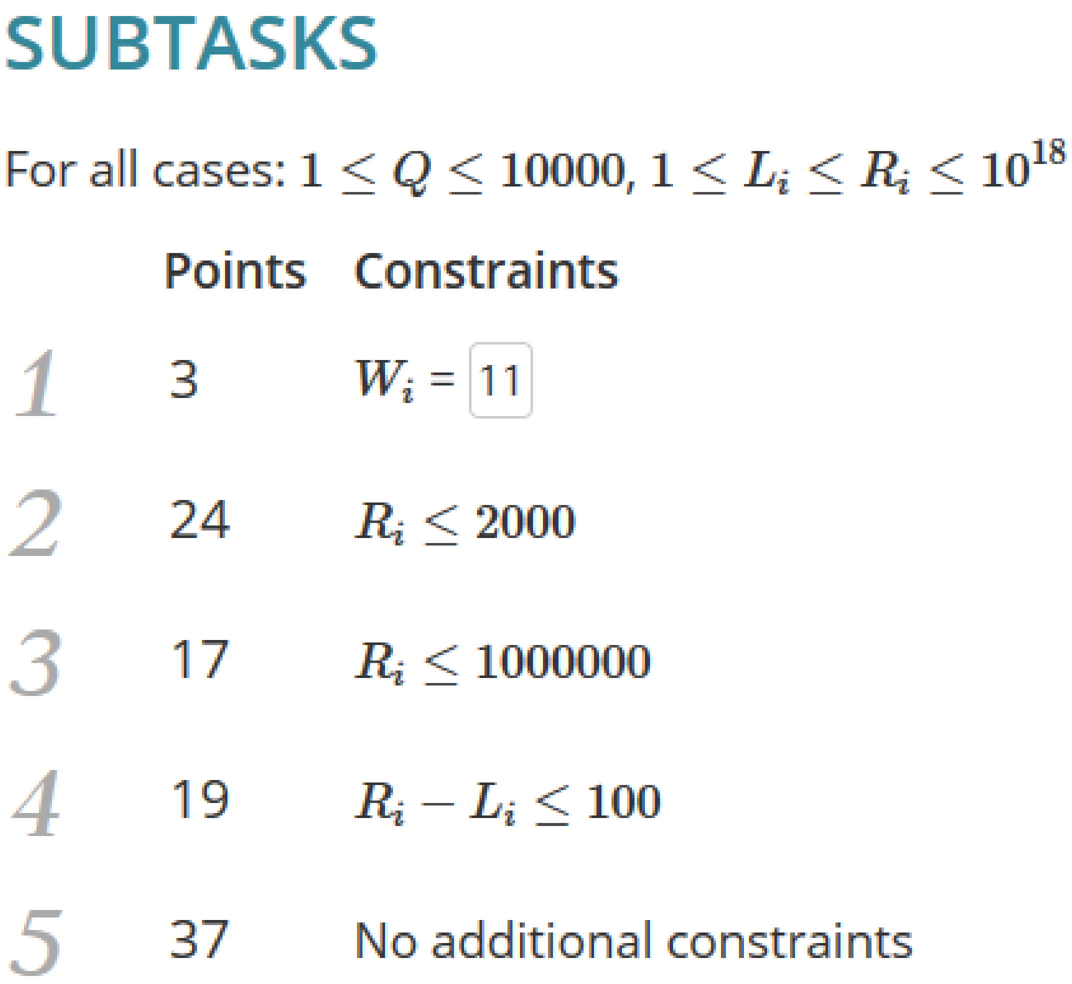
21/1/2017

Problem description

- F(1) = “0”, F(2) = “01”
- F(n+2) = F(n+1) + F(n)

- F = “010010100100101001010...”

- Find number of occurrences of W_i in F[L...R]
- W_i = “00” / “01” / “10” / “11”

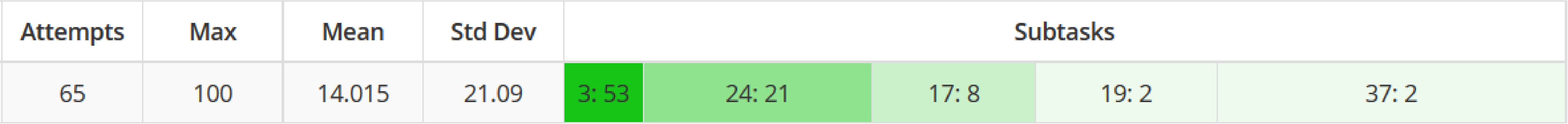


SUBTASKS

For all cases: $1 \leq Q \leq 10000$, $1 \leq L_i \leq R_i \leq 10^{18}$

	Points	Constraints
1	3	$W_i = \boxed{11}$
2	24	$R_i \leq 2000$
3	17	$R_i \leq 1000000$
4	19	$R_i - L_i \leq 100$
5	37	No additional constraints

