解説

手持ち花火
sparklers

問題概要

- N人で一直線に並んで花火をする
- 花火はT秒だけ燃え続ける
- はじめJOI君の花火だけ着火する
- うまく火を移しあって全員の花火を着火するには、どれくらい速く動けばいいか？
- 速度制限の最小値を求めよ

- N≦100,000

問題概要

- N人で一直線に並んで花火をする
- 花火はT秒だけ燃え続ける
- はじめJOI君の花火だけ着火する
- うまく火を移しあって全員の花火を着火するには、どれくらい速く動けばいいか？
- **速度制限の最小値を求めよ**

- N≦100,000

考察0

- ある速度制限sで全員に火を移せるならば、それよりsが大きい速度制限でも全員に火を移せる
- 全員に火を移せるsの中で最小値を求める
- このタイプの最適化問題で使いやすいアルゴリズムがある

考察0

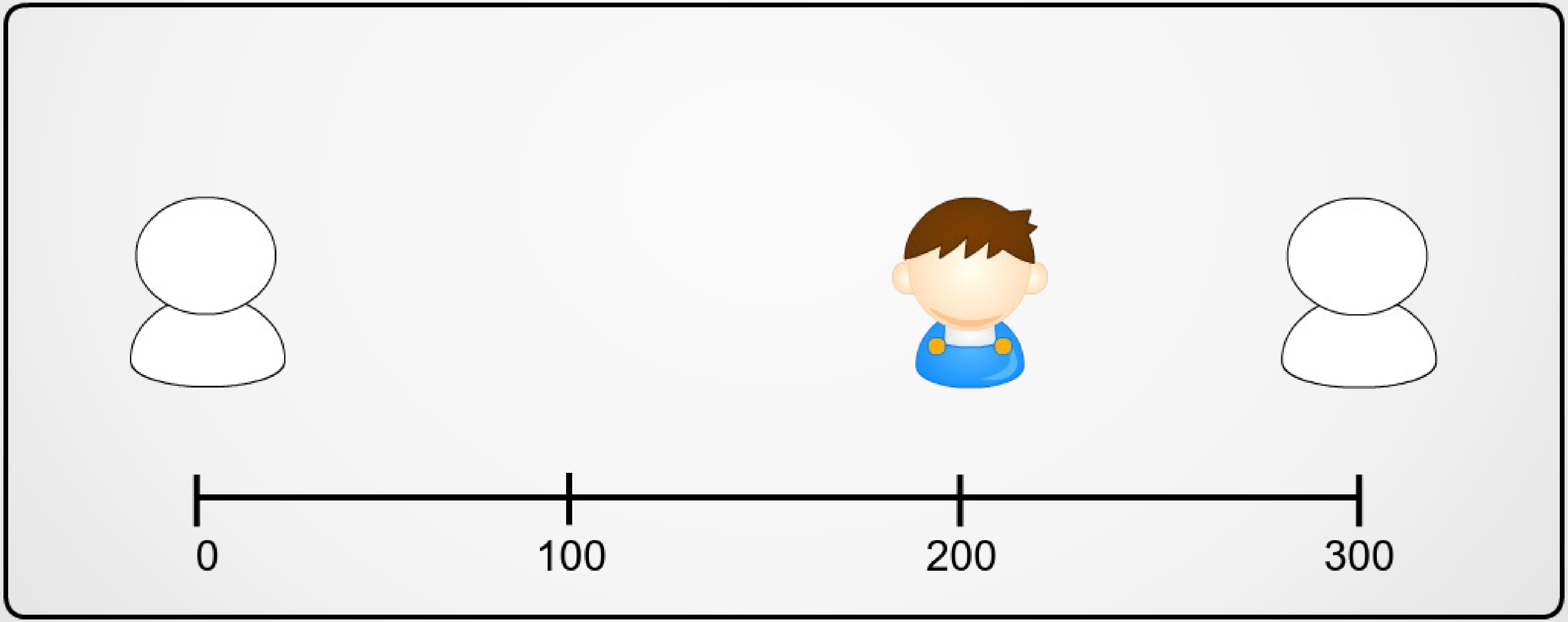
- ある速度制限sで全員に火を移せるならば、それよりsが大きい速度制限でも全員に火を移せる
- 全員に火を移せるsの中で最小値を求める
- このタイプの最適化問題で使いやすいアルゴリズムがある

