გასწრება

ბუდაპეშტის აეროპორტიდან სასტუმრო Forrás-მდე ერთზოლიანი, ცალმხრივი გზა არსებობს, რომლის სიგრძე L კილომეტრია.

IOI 2023 ოლიმპიადის მიმდინარეობის დროს ამ გზაზე $(N + 1)$ რაოდენობის სატრანსფერო ავტობუსი მოძრაობს. ავტობუსები გადანომრილია 0-დან N-მდე. i-ური ავტობუსი $(0 \leq i < N)$ აეროპორტს ოლიმპიადის მიმდინარეობის $T[i]$-ურ წამზე ტოვებს და 1 კილომეტრს $W[i]$ წამში გადის. N-ური ავტობუსი სარეზერვო ავტობუსია და 1 კილომეტრს X წამში გადის. Y დრო, როცა ის აეროპორტს დატოვებს, ჯერ კიდევ არ არის განსაზღვრული.

საერთოდ გზაზე გასწრება აკრძალულია, მაგრამ ავტობუსებს უფლება აქვთ ერთმანეთს **დამხარისხებელ სადგურებზე** გაუსწრონ. გზაზე M რაოდენობის დამხარისხებელი სადგურია, რომლებიც გადანომრილია 0-დან $(M - 1)$-მდე. დამხარისხებელი სადგური j $(0 \leq j < M)$ გზაზე აეროპორტიდან $S[j]$ კილომეტრზე მდებარეობს. დამხარისხებელი სადგურები დასორტირებულია აეროპორტიდან დაშორების მიხედვით, ანუ, $S[j] < S[j + 1]$ ყველა $(0 \leq j \leq M - 2)$-თვის. პირველი დამხარისხებელი სადგური თვითონ აეროპორტია, ხოლო ბოლო კი სასტუმროა. ანუ, $S[0] = 0$ და $S[M - 1] = L$.

თითოეული ავტობუსი მაქსიმალური სიჩქარით მოძრაობს მანამ, სანამ გზაზე ის არ დაეწევა მის წინ მიმავალ უფრო დაბალი სიჩქარით მოძრავ ავტობუსს. ასეთ შემთხვევაში ისინი ჯგუფდებიან და იძულებული არიან იმოძრავონ უფრო ნელა მოძრავი ავტობუსის სიჩქარით მანამ, სანამ არ მიაღწევენ მომდევნო დამხარისხებელ სადგურს, სადაც უფრო სწრაფი ავტობუსები გაასწრებენ უფრო ნელა მოძრავ ავტობუსებს.

ფორმალურად, ყველა ისეთი i და j-სათვის, სადაც $0 \leq i \leq N$ და $0 \leq j < M$, დრო $t_{i,j}$ (წამებში), როცა ავტობუსი i **მიდის** დამხარისხებელ სადგურში j, განისაზღვრება შემდეგნაირად. ვთქვათ $t_{i,0} = T[i]$ ყოველი $0 \leq i < N$ და ვთქვათ $t_{N,0} = Y$. ყოველი j-სათვის, სადაც $0 < j < M$:

- განვსაზღვროთ i-ური ავტობუსის j-ურ დამხარისხებელ სადგურში **მოსალოდნელი მისვლის დრო** $e_{i,j}$ (წამებში), როგორც დრო, რომელშიც i-ური ავტობუსი მივიდოდა j-ურ დამხარისხებელ სადგურში, მას რომ მაქსიმალური სიჩქარით ევლო $(j - 1)$-ე სადგურში მისვლის მომენტიდან. ანუ, ვთქვათ
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j - 1])$ თითოეული $0 \leq i < N$ და
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j - 1])$.
- i-ური ავტობუსი მიდის j-ურ დამხარისხებელ სადგურში *მაქსიმალურ დროში* i-ური ავტობუსის და ყველა იმ სხვა ავტობუსის მოსალოდნელ მისვლის დროებს შორის, რომლებიც $(j - 1)$-ე

სადგურში i-ურ ავტობუსზე უფრო ადრე მივიდნენ. ფორმალურად, ვთქვათ $t_{i,j}$ არის მაქსიმალური $e_{i,j}$-სა და ყველა $e_{k,j}$-ებს შორის, სადაც $0 \leq k \leq N$ და $t_{k,j-1} < t_{i,j-1}$.