Statistics



Attempts	Max	Mean	Std Dev	Subtasks				
65	100	14.015	21.09	3: 53	24: 21	17: 8	19: 2	37: 2

Subtask 1

- W = “11”

- Prove / Observe that **answer = 0** :P

Subtasks 2 – 3 (R small)

- Generate a prefix of F
- After 28 iterations,
  F.length() = 1346269

```
//C++ implementation

string F, temp, temp2;

F = “01”;
temp = “0”;

for(int i = 1; i <= 28; i++){
    temp2 = F + temp;
    temp = F;
    F = temp2;
}
```

O(RQ) solution

- Step 1: generate string
- Step 2: for each query, count **directly**

O(Q) solution

- Step 1: generate string
- Step 2: **precompute** the number of occurrences of “00” / “01” / “10” / “11” in F[1...K]
- Step 3: for each query (L, R, W),

  output cnt[W][R] – cnt[W][L]

(This technique is called ‘partial sum’.)

Example

- F = “010010100100101001010...”
- query(4, 12, “00”)

i	1	2	3	4	5	6	7	8	9	10	11	12
cnt[“00”][i]	0	0	0	1	1	1	1	1	2	2	2	3

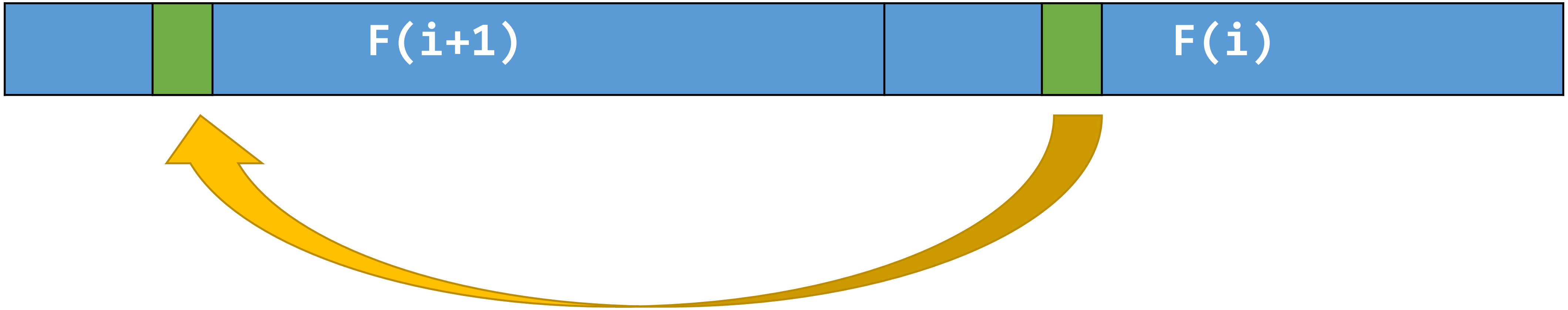
- Answer = cnt[“00”][12] – cnt[“00”][4]

  = 3 – 1

  = 2

Subtask 4 ((R – L) small)

- Idea: want to generate F[L...R], by finding out what **each character** is `len[n] := length of F(n)`



F(i+1)

F(i)

`pos -= len[i+1];`

- Time complexity: O(Q log R)

Subtask 5

- Merge ideas from subtask 3 solution and subtask 4 solution

- Idea 1: “answer = cnt[W][R] – cnt[W][L]”
- Idea 2: “pos -= len[i+1];”