二分法

入力例

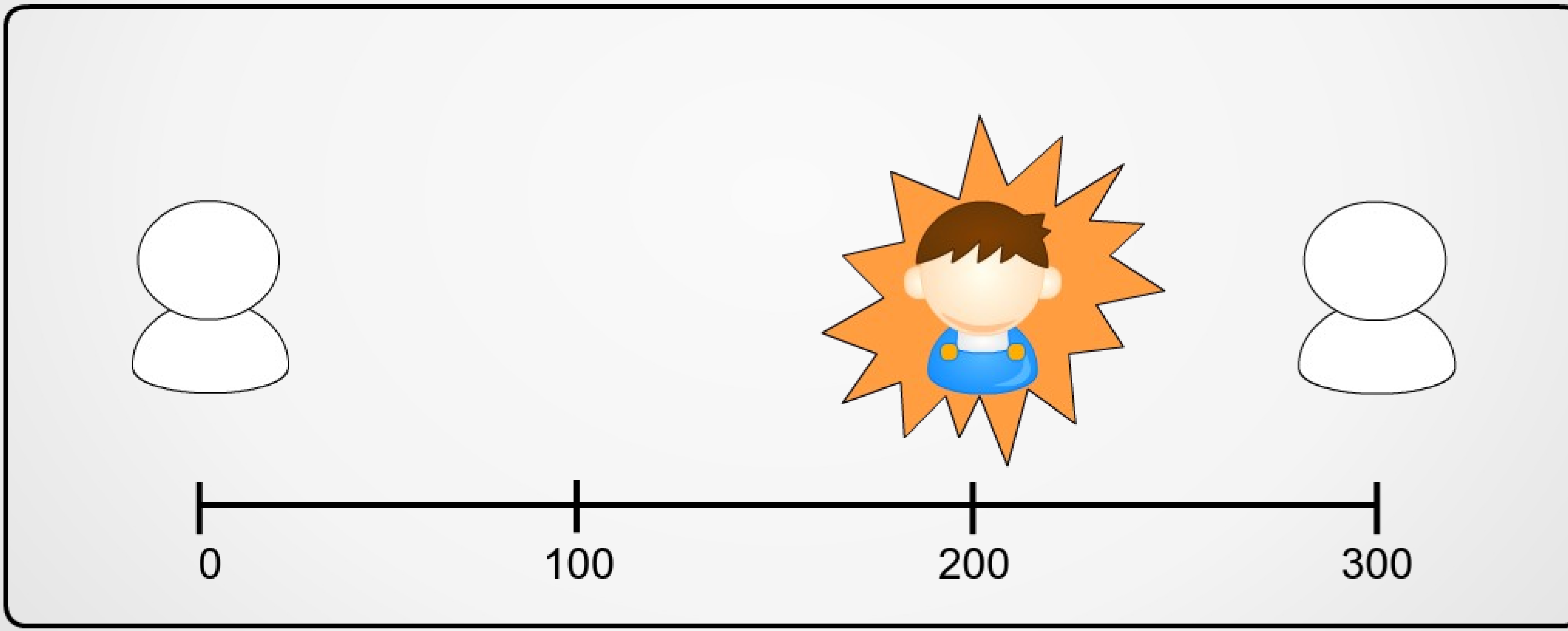
入力例1

- N=3, K=2, T=50



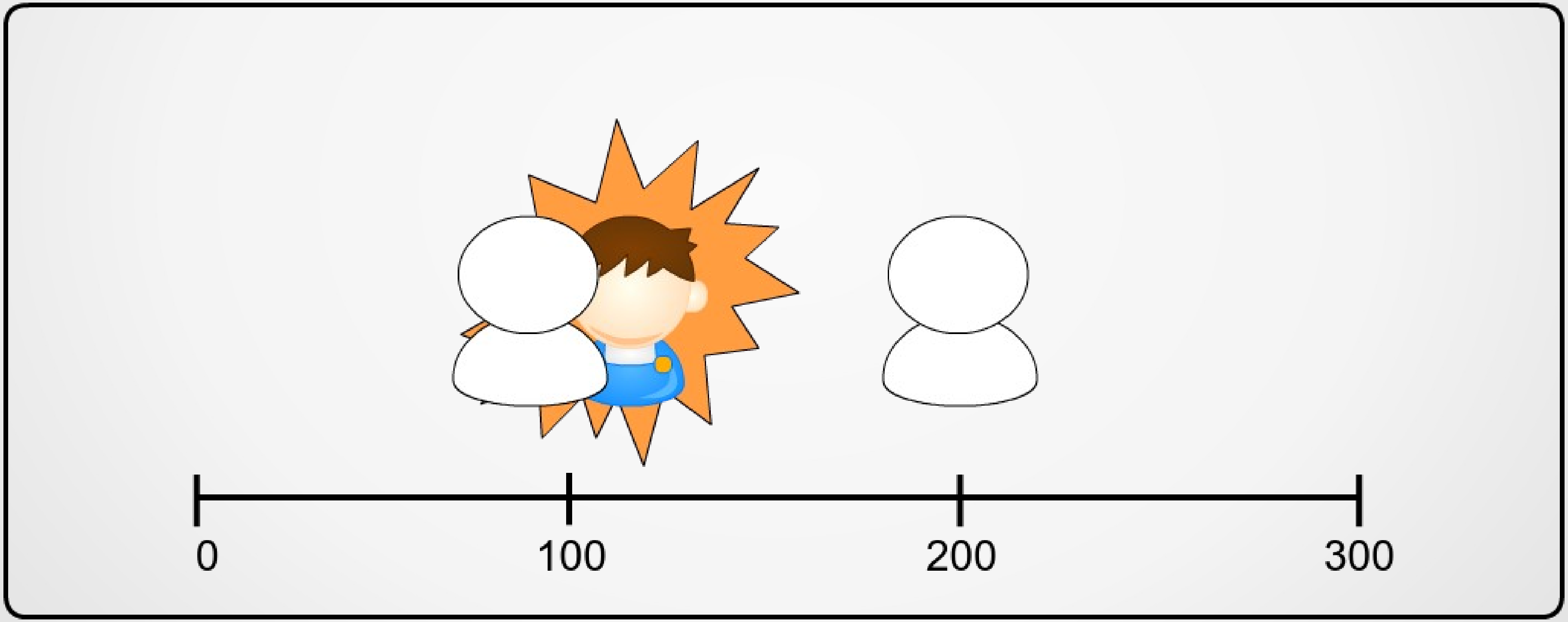
入力例1 (s=2)

- 時刻 0 (JOI君着火)



