JOIG2022/2023 春季トレーニング 競技2日目

Smartphone解説

清水健吾 (Thistle)

問題概要

- N 台のスマホがあり、i番目のスマホは区間 $[A_i, B_i]$ のうち C_i 日間使える。
- $[1,K]$ の内最大何日間スマホを使えるか？

制約

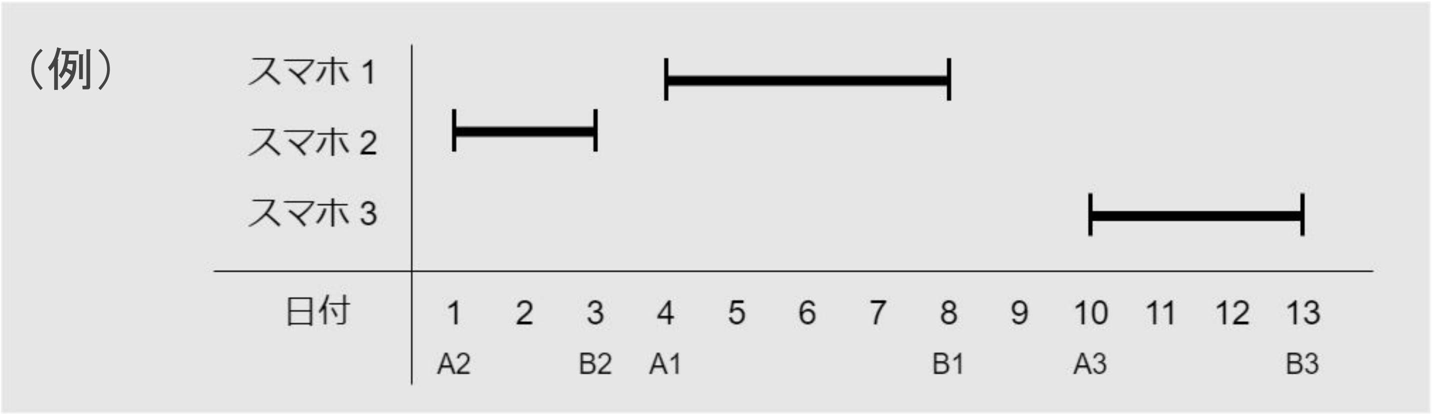
小課題	特殊制約	N	K
1	$B_i<A_j$ or $B_j<A_i$ $(i<j)$	1000	1000
2	$C_i=B_i-A_i+1$	300000	10^15
3	$A_i=1$	1000	1000
4		1000	1000
5		300000	300000
6 (満点)		300000	10^15

小課題 1

制約 **$B_i<A_j$ or $B_j<A_i$ ($i<j$)** を読み解こう。

$A_i<B_i$, $A_j<B_j$ と合わせると……

$(A_i<B_i) < (A_j<B_j)$ or $(A_j<B_j) < (A_i<B_i)$ ⇒ 区間が被っていない！



小課題 1

他のスマホのことを考えずに使うことが出来る。

スマホを Ci 日未満使うメリットはないため、全て使うことにする。

⇒ Ci の総和を出力すればよい

⇒ **AC!**

小課題 2

制約は **Ci = Bi-Ai+1 ⇒ 区間全てで使うのが最適**

問題文の言い換えをすると、

N 個の区間が与えられる。1 つ以上の区間で被覆された部分の長さの和を求めよ

よくわからないので、とりあえず愚直解を考えてみよう。

- 長さ K の配列 D を用意し、全ての i に対して Ai <= x <= Bi である D[x] に 1 を足す O(NK)
- 値が 1 以上の要素の数を数える　O(K)

小課題 2

愚直解を高速化していこう。

まず、1step目では、**区間**に対して**均一に** 1 を加算するということをしている。

これは、**imos法を使うことで高速にできる**。（セグ木でも出来る）

これで、

- 長さ K の配列 D を用意し全ての i に対して Ai <= x <= Bi である D[x] に 1 を足す　**O(K)**
- 値が 1 以上の要素の数を数える　O(K)

となった。

小課題 2

区間への加算の様子をよく見てみると、**Ai,Bi 周辺以外では連続な場所は値が同じである**、ということが分かる。

例 :

D : 000000 1111 22222 11111 33333 2222 1111 0000 ....