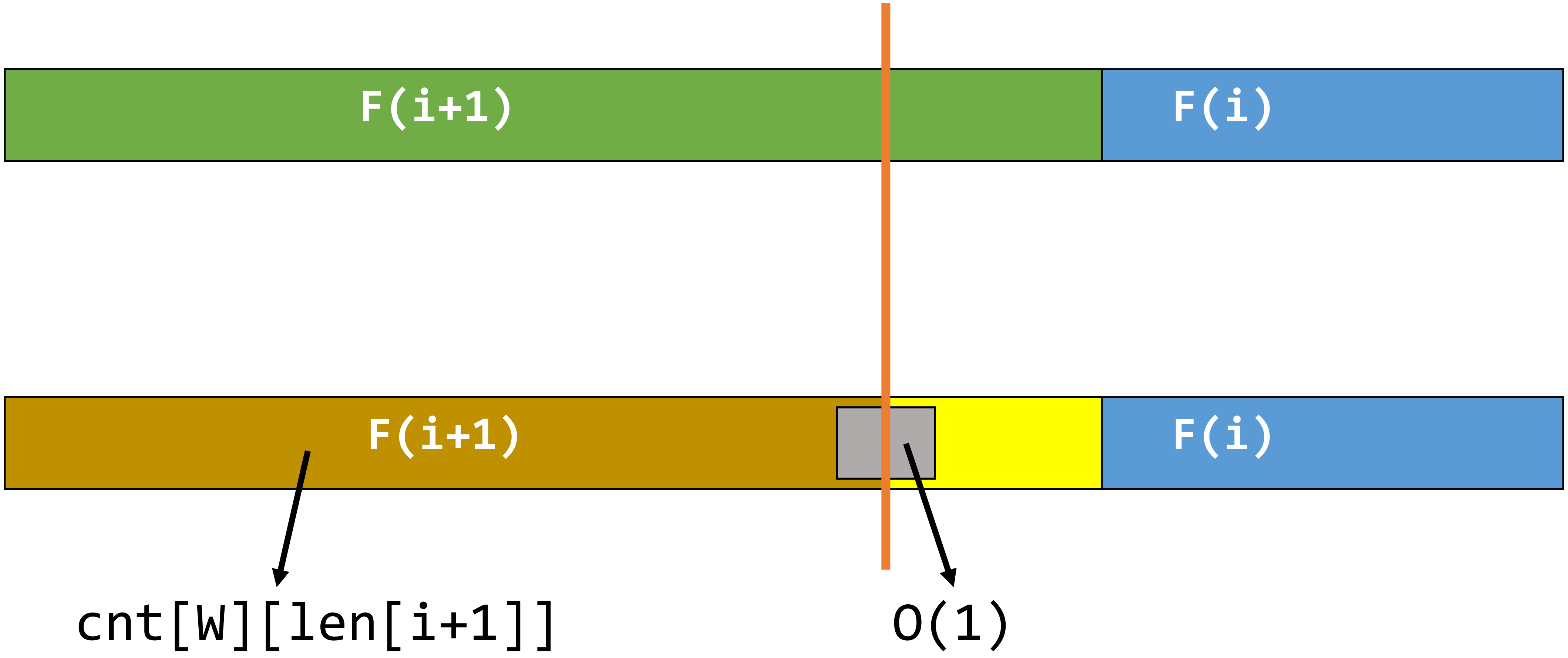
The algorithm

- Step 1: Find the values of cnt[W][len[i]] and the first and last characters of F(i)

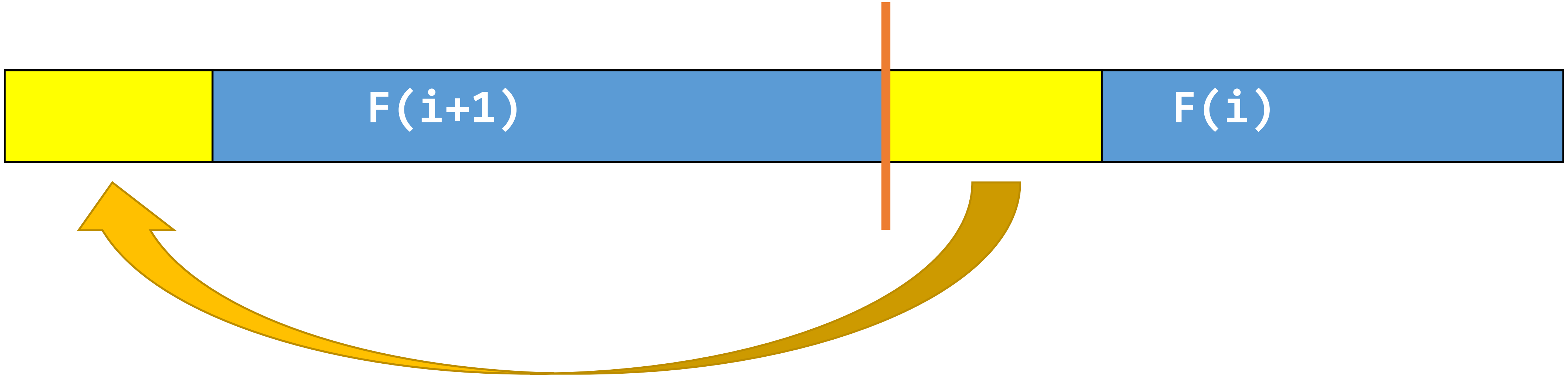
- First character: always ‘0’
- Last character: alternating between ‘0’ and ‘1’

- Step 2: Calculate query(1, R, W) by **reducing it into smaller cases** (similar for query(1, L, W))

Calculating query(1, R, W)



Calculating query(1, R, W)



- Time complexity: O(Q log R)

- To solve for R <= 10^{18}, need to find cnt[W][len[1]]...cnt[W][len[**87**]]

The End

- Any questions?