JOIG 2021/2022 春合宿 競技2 2問目

Airport 解説

大佐 健人

問題概要

- 空港に、N 本の滑走路がある
- M 個の着陸計画がある（使う滑走路は自由に決められる）
- 合間を縫って、T 分間でできるだけ多くの飛行機を離陸させる

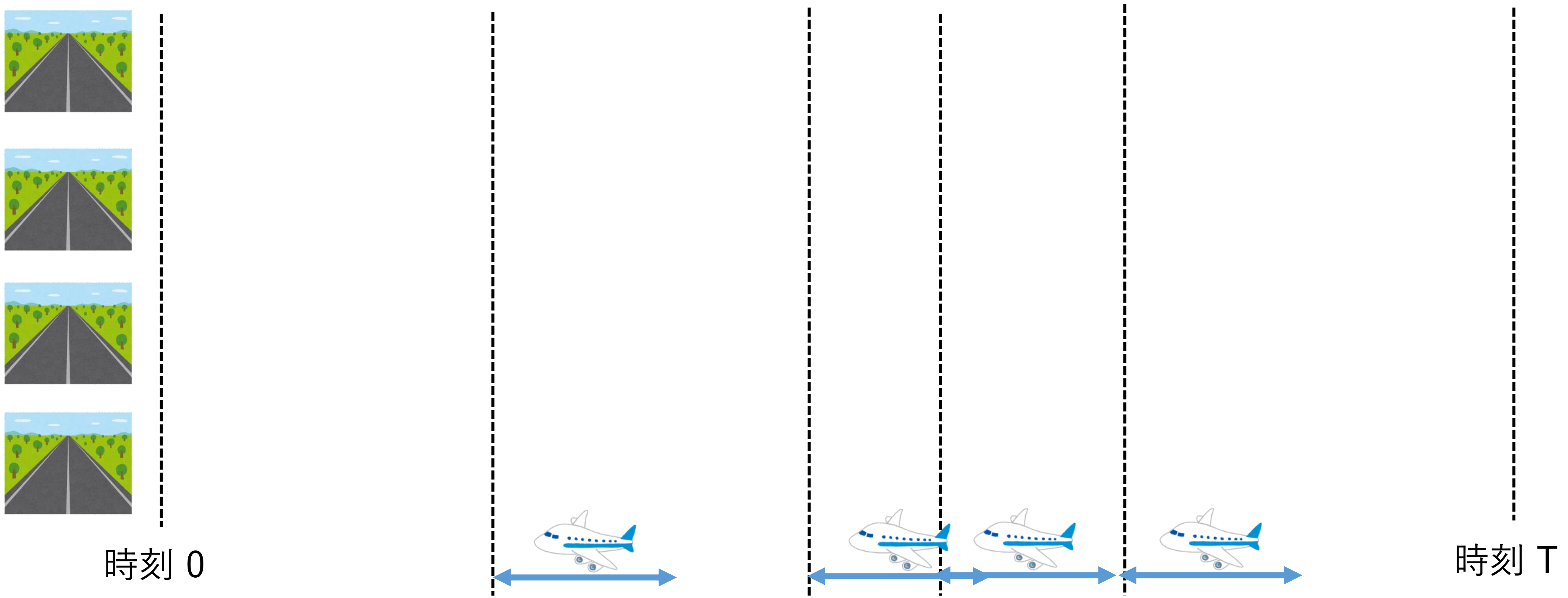
- ただし、1 回の離陸で K 分間、着陸で L 分間 1 つの滑走路を占有する