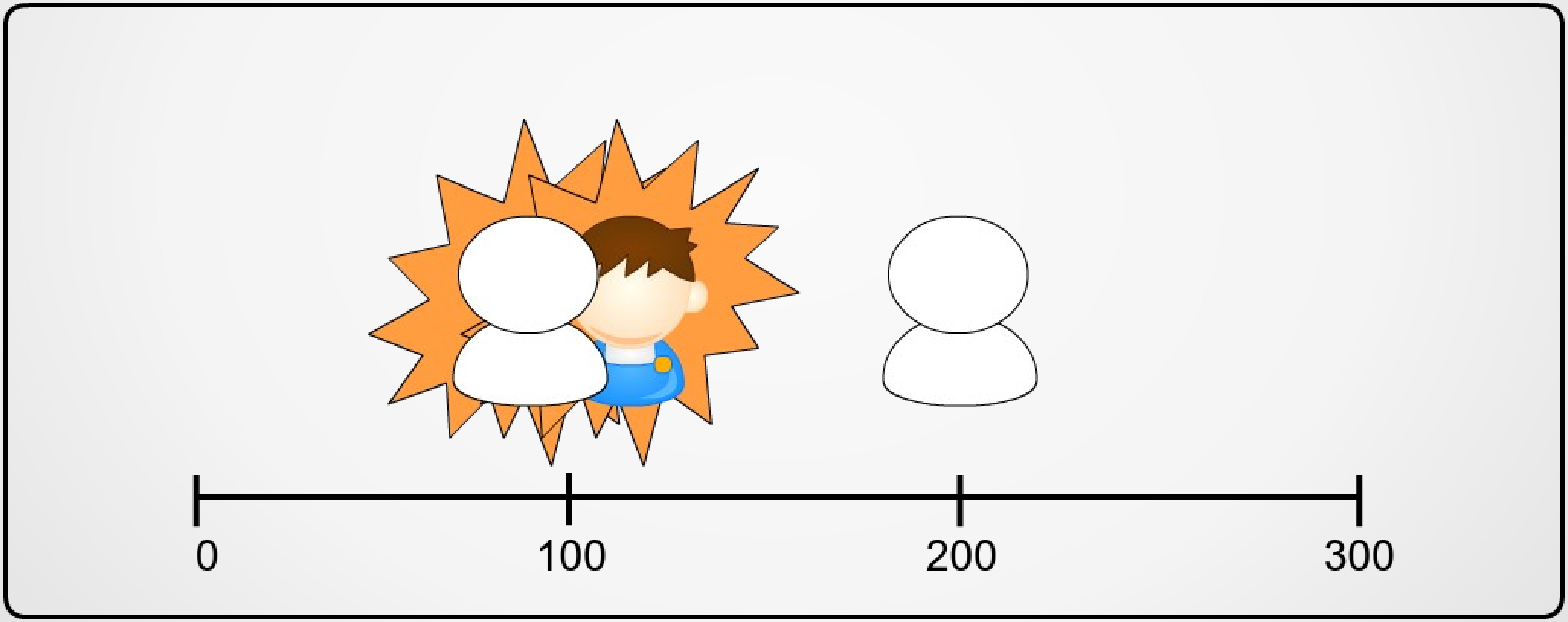
入力例1 (s=2)

- 時刻 50



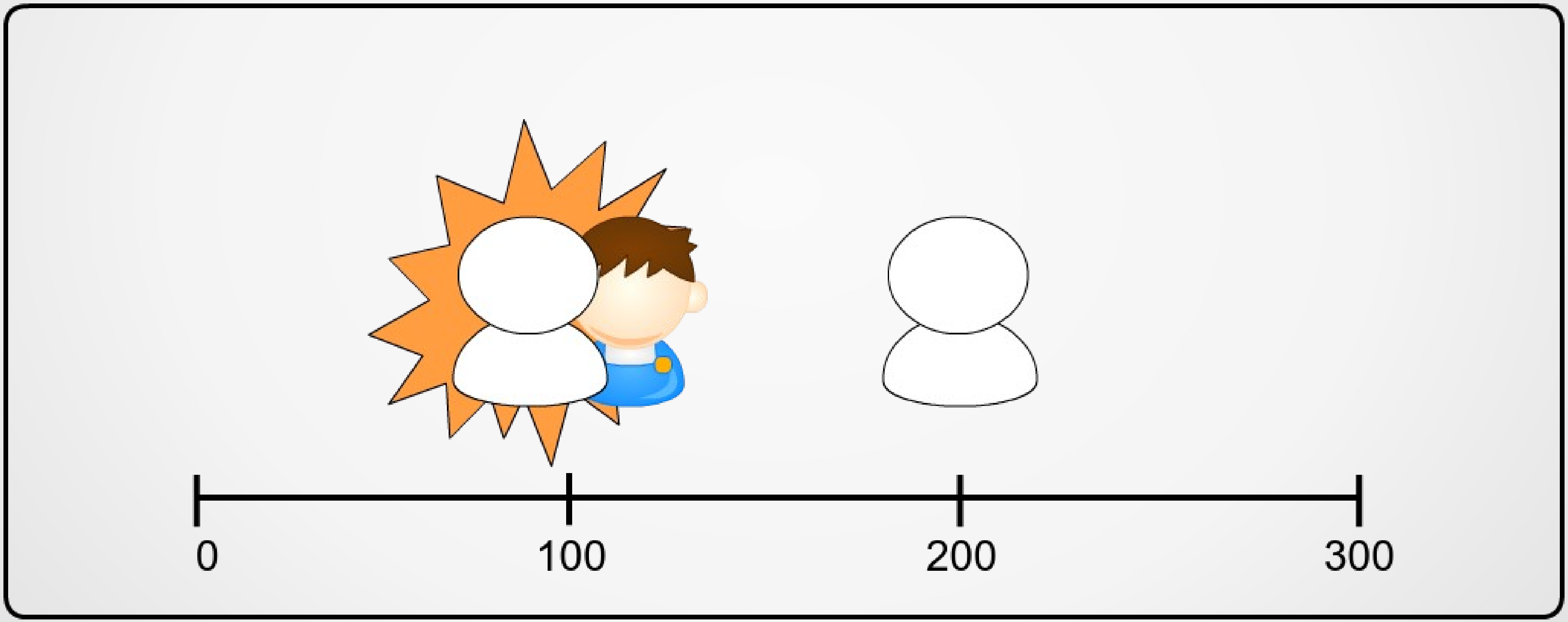
入力例1 (s=2)