IOI-ს ორგანიზატორებს სურთ დაგეგმონ სარეზერვო ავტობუსის (N-ური ავტობუსის) მოძრაობის განრიგი. თქვენი ამოცანაა უპასუხოთ ორგანიზატორების Q რაოდენობის კითხვას, რომლებიც შემდეგი ფორმით არიან დასმული: თუ სარეზერვო ავტობუსმა აეროპორტი უნდა დატოვოს მოცემული Y დროისათვის (წამებში), მაშინ რა დროს მივა იგი სასტუმროში?

იმპლემენტაციის დეტალები

თქვენ იმპლემენტაცია უნდა გაუკეთოთ შემდეგ პროცედურას.

```
void init(int L, int N, int64[] T, int[] W, int X, int M, int[] S)
```

- L: გზის სიგრძე.
- N: არასარეზერვო ავტობუსების რაოდენობა.
- T: N სიგრძის მასივი, რომელიც აღწერს არასარეზერვო ავტობუსების აეროპორტიდან გამოსვლის დროებს.
- W: N სიგრძის მასივი, რომელიც აღწერს არასარეზერვო ავტობუსების მაქსიმალურ სიჩქარეებს.
- X: დრო, რომელიც სჭირდება არასარეზერვო ავტობუსს 1 კილომეტრის გასავლელად.
- M: დამხარისხებელი სადგურების რაოდენობა.
- S: M სიგრძის მასივი, რომელიც აღწერს დამხარისხებელი სადგურების აეროპორტიდან დაშორებებს.
- ეს პროცედურა გამოძახებული იქნება ზუსტად ერთხელ თითოეული ტესტისათვის, იქამდე, სანამ არ მოხდება `arrival_time` გამოძახება.

```
int64 arrival_time(int64 Y)
```

- Y: დრო, რომელზეც სარეზერვო ავტობუსმა (ავტობუსი N) უნდა დატოვოს აეროპორტი.
- ამ პროცედურამ უნდა დააბრუნოს დრო, რომელზეც სარეზერვო ავტობუსი მივა სასტუმრომდე.
- ეს პროცედურა გამოიძახება ზუსტად Q-ჯერ.

მაგალითი

განვიხილოთ გამოძახებების შემდეგი მიმდევრობა:

```
init(6, 4, [20, 10, 40, 0], [5, 20, 20, 30], 10, 4, [0, 1, 3, 6])
```

ავტობუსი 4-ის გამოტოვებით (რომელიც ჯერ არ დაგეგმილა), ქვემოთ მოცემული ცხრილი გიჩვენებთ არასარეზერვო ავტობუსების მოსალოდნელ და რეალურ მისვლის დროებს თითოეულ დამხარისხებელ სადგურამდე:

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	55
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180

სადგურ 0-მდე მისვლის დროები არის დროები, როცა დაგეგმილია ავტობუსების აეროპორტიდან გამოსვლა. ანუ, $t_{i,0} = T[i]$ ყველა $(0 \leq i \leq 3)$-სათვის.

მოსალოდნელი და რეალური დროები სადგურ 1-მდე მისასვლელად გამოითვლება შემდეგნაირად:

- მოსალოდნელი დროები სადგურ 1-მდე:
  - ავტობუსი 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - ავტობუსი 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - ავტობუსი 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - ავტობუსი 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- რეალური დროები სადგურ 1-მდე:
  - ავტობუსები 1 და 3 სადგურ 0-მდე მივლენ უფრო ადრე, ვიდრე ავტობუსი 0, შედეგად $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - ავტობუსი 3 მივა სადგურ 0-მდე უფრო ადრე, ვიდრე ავტობუსი 1, შედეგად $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - ავტობუსები 0, 1 და 3 მივლენ სადგურ 0-მდე უფრო ადრე ვიდრე ავტობუსი 2, შედეგად $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - არცერთი ავტობუსი არ მივა სადგურ 0-მდე უფრო ადრე, ვიდრე ავტობუსი 3, შედეგად $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