問題概要

- 空港に、N 本の滑走路がある

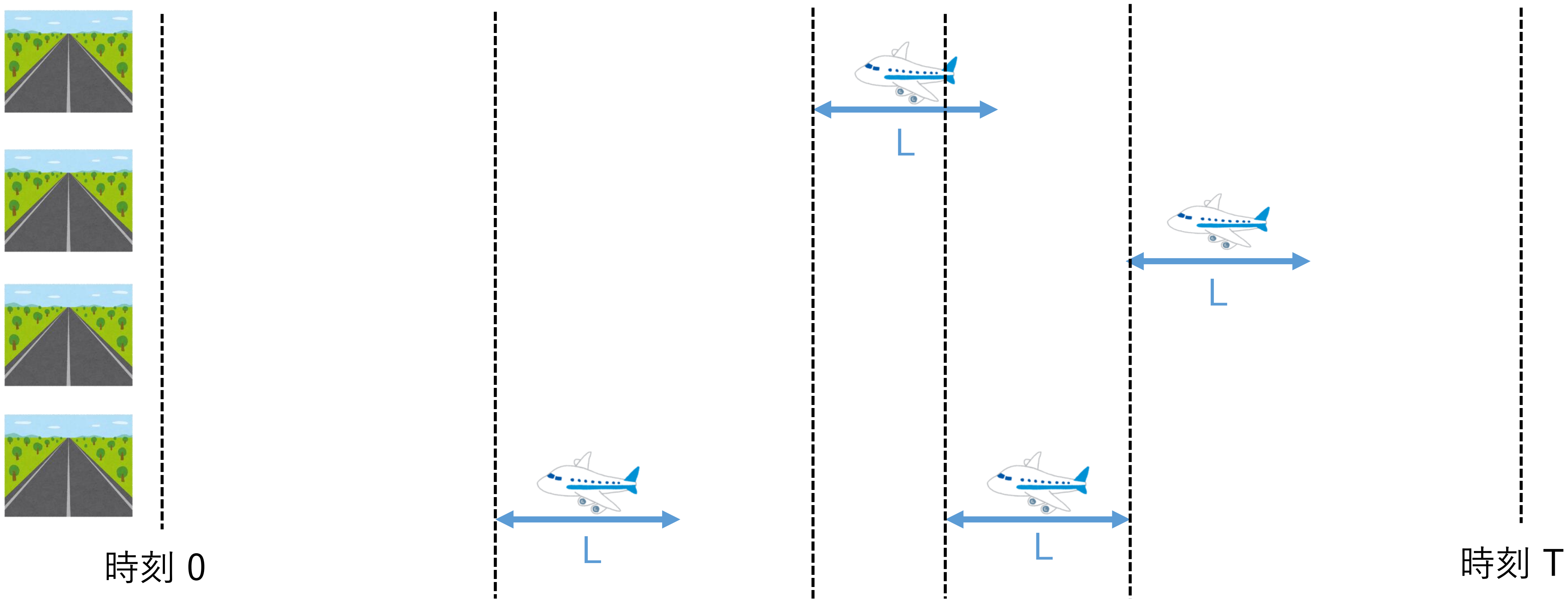
問題概要

- 空港に、N 本の滑走路がある
- M 個の着陸計画がある（使う滑走路は自由に決められる）



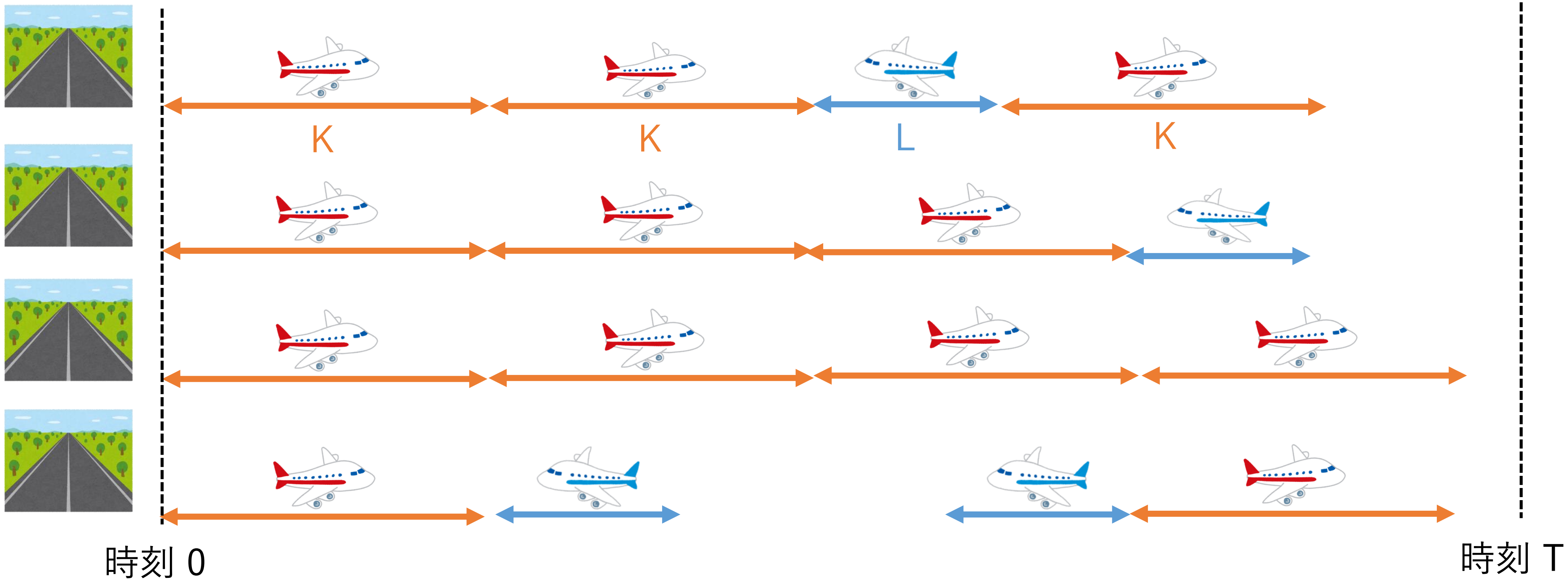
問題概要

- 空港に、N 本の滑走路がある
- M 個の着陸計画がある（**使う滑走路は自由に決められる**）



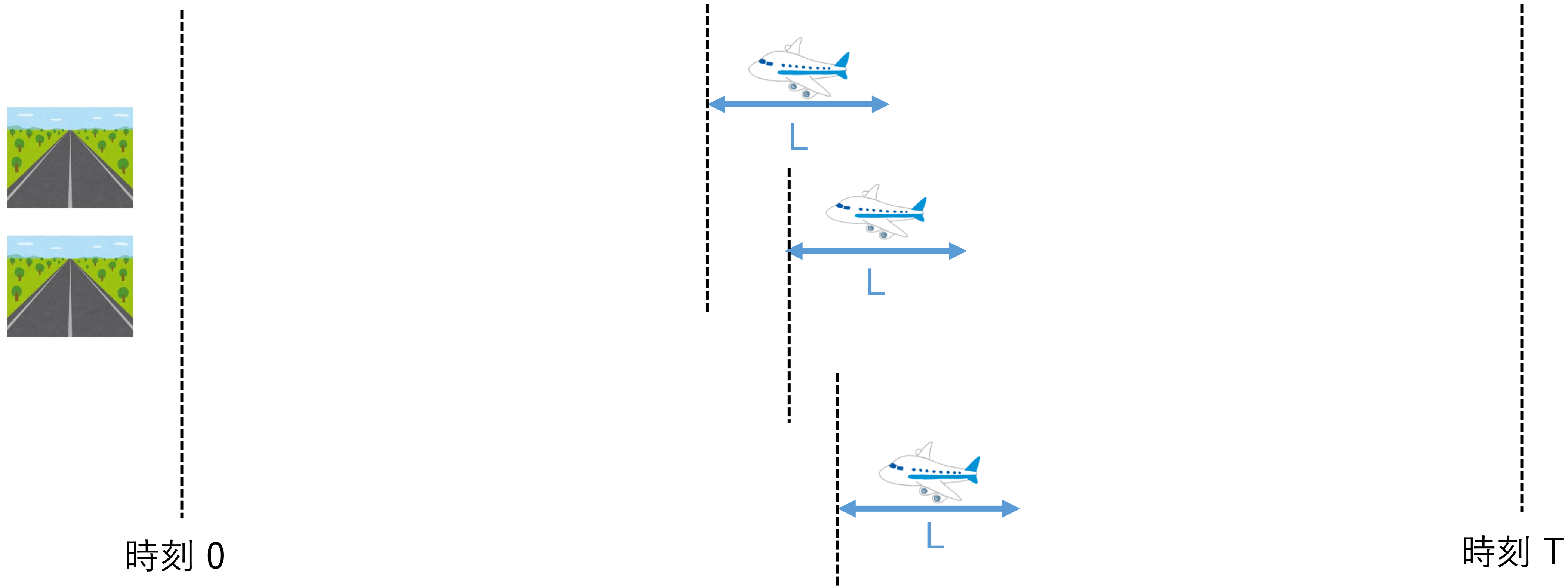
問題概要

- 空港に、N 本の滑走路がある
- M 個の着陸計画がある
- 合間を縫って、T 分間でできるだけ多くの飛行機を離陸させる



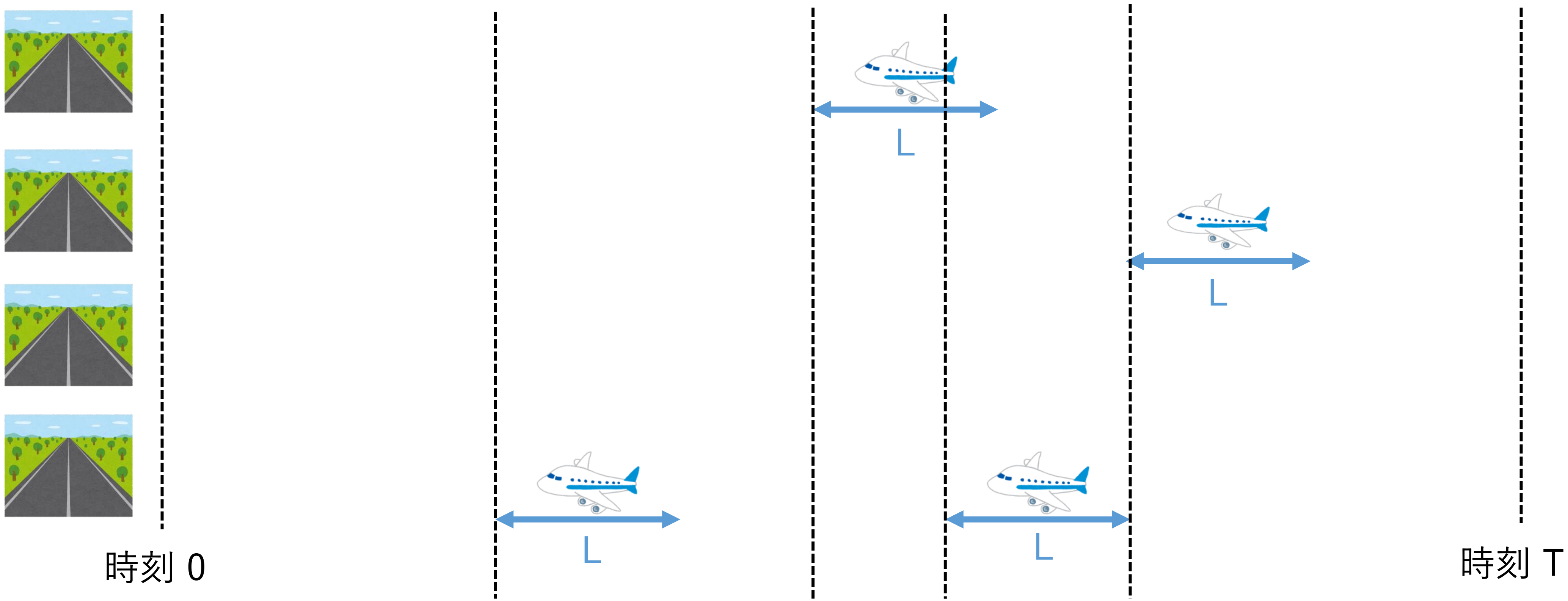
問題概要

- 下図のような場合は、着陸する時間がどうしても被るので、スケジュールを立てられない（-1 を出力）