- 時刻 50 (火を移す)



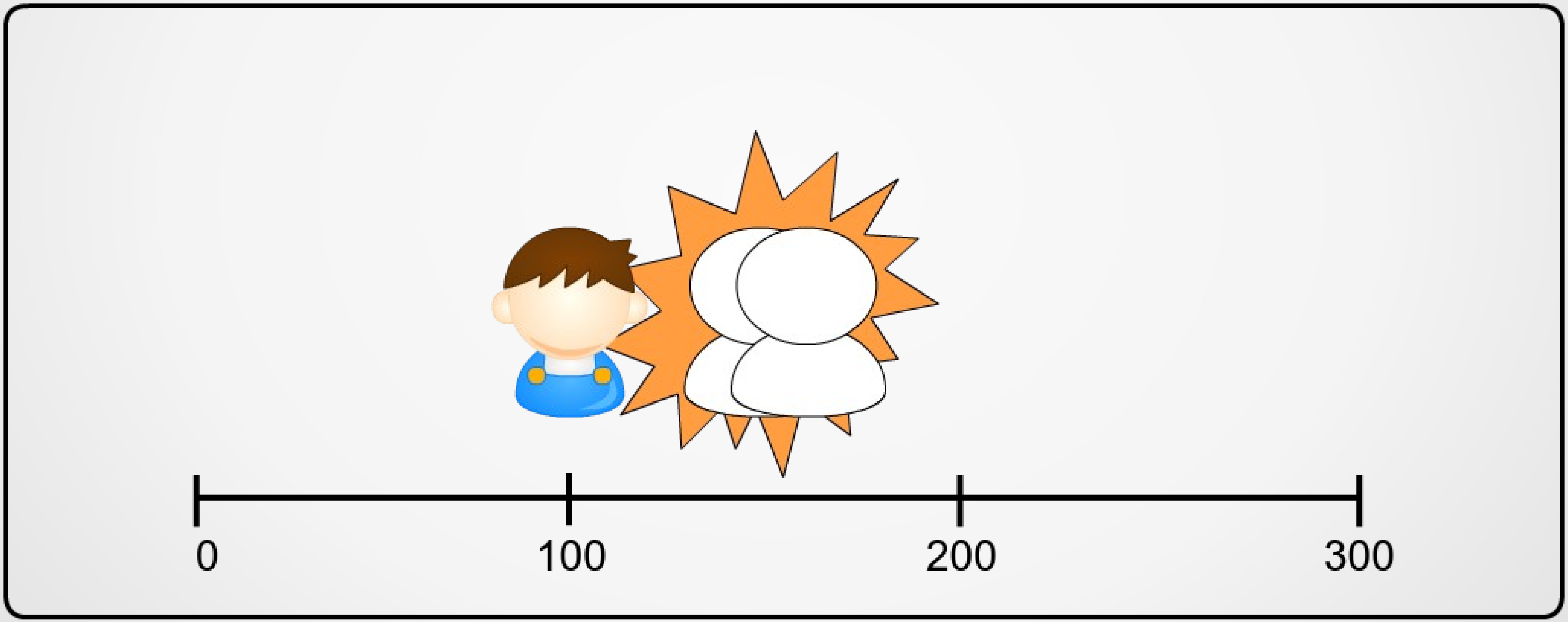
入力例1 (s=2)

- 時刻 50 (JOI君燃え尽きる)