ავტობუს 4-ს სჭირდება 10 წამი, რომ გაიაროს 1 კილომეტრი და მისი აეროპორტიდან გასვლა დაგეგმილია 0 წამზე. ამ შემთხვევაში, შემდგომი ცხრილი გვიჩვენებს მისვლის დროებს. ერთადერთი ცვლილება, რომელიც მოსალოდნელ და რეალურ დროებს შეეხო არის ხაზგასმული.

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	$\underline{60}$
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180
4		0		10	10		30	30		60	60

ვხედავთ, რომ ავტობუსი 4 მიდის სასტუმროსთან 60 წამზე. შესაბამისად პროცედურამ უნდა დააბრუნოს 60.

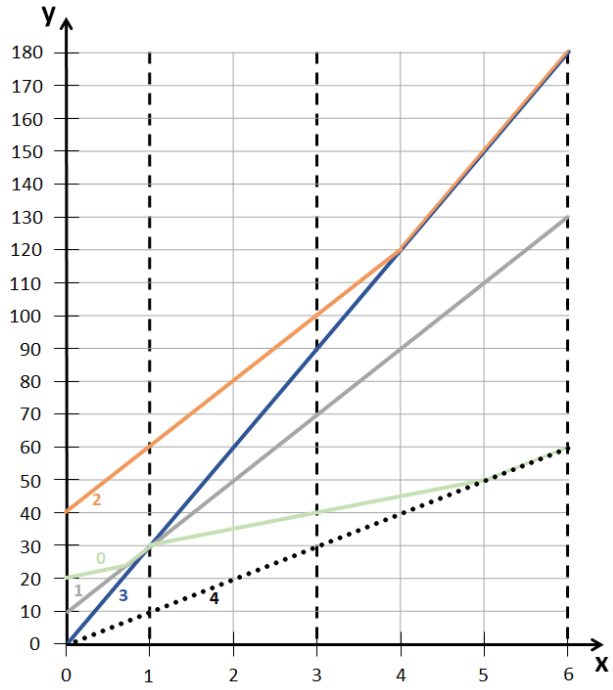
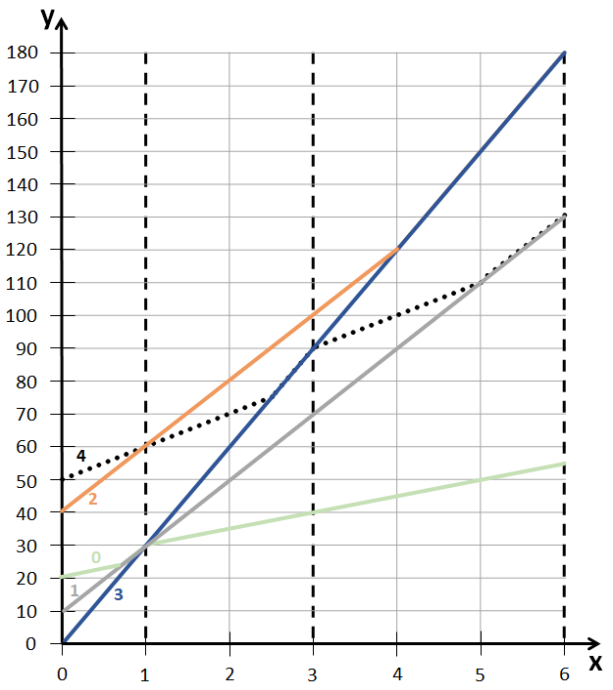
```
arrival_time(50)
```