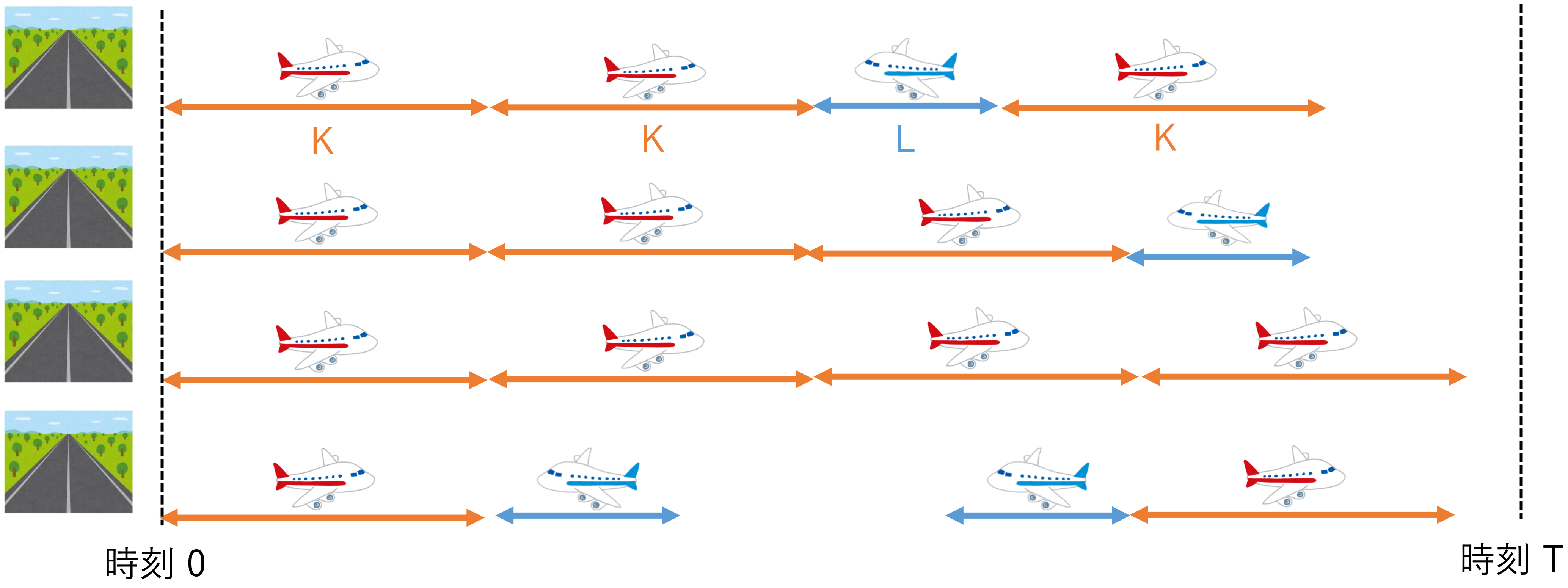
基本的な考え方

①まず、各着陸でどの滑走路を使うかを決める



基本的な考え方

①まず、各着陸でどの滑走路を使うかを決める
②そのもとで、最大どれだけ離陸させられるかを計算



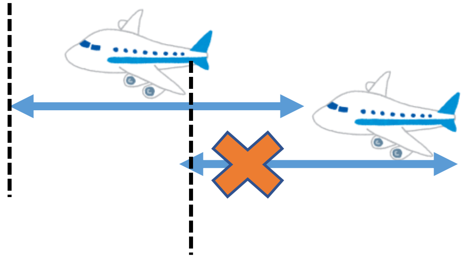
小課題 1

- $N = 1$
- 滑走路は 1 つなので、どの滑走路を使うか気にする必要はない
- とにかく離陸数を多くすることを考える

- A_i の番号の付けかたに意味はないので、まずソートして良い
- 以降、$A_1 \leq A_2 \leq \cdots \leq A_M$ とする

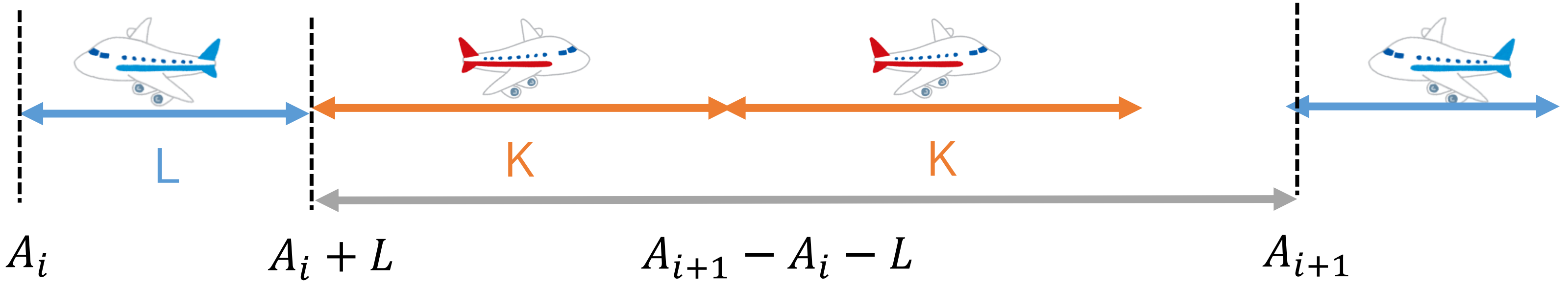
小課題 1：-1 になる条件

- $A_{i+1} - A_i < L$ なる i が存在すると NG