A0 A1 B1 A2,A3 B2 B0 B3

⇒ 値が同じ区間については持っておく情報を 1 つに減らせる。

＝ **座標圧縮**によって K 日を O(N) 区間にまで減らせる。

小課題 2

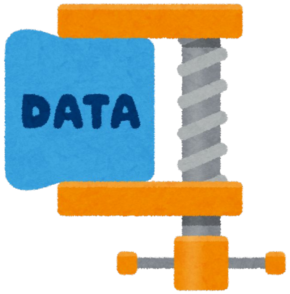
これによって愚直解の O(K) 部分が N に依存する計算量に変わり、

- **座標圧縮をする** **O(N log N)**
- 長さ **O(N)** の配列 D を用意し全ての i に対して Ai <= x < Bi+1 である D[x] に 1 を足す **O(N log N)**
- 値が 1 以上の要素の **D[x+1]-D[x] の総和を求める** **O(N)**

となる。

座標圧縮時の処理と合わせて **O(N log N)** で解くことが出来た。

小課題 2



このように、座標圧縮は、

座標の制約の値が大きく、かつ、

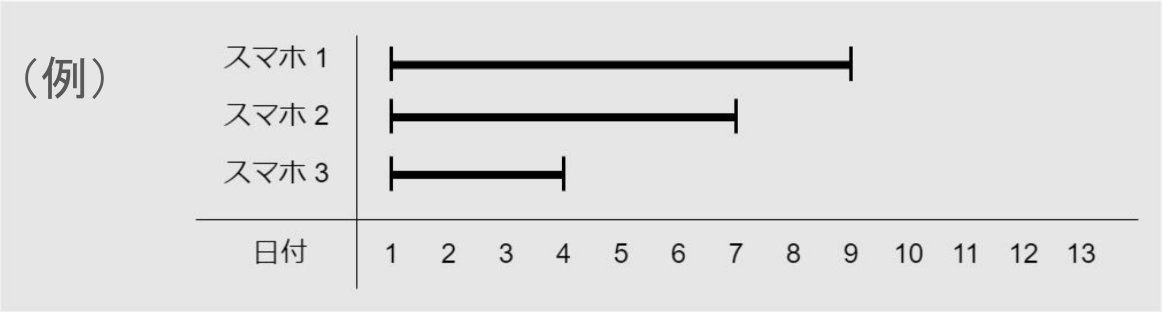
入力で与えられた数値で分割した区間の中では操作や性質が同じ時

に使うことが出来る。

このような問題を見たら反射で使えるようにしておくとよいだろう。

小課題 3

制約 Ai = 1 の場合の状況を考える。Bi が大きい順でソートすると、



のようになる。

ここで重要な考察は、

スマホの使用可能区間同士は完全な包含関係にあるということ。

小課題 3

この場合、各日について、使用日が余っているスマホのうち
早く使えなくなる＝Bi が小さい順に使うという貪欲法が最適である。

直感的には、Bi が大きいものは後でも使えるから後に回した方が
得しそうという感じ。

ある日に使えるスマホが2つあったとして、Bi が大きいスマホは Bi が小さい
スマホの代わりに必ず使えるが、逆は出来ない日がある。

小課題 3

<u>貪欲法の厳密な証明</u>

スマホの使用日数が最大となるあるスマホの使い方において、

1. B_i < B_j なスマホ i, j について、k 日目にスマホ i の使用日数が残っているにも関わらずスマホ j を使ったとする

   ・もし k+1 日目以降でスマホ i を使っていないなら、k 日目はスマホ j の代わりにスマホ i を使う

   ・もし k+1 日目以降でスマホ i を使っているなら、その日と使うスマホを交換する

2. k 日目にスマホ i の使用日数が残っているにも関わらずどのスマホも使わなかったとする

   ・もし k+1 日目以降でスマホ i を使っていないなら、なにもしない

   ・もし k+1 日目以降でスマホ i を使っているなら、スマホ i を使っている日をその日と交換する

この交換を繰り返すことで、Biが小さく使用日数が余っているスマホから貪欲に選ぶ使い方に帰着される

ソートがボトルネックとなり、**O(K + N log N)** でこの小課題が解けた。

小課題 3

各日について考えるということをしなくても、この小課題だけなら

区間は終点でソート

という格言に従い、終点の早い区間から順に、前から C_i 日割り当てていく、
という方法でも解くことが出来る。

小課題 4　: N, K <= 1000

特殊な制約はない。

小課題 3 で行った考察は $A_i = 1$ でなくても成立する。

よって、各日について、今使えるスマホの中で最も B_i が小さいものを選んで使用する、という貪欲法で解くことが出来る。

計算量は **O(NK)**

小課題 5　: N, K <= 300000

N と K の値が大きくなり、O(NK) では通らなくなった。

各日について、というところは変えられないので、O(K) は削れない。

そこで、最も Bi が小さいスマホを選ぶ、というパートを削減することにしよう。

こういったときに使用できるデータ構造に、**priority_queue** というものがある。

・1 日目から順に探索し、Ai 日目になったら prique にスマホ i を入れる
・その日使うスマホを選ぶため、prique 内で Bi 最小のスマホを確認する
　もし使えなかったらそのスマホを取り出し、次のスマホを確認

とすることで全体で **O(K + N log N)** で実行できる。

小課題 6 (満点)　: N <= 300000, K <= 10^15

K が大きくなったため、計算量に K が入ることも許されなくなった。

この問題の特徴は、

・K の制約が大きすぎる
・Ai,Bi で区切られた区間内の日の性質は均質である

ということで、**座標圧縮**をしよう。
計算量は **O(N log N)** となり、この問題を解くことが出来た
🎉🎉

得点分布