ავტობუსი 4-ის აეროპორტიდან გასვლა ახლა დაგეგმილია 50 წამზე. ამ შემთხვევაში, არ გვექნება ცვლილებები არასარეზერვო ავტობუსების მისვლის დროებში საწყის ცხრილთან შედარებით. მისვლის დროები ნაჩვენებია შემდეგ ცხრილში.

i	$t_{i,0}$	$e_{i,1}$	$t_{i,1}$	$e_{i,2}$	$t_{i,2}$	$e_{i,3}$	$t_{i,3}$
0	20	25	30	40	40	55	55
1	10	30	30	70	70	130	130
2	40	60	60	100	100	160	180
3	0	30	30	90	90	180	180
4	50	60	60	80	90	120	130

ავტობუსი 4 გადაასწრებს უფრო ნელ ავტობუს 2-ს სადგურ 1-ში, სადაც ისინი ერთდროულად მივლენ. შემდეგ, ავტობუსი 4 ჯგუფდება ავტობუს 3-თან სადგურ 1-სა და 2-ს შორის, რაც იწვევს ავტობუსი 4-ის მისვლას სადგურ 2-ში 90 წამზე 80-ის ნაცვლად. სადგური 2-დან გასვლის შემდეგ, ავტობუსი 4 ჯგუფდება ავტობუს 1-თან სანამ ისინი არ მივლენ სასტუმროსთან. ავტობუსი 4 მივა სასტუმროში 130 წამზე. რაც ნიშნავს რომ პროცედურამ უნდა დააბრუნოს 130.

შეგვიძლია დავხაზოთ დრო, რომელიც თითოეულ ავტობუსს დასჭირდება მისასვლელად თითოეულ დისტანციაზე აეროპორტიდან. ნახაზზე x ღერძი წარმოადგენს აეროპორტიდან დისტანციას (კილომეტრებში) და y ღერძი წარმოადგენს დროს (წამებში). ვერტიკალური წყვეტილი ხაზებით მონიშნულია დამხარისხებელი სადგურები. დანარჩენი წყვეტილი ხაზები აღნიშნავს არასარეზერვო ავტობუსებს. შავი წერტილოვანი ხაზი გვიჩვენებს სარეზერვო ავტობუსს.

arrival_time(0)	arrival_time(50)
	

შეზღუდვები

- $1 \leq L \leq 10^9$
- $1 \leq N \leq 1\,000$
- $0 \leq T[i] \leq 10^{18}$ (ყველა i-სთვის, სადაც $0 \leq i < N$)
- $1 \leq W[i] \leq 10^9$ (ყველა i-სთვის, სადაც $0 \leq i < N$)
- $1 \leq X \leq 10^9$
- $2 \leq M \leq 1\,000$
- $0 = S[0] < S[1] < \cdots < S[M-1] = L$
- $1 \leq Q \leq 10^6$
- $0 \leq Y \leq 10^{18}$

ქვეამოცანები

1. (9 ქულა) $N = 1, Q \leq 1\,000$
2. (10 ქულა) $M = 2, Q \leq 1\,000$
3. (20 ქულა) $N, M, Q \leq 100$
4. (26 ქულა) $Q \leq 5\,000$
5. (35 ქულა) დამატებითი შეზღუდვების გარეშე.

სანიმუშო გრადერი

სანიმუშო გრადერი კითხულობს ინფორმაციას შემდეგი ფორმატით:

- სტრიქონი 1: $L\ N\ X\ M\ Q$
- სტრიქონი 2: $T[0]\ T[1]\ \ldots\ T[N-1]$
- სტრიქონი 3: $W[0]\ W[1]\ \ldots\ W[N-1]$
- სტრიქონი 4: $S[0]\ S[1]\ \ldots\ S[M-1]$
- სტრიქონი $5+k$ ($0 \le k < Q$): Y შეკითხვა k-სთვის

სანიმუშო გრადერს გამოაქვს შედეგი შემდეგი ფორმატით:

- სტრიქონი $1+k$ ($0 \le k < Q$): `arrival_time`-ს მიერ დაბრუნებული მნიშვნელობა k-ური შეკითხვისათვის.