- 着陸している最中に次の飛行機が着陸を始めてしまうため



小課題 1

- -1 でないとき、着陸の合間に$A_{i+1} - A_i - L$ 分間の空き時間がある
- Q. この間にどれだけ離陸させられる？
- A. $\frac{A_{i+1} - A_i - L}{K}$ の小数点以下を切り捨てた数



小課題 1

- 答えは、すべての i についての $\frac{A_{i+1}-A_i-L}{K}$ (の切り捨て)の和
- A のソートがボトルネックで、計算量は $O(MlogM)$

- 7 点

小課題 2

- $N = 2, M \leq 20$
- 各飛行機について、どちらの滑走路を使うかを決める必要がある
- 着陸する飛行機だけでも決められないか？

小課題 2：考察

- M 機の飛行機について、滑走路の選び方は 2 通りあるので、全体で 2^M 通り
- 全ての選び方について、そのときの離陸数の最大値を求める
  - -1 になる場合（同じ滑走路で着陸が重なる）はちゃんと除く

- 小課題 1 と同じように、(空き時間)/ K を切り捨てた値の和を 2 つの滑走路について計算
- その合計が離陸数

小課題 2：計算量

- 滑走路の選び方を決めるのに $O\left(2^M\right)$
- 滑走路ごとに離陸数の最大値を求めるのに $O(NM)$
- 全体で $O\left(2^M NM\right)$ で、$M \leq 20$ なので間に合う $\left(2^{20} \approx 10^6\right)$

- 12 点

小課題 2：実装

- 2^M 通りの全探索は、2 進数を使った方法が有名
- 飛行機 i が滑走路 1 を使うなら i 桁目を 0 、滑走路 2 を使うなら i 桁目を 1 とした 2 進数で状態を表現する
- 10 進数で 0 から $2^M - 1$ まで探索すれば、全状態を探索できる

