Lab4 Solution&Hint

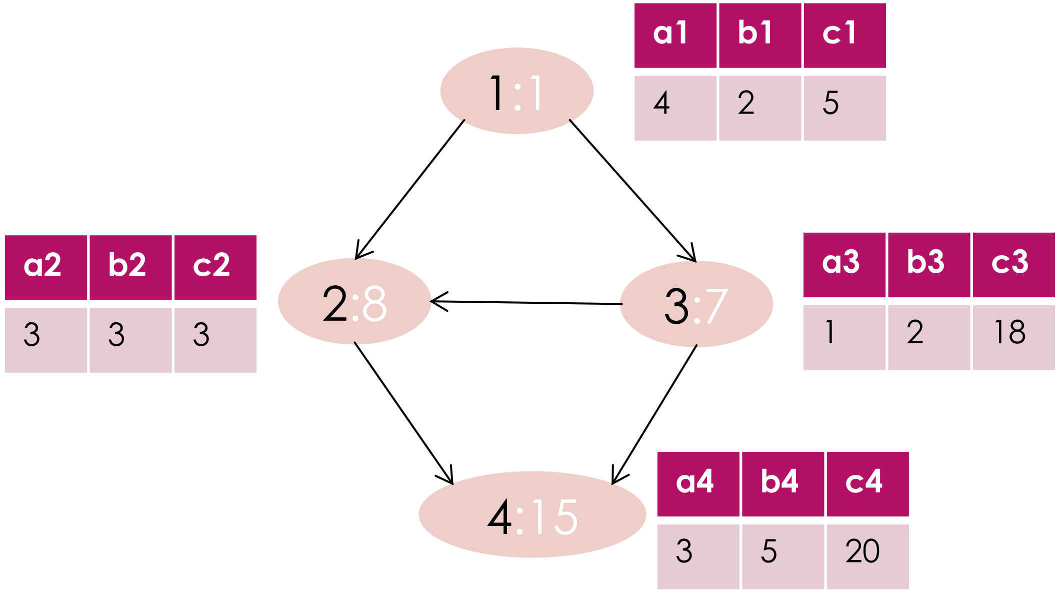
YAO ZHAO

Lab4.A: Brknight

- Recently Andrea is obsessed with a Roguelike game named Brknight.
- The map of Brknight can be seen as a directed acyclic graph (DAG) with N points and M edges, in which there exists one single starting point and multiple terminal points. A point is a terminal point iff Andrea can go nowhere from it.
- Andrea's squad is at point 1 (the starting point) and has Combat Effectiveness (CE) C initially. At point i, she would encounter a stage with dangerous level h_i, and if Andrea's CE is less than h_i, she would lose immediately. Otherwise, she would complete the stage successfully and can choose one reward among the three below:
  - Recruit an operator: Andrea's current CE would increase by $a_i (0 \leq a_i \leq 10^9)$
  - Obtain a powerful collection: Andrea's current CE would multiply by $b_i (2 \leq b_i \leq 10^9)$
  - Reset her squad: Andrea's current CE would become $c_i (0 \leq c_i \leq 10^9)$
- After that, if this point is a terminal point, Andrea will fight against a BOSS. Otherwise, she would choose an edge from point i to another point j and move to j.
- If Andrea can make her way to the BOSS, tell her the maximum possible CE module $10^9 + 7$ when she fights the BOSS.