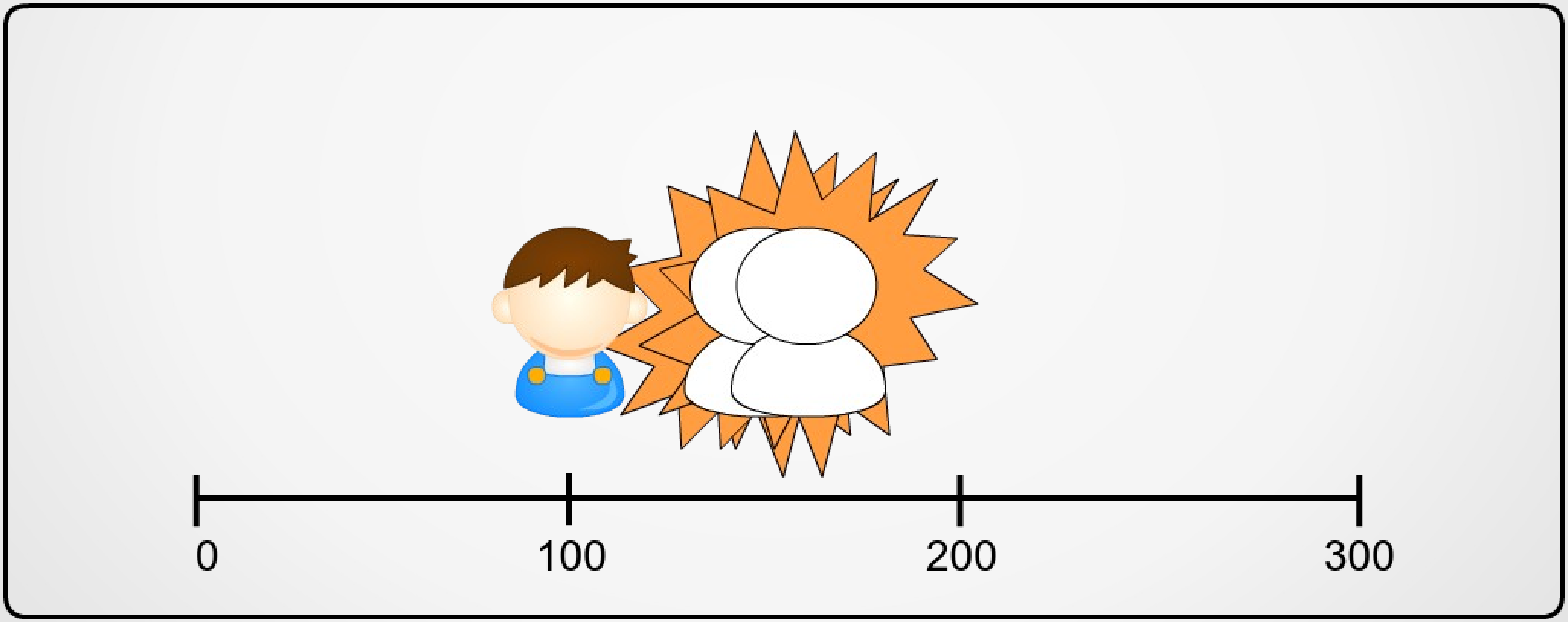
入力例1 (s=2)

- 時刻 75



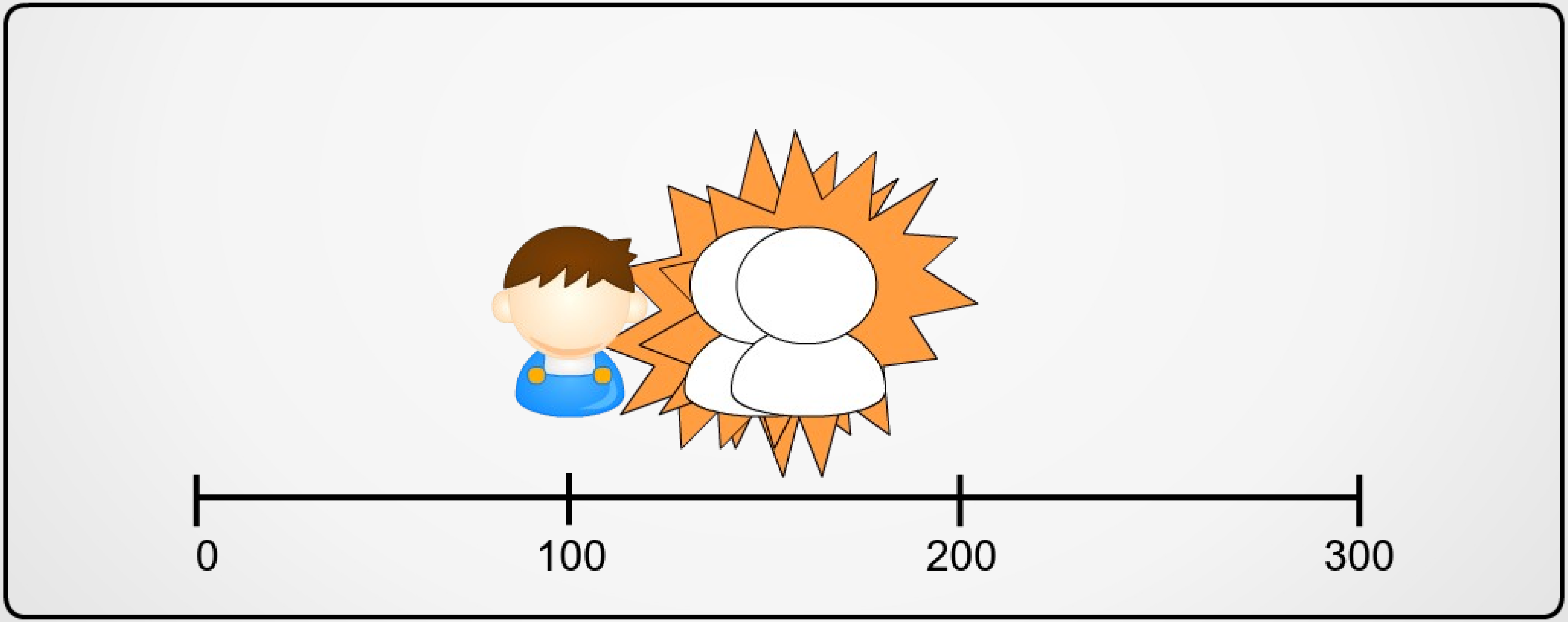
入力例1 (s=2)