- いわゆる「bit 全探索」

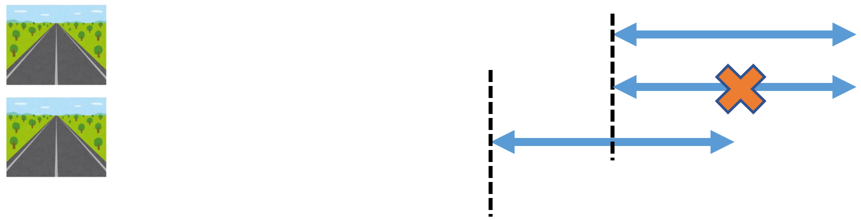
小課題 3

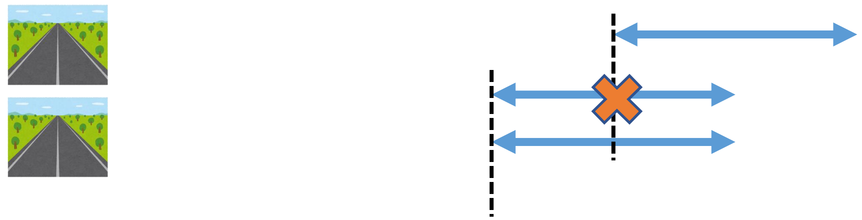
- $N = 2, K = 2, L = 2$
- 偶奇が重要な小課題？

小課題 3：-1 の場合

- 明らかに NG な場合：$A_i = A_{i+1} = A_{i+2}$

- これも NG：$A_i + 1 = A_{i+1} = A_{i+2}$ または $A_i + 1 = A_{i+1} + 1 = A_{i+2}$

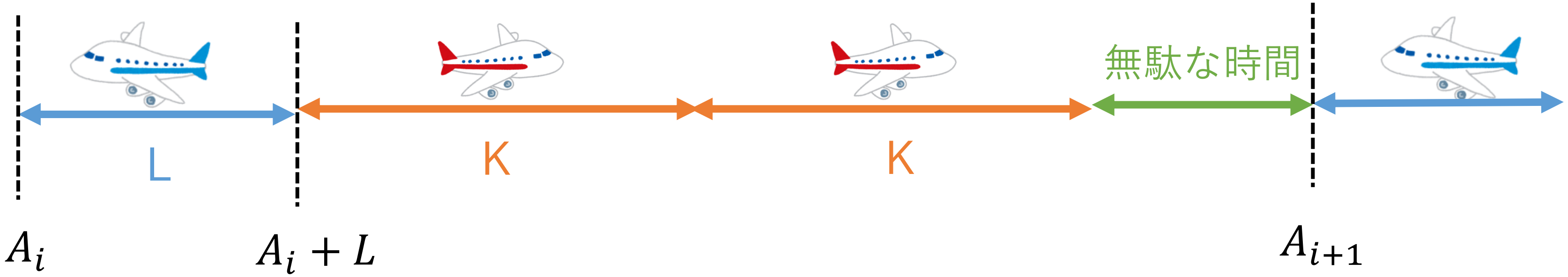




- つまり、長さ 2 の区間の中に A_i が 3 つ以上あるとNG
    - 逆に、そうじゃなかったら OK であることがいえる

考え方の転換

- (空き時間)$/K$ を切り捨てた数だけ離陸できるということは、切り捨てられた分の時間が無駄になっている
- 無駄な時間の和を最小化できれば、答えが最大化できる
- (無駄な時間)$=$ 空き時間を K で割った余り をできるだけ小さくすることを考えよう



小課題 3：A_i が相異なる場合

- 無駄な時間：空き時間が奇数のとき 1、偶数のとき 0 になる
  →できるだけ空き時間を偶数にしたい

- A_i が偶数のものを滑走路 1 に割り当てる
- A_i が奇数のものを滑走路 2 に割り当てる

- すると、同じ滑走路内での空き時間（端を除く）は必ず偶数！
  - $A_i + L$は $L = 2$ より A_i と偶奇が同じで、かつ (偶数) − (偶数) = (偶数)、(奇数) − (奇数) = (偶数) だから

小課題３：A_i が相異なる場合

- 空き時間が偶数、かつ $K = 2$ より、空き時間に離陸を詰め込める

  →無駄な時間が端以外に生まれず、これが最適になる

  - 無駄な時間をこれ以下にすることができないことが示せる

- A_i に被りがある場合も、この戦略を応用すれば OK

小課題 3

- A_iが被っている場所は他にやりようがないので、滑走路を両方使う
- その他の場所は、A_i の偶奇で滑走路を分ける

- 実は、これが離陸数を最大にする戦略
  - 滑走路を同時刻に両方使うと滑走路 1 と滑走路 2 が等価になり、時刻0と同じ状況になる
- この通りに滑走路を割り振って計算すれば、$O(M)$ で答えが求まる