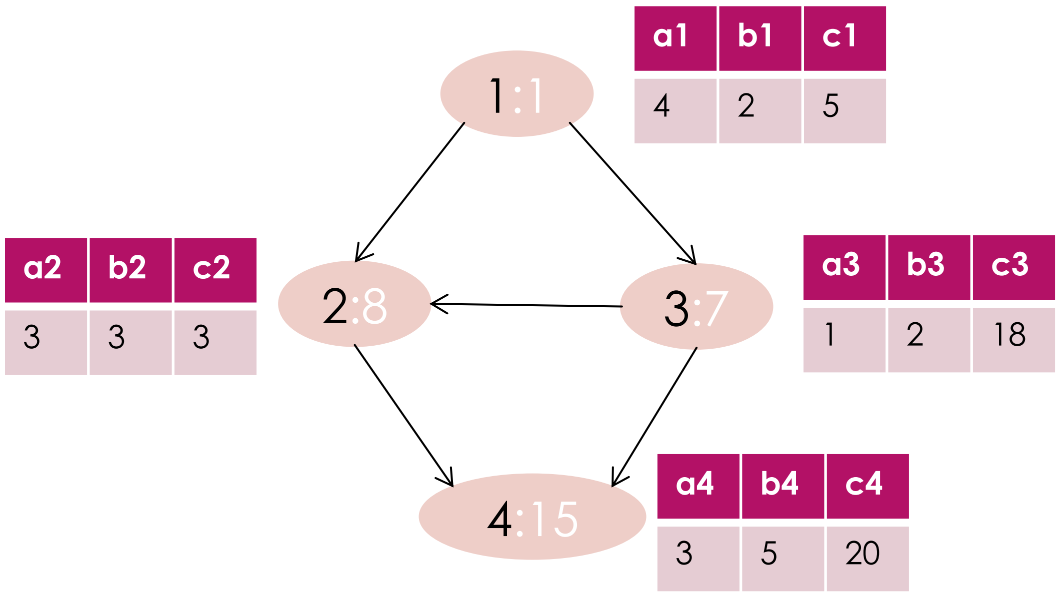
Sample Input

4 5 3
1 4 2 5
8 3 3 3
7 1 2 18
15 3 5 20
1 2
2 4
1 3
3 2
3 4



Sample Input

4 5 3
1 4 2 5
8 3 3 3
7 1 2 18
15 3 5 20
1 2
2 4
1 3
3 2
3 4



Initial c = 3

Point 1: c≥ h1
c=3 -> c=3+a1=7

Move to Point 3: c≥h3
c=7->c=c3=18

Move to Point 2: c≥h2
c=18->c=18*b2=54

Move to Point 4: c≥h4
c=54->c=54*b4=270

Sample Output 1

270

$0 \le a_i \le 10^9$
$2 \le b_i \le 10^9$
$0 \le c_i \le 10^9$
$0 \le h_i \le 10^9$
$0 \le C \le 10^9$
$1 < N \le 10000$

int: -2147483648~2147483647[-2^31~2^31-1] 10^9

long: -9223372036854775808~9223372036854775807 [-2^63~2^63-1] 10^{18}

The CE maybe become a huge number: $\underbrace{10^9 * 10^9 \ldots * 10^9}_{10001} = 10^{90009}$

How to accumulate and compare?

Notice:

$$0 \le a_i \le 10^9$$
$$2 \le b_i \le 10^9$$
$$0 \le c_i \le 10^9$$
$$0 \le h_i \le 10^9$$

If CE ≥ 10^9, always win the point i and choose b_i, then CE = CE* b_i
If CE < 10^9, always choose the maximum of CE+ a_i, CE* b_i , c_i

We can use **log** to compare since CE maybe become a huge number
If CE ≥ 10^9， CE_log = CE_Log+log b_i
If CE < 10^9, CE_log = log(max(CE+ a_i, CE* b_i , c_i))

Try to find all topo order and get the maximum answer

```
put node 1 to queue Q
While Q is not empty:
   get a node e from Q
   if the actual CE >= e.h:
      choose the best reward and update e.CE_mod and e.CE_log
      if e.out_degree ==0 record to the answer and the answer_log
      for each child c of node:
         c.indegree ← c.indegree -1
         if (e.CE_log > c.CE_log): update c.CE_mod and c.CE_log
         if c.indegree == 0 :put c to Q
   else:
      for each child c of node:
         c.indegree ← c.indegree -1
Find the maximum answer_log and output the responds answer
```

Lab4.B: Precarious Computing

- Sky has N expendable computers $1,2,\ldots,N$ lining up from left to right.
- The i^{th} computer can only be powered up for a_i seconds and generates b_i income per second when powered. If the total workload exceeds a_i seconds, the computer will blow up Sky's bedroom. It is guaranteed that a_i is unique.
- Sky has a special cable. He can only power up a consecutive interval of computers at a time. Sky can change the powered interval at anytime.
- Unfortunately, the cable is also expendable and can only function for M seconds. How much income can Sky obtain if he operates optimally? Sky does not want any splendid explosion since he still needs the bedroom to sleep.