- 時刻 75 (火を移す)



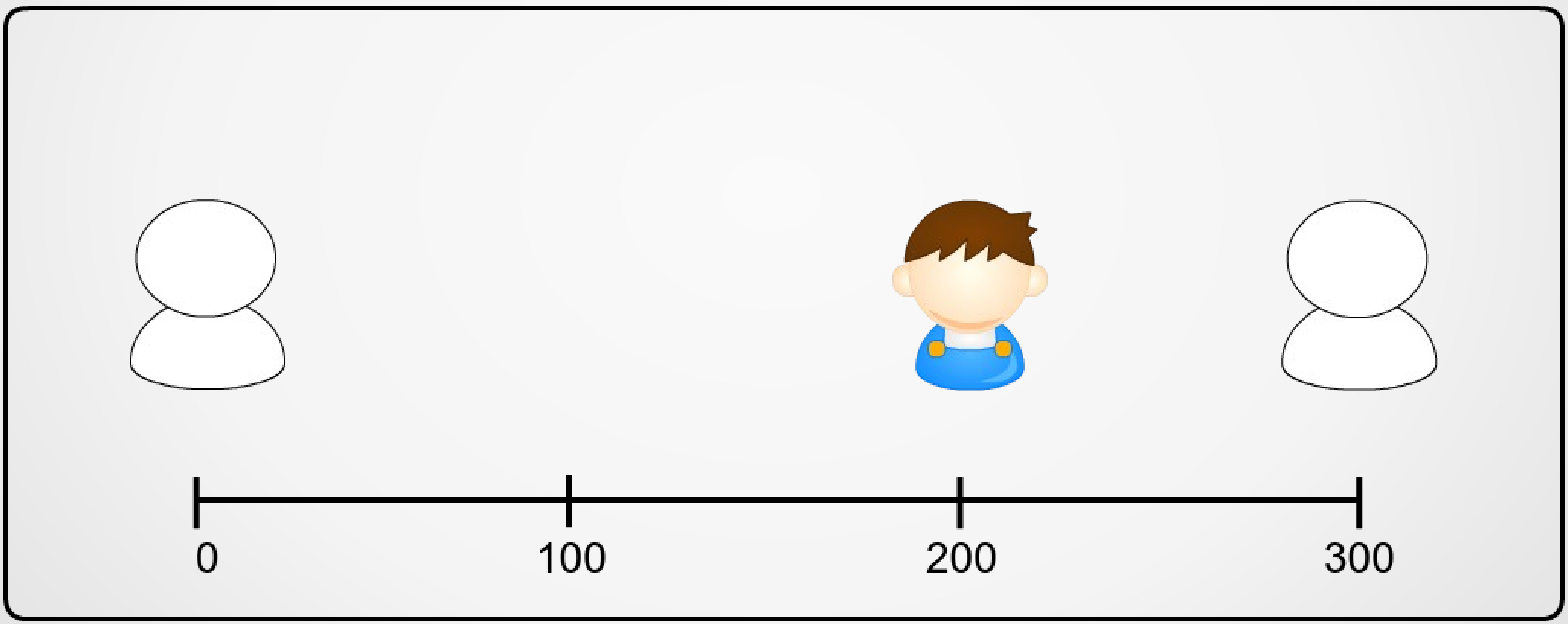
入力例1 (s=2)

- 時刻 75 (全員に火が移った!)