- 17 点

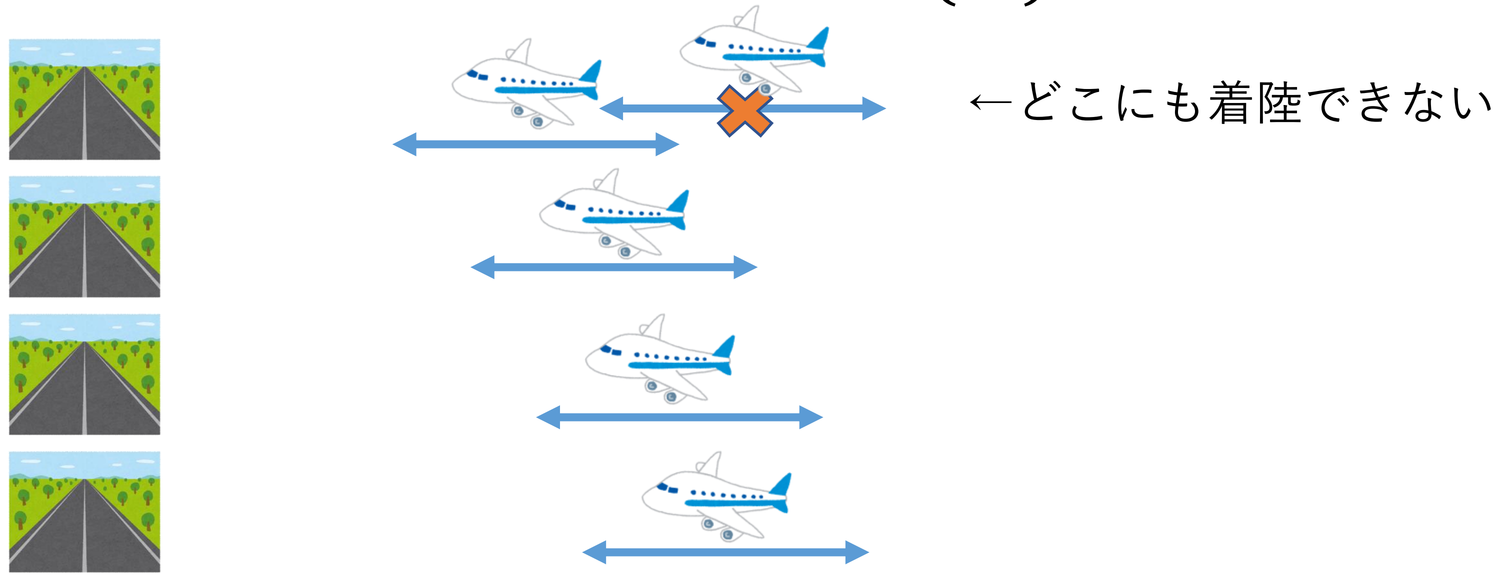
小課題 4

- $N = 2$
- 今までの小課題で分かったこと
  - 着陸する滑走路が決まると、空き時間を K で割って切り捨てたものの和が答えになる→空き時間を K で割った余りが無駄
  - $K = 2, L = 2$ なら 長さ 2 の区間に 3 つ以上 A_i があると NG
  - $K = 2, L = 2$ なら A_i の偶奇を滑走路内でできるだけそろえると良い

- これを総動員しよう

小課題 4：-1 になる条件

- $N = 2, K = 2, L = 2$ のとき、長さ 2 の区間の中に A_i が 3 つ以上あるとNG だった
- 一般の場合では、長さ L の区間の中に A_i が $N + 1$ 個以上あると NG という条件になり、$O(M)$ で判定できる