Sample Input 1

5 3
1 3 5 2 4
3 5 5 3 1

1 3 5 2 4
3 5 5 3 1

1s, computer 1..5 income:17

0 2 4 1 3
3 5 5 3 1

1s, computer 2..5 income:14

0 2 4 1 3
3 5 5 3 1

1s, computer 2..3 income:10

0 1 3 0 2
3 5 5 3 1

Total: 41

Sample Output 1
41

Sample Input 2

10 30
44 32 6 1 7 18 27 21 40 48
97 39 32 23 18 32 34 34 21 5

44 32 6 1 7 18 27 21 40 48
97 39 32 23 18 32 34 34 21 5

1s, computer 1..10, income:335

43 31 5 0 6 17 26 20 39 47
97 39 32 23 18 32 34 34 21 5

5s, computer 1..3, income:168*5

38 26 0 0 6 17 26 20 39 47
97 39 32 23 **18 32 34 34 21 5**

6s, computer 5..10, income:144*6

38 26 0 0 0 11 20 14 33 41
97 39 32 23 18 32 34 34 21 5

18s, computer 1..2, income:136*18

20 8 0 0 0 11 20 14 33 41
97 39 32 23 18 32 34 34 21 5

1s+5s+6s+18s = 30s
Total income: 4487

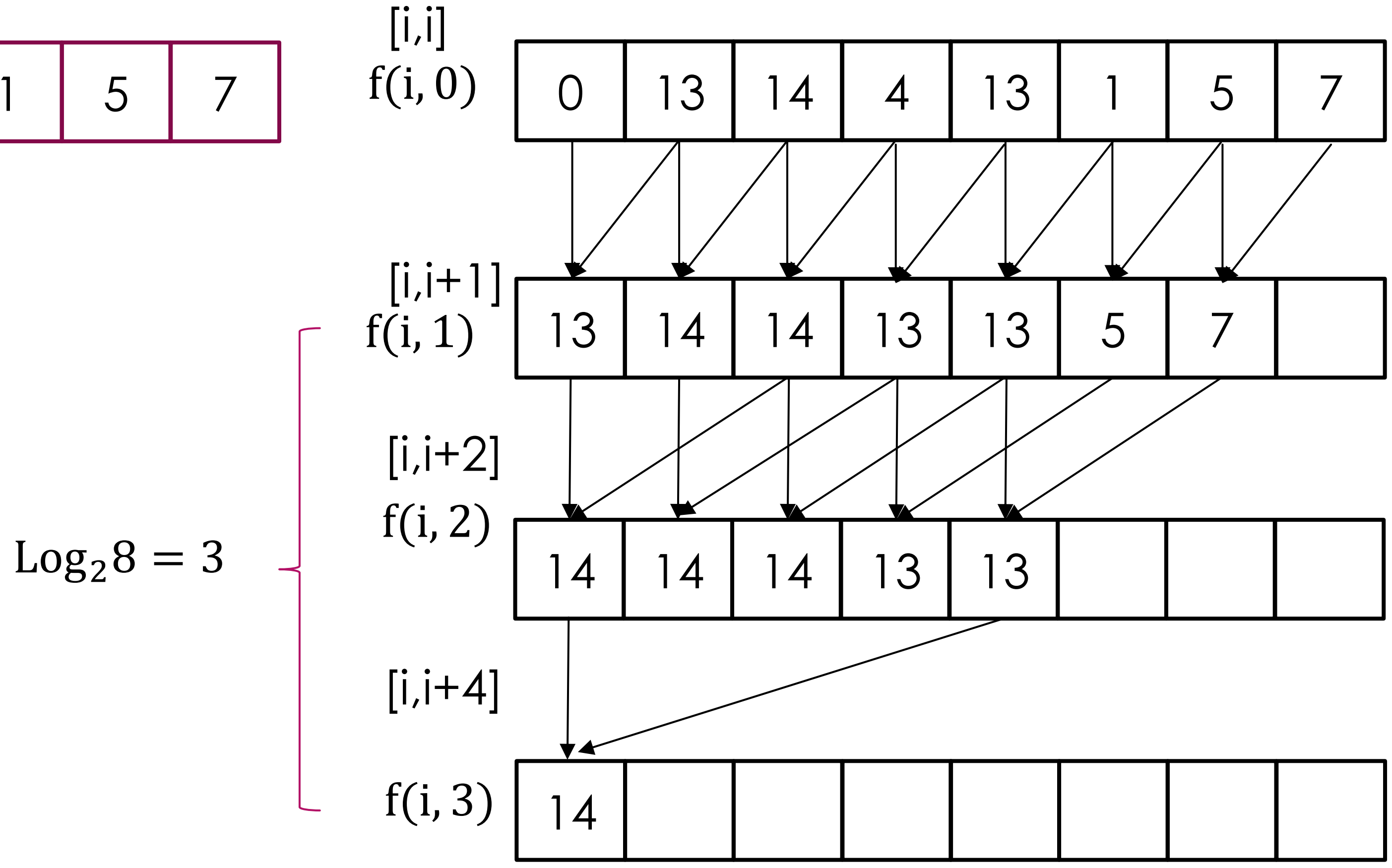
Sample Output 2
4487

ST table

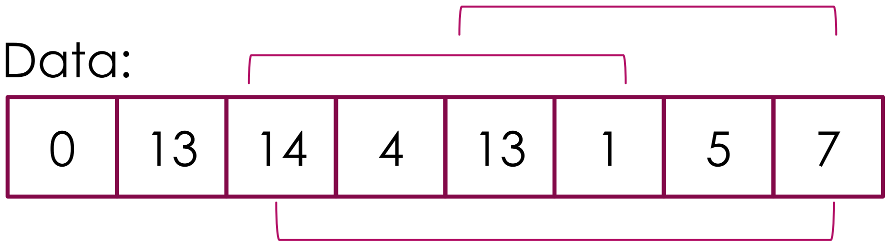
Creating ST table:

Data:

0	13	14	4	13	1	5	7



Query:



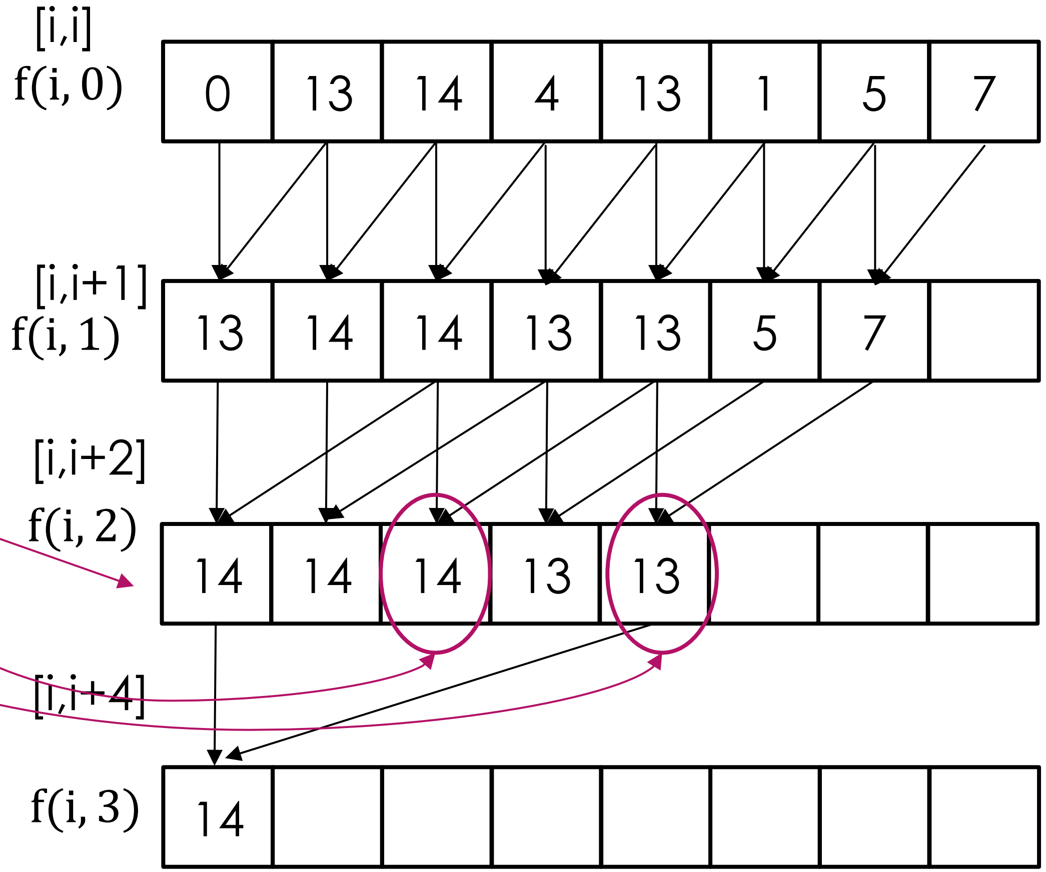
Maximum Query:
Range: [2,7]

S=$\lfloor log_2(7-2+1) \rfloor = 2$

L=2, index1 =2

R=7,index2 =7-2^2+1=4

Max(14,13) = **14**



https://oi-wiki.org/ds/sparse-table/

Notice: $0 \le a_i \le 10^9$ a_i is unique
$1 \le b_i \le 10^4$

First, use ST to get the minimum in a_i [1, N]

Sample Input 2

10 30
44 32 6 1 7 18 27 21 40 48
97 39 32 23 18 32 34 34 21 5

Mini a_i : 1→min_a
Mini a_i index : 4 →min_index

divide to 2 intervals: [1,min_index-1] and [min_index+1,N]
add a property base_value = min_a for above intervals
Income += min_a*sum(1,N)

44 32 6 | 1 | 7 18 27 21 40 48
97 39 32 | 23 | 18 32 34 34 21 5

Compare sum(1,min_index-1) and sum(min_index+1,N)
Get the larger one, for example [1,min_index-1]
Query the min_a in [1,min_index-1]
Income += (min_a-base_value)*sum(1, min_index-1)
Continue divide to 2 intervals
add a property base_value = min_a for above intervals

44 32 | 6 | 1 | 7 18 27 21 40 48
97 39 | 32 | 23 | 18 32 34 34 21 5

…

How to improve efficiency?

Query the min_a in [l,r] → ST
Add a property like base_value, you don't need to modify ai

Query any range sum → Prefix sum (Lab4:算法常用解题技巧.pdf)

Get the largest Prefix sum → Priority queue(Maximum heap)