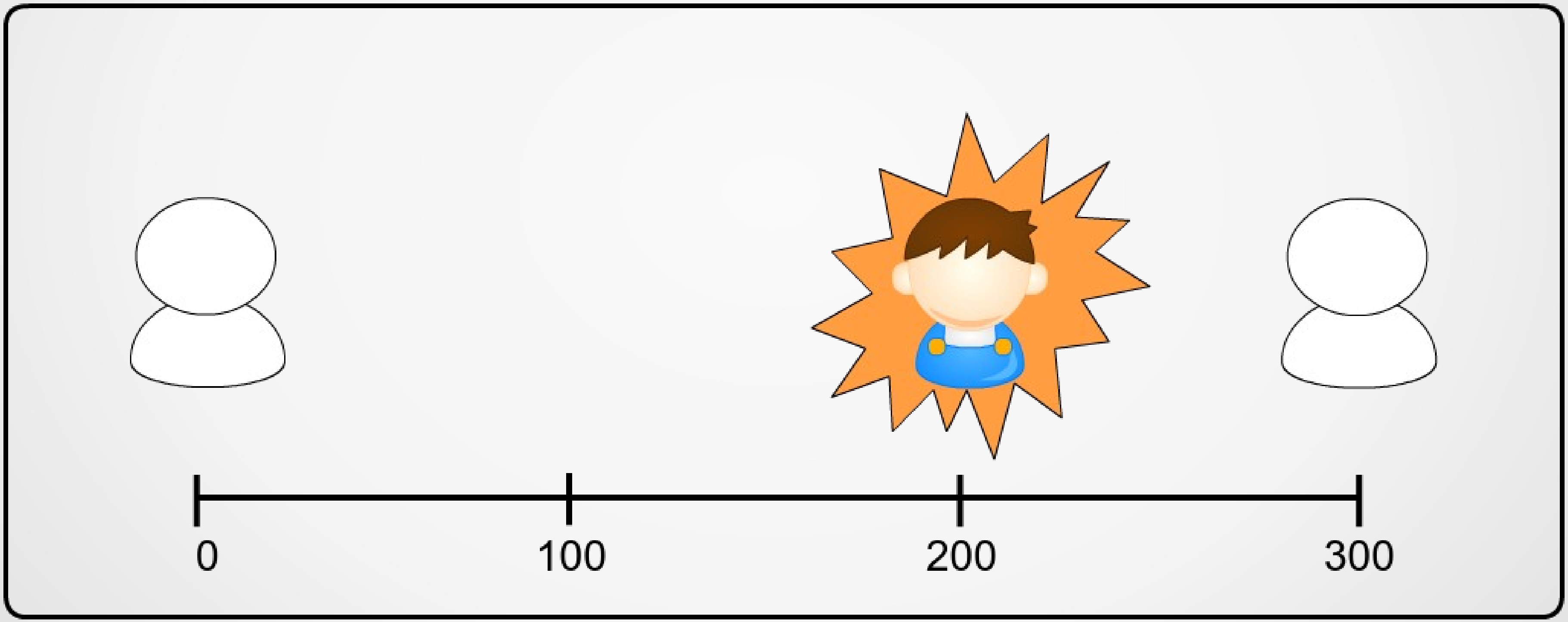
入力例2

- N=3, K=2, T=10



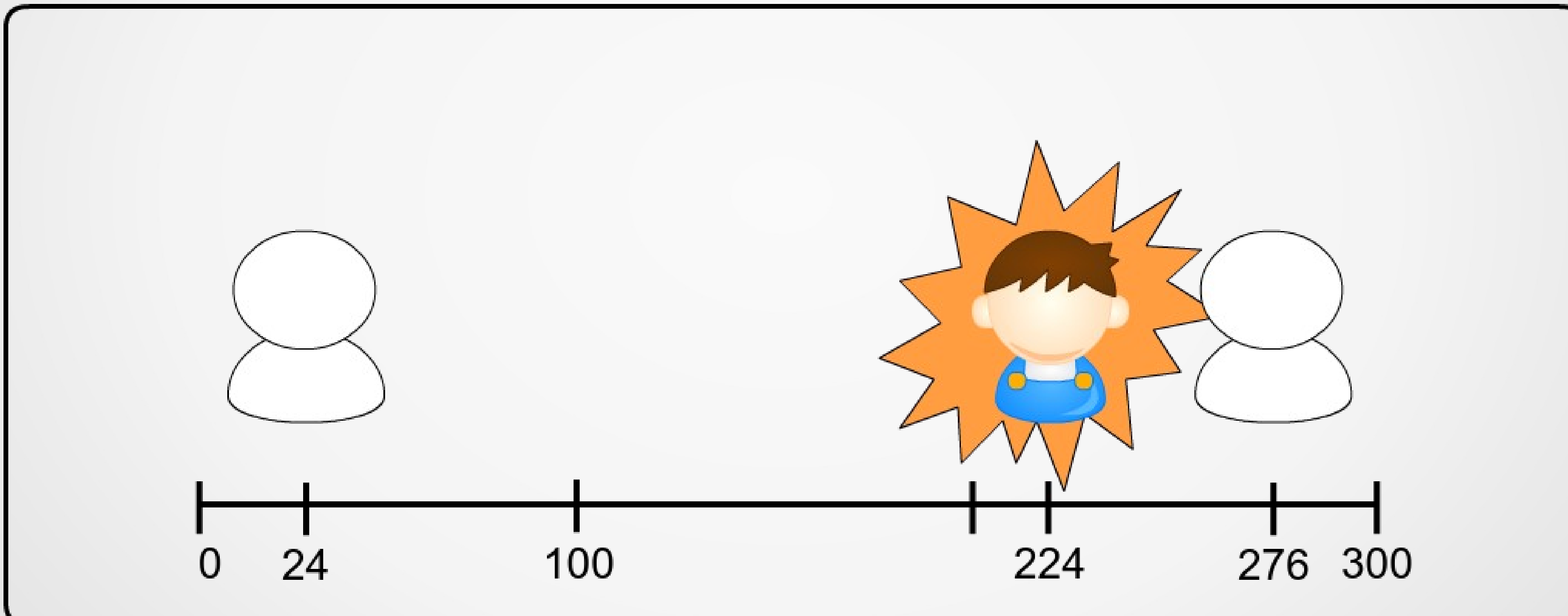
入力例2 (s=8)

- 時刻 0(JOI君着火)