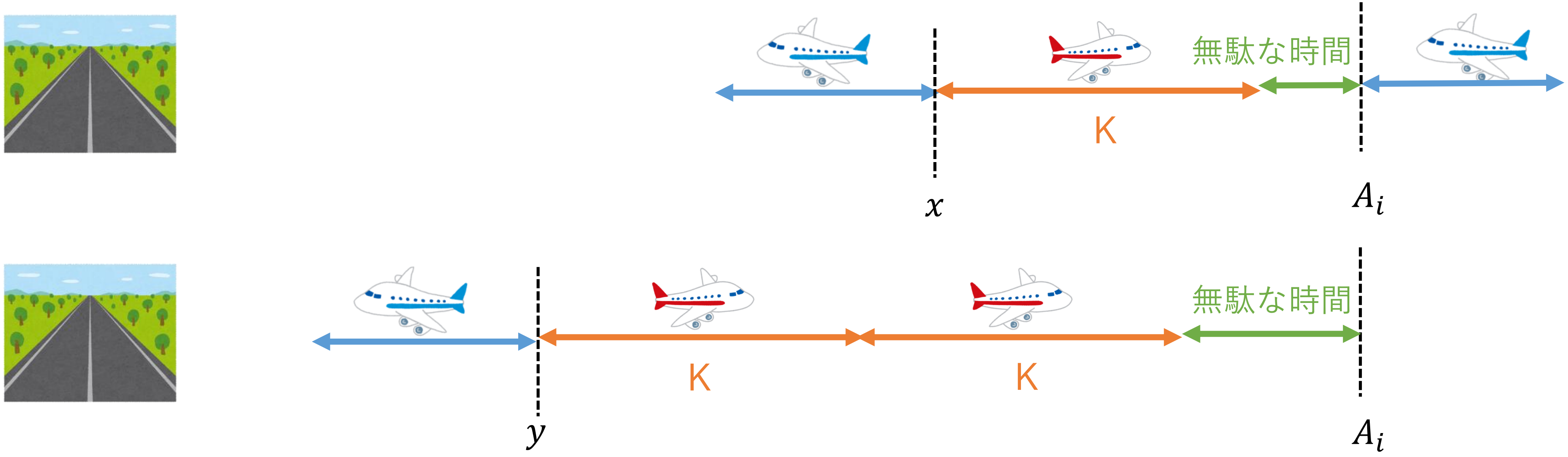
小課題 4

- 空き時間を K で割った余りの和をできるだけ小さくしたい
  - 小課題 3 では、同じ滑走路ではなるべくA_i の偶奇を同じにすればよかった
- 滑走路 1, 2 で最後に着陸を終えた時間を x, y とする

- 次の飛行機が時刻 A_i に着陸するとき、生じる無駄な時間は？
  - 滑走路 1 に着陸するとき $(A_i - x + K)\%K$
  - 滑走路 2 に着陸するとき $(A_i - y + K)\%K$

小課題 4

- $(A_i - x + K)\%K, (A_i - y + K)\%K$ を計算することで、どちらに着陸すれば次に生じる無駄な時間が小さくなるかわかる
- つまり、毎回「生じる無駄な時間が最も小さい滑走路を選ぶ」という貪欲法ができる



小課題 4

- 普通、貪欲法は最終的に最適になるとは限らない（注意！）
  - そういう時は動的計画法が使われる

- ただし、**この問題では**貪欲法で最適解を出せる
- 各滑走路で最後に着陸を終えた時間を持てば、$O(NM)$ で実装できる

- 12 + 17 + 28 = 57 点
- 貪欲法は正当性を確かめるのが大事（今回はちょっと難しい）

貪欲法の正当性

- 証明の概要
  - 滑走路 1,2 で最後に着陸を終えた時間を x, y とする
  - 滑走路 1 を選ぶと、最後に着陸を終えた時間は (滑走路 1,滑走路 2)= $(A_i + L, y)$となる
  - 滑走路 2 を選ぶと、最後に着陸を終えた時間は(滑走路 1,滑走路 2)= $(x, A_i + L)$ となるが、これは $(A_i + L, x)$と等価
  - 貪欲法だと滑走路 1 を選ぶことになるとする($(A_i - x + K)\%K \leq (A_i - y + K)\%K$)
  - このもとで、任意の z について、

$(A_i - x + K)\%K + (z - y + K)\%K \leq (A_i - y + K)\%K + (z - x + K)\%K$ がいえる

  - よって、**滑走路 2 を選んでも得になることはない**ので、滑走路 1 を選ぶのが最適

- 一般の N の場合も同様に示せるので、以降の小課題でも貪欲法が使える

小課題 4 別解

- $N = 2$ ならば、$dp[i][j]$: 滑走路 1,2 で最後に着陸したのが飛行機 i, j のときの離陸数の最大値 という DP ができる
- これは $O\left(M^2\right)$ かかるが、Segment Tree で $O(MlogN)$ に高速化でき、小課題 4 が通る

- ただ、この解法は $N = 2$ 限定なので、満点には繋がらない…
  - この解法自体けっこう難しい

小課題 5

- $N \leq 100$
- 小課題 4 を、一般の N に応用するだけ
- 各滑走路について、「最後に着陸で使ったときの終了時刻を K で割った余り」を保存した配列をつくる

- 最も無駄な時間が少なくなる滑走路を $O(N)$ で選ぶ
  - ただし、着陸時間が被るものを除くようにする
- $O(NM)$ で解けた
- 75 点

満点解法

- $N \leq 10^5$
- 小課題 5 の解法を高速化したい

- 「最も無駄な時間が少なくなる滑走路」を高速に求めたい

→二分探索！

- ただし、二分探索はソートされていないとできないので、工夫する必要がある
  - 終了時刻 % K の配列は中身の大きさ順に並んでないことに注意

満点解法

- multiset という、集合を管理するデータ構造がある
  - set の重複を許したバージョン
  - 挿入、削除、検索、二分探索などが $O(logN)$ でできる
  - C++ の標準ライブラリにある
- multiset で終了時刻 % K を管理する
- 各 A_i について、終了時刻 % K が A_i% K 以下のうち最大の滑走路を二分探索で求めれば、使う滑走路が分かる
- $O(MlogN)$ で解けた
- 100 点

満点解法

- 着陸時間が被るものを multiset に入れないように注意
- 被らなくなるまで queue に入れておいたりして取り除けば OK

得点分布