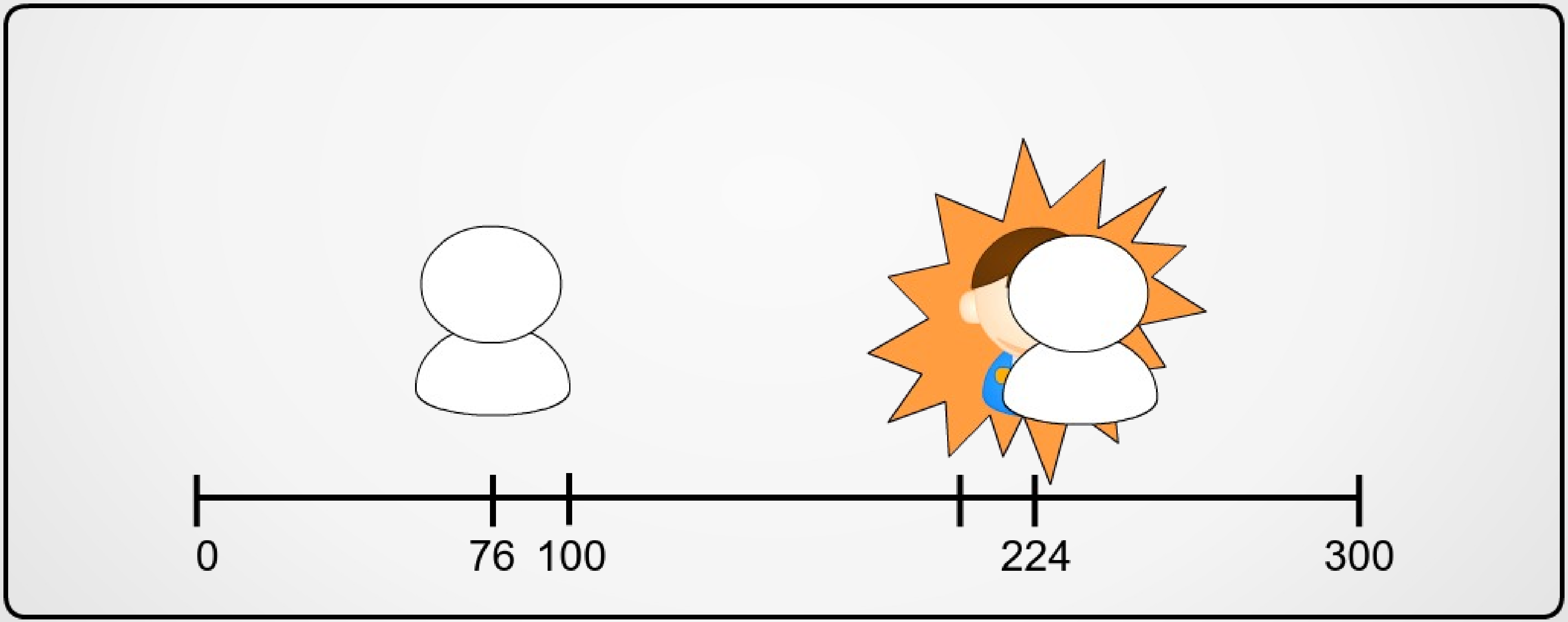
入力例2 (s=8)

- 時刻 3(JOI君停止)



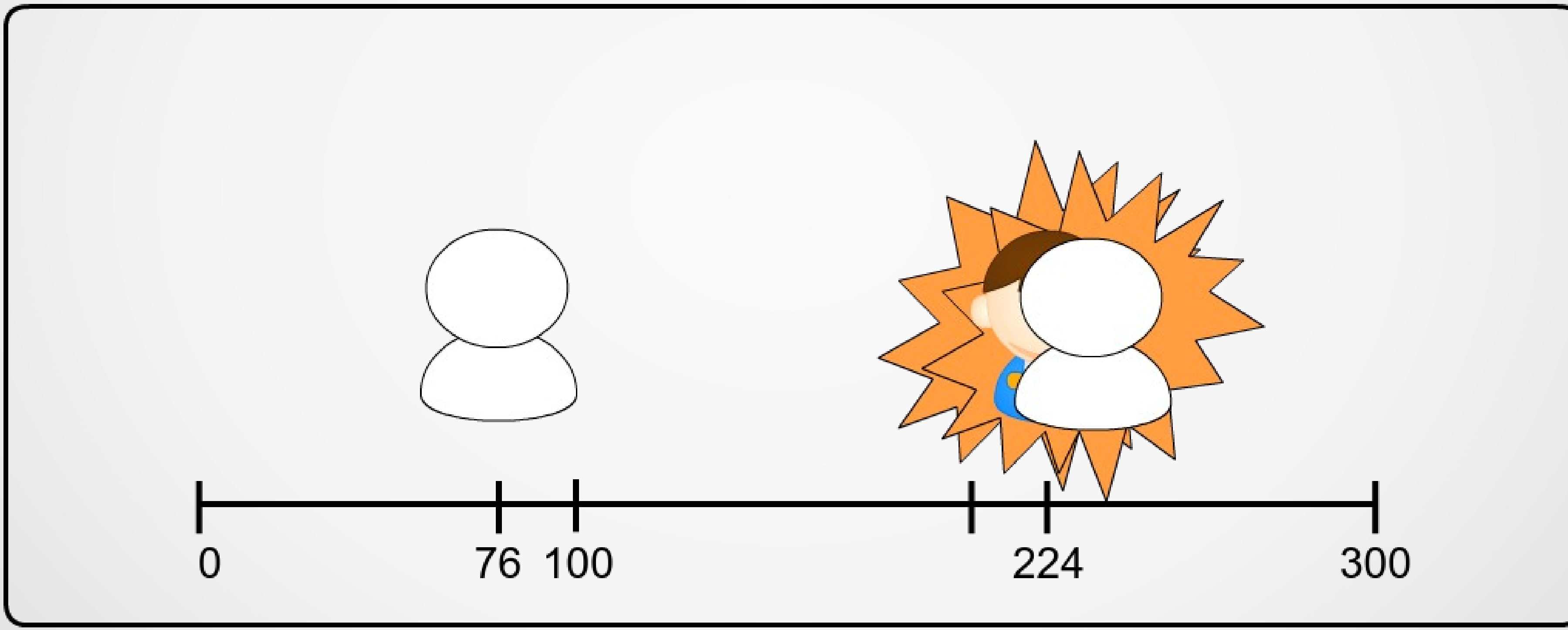
入力例2 (s=8)

- 時刻 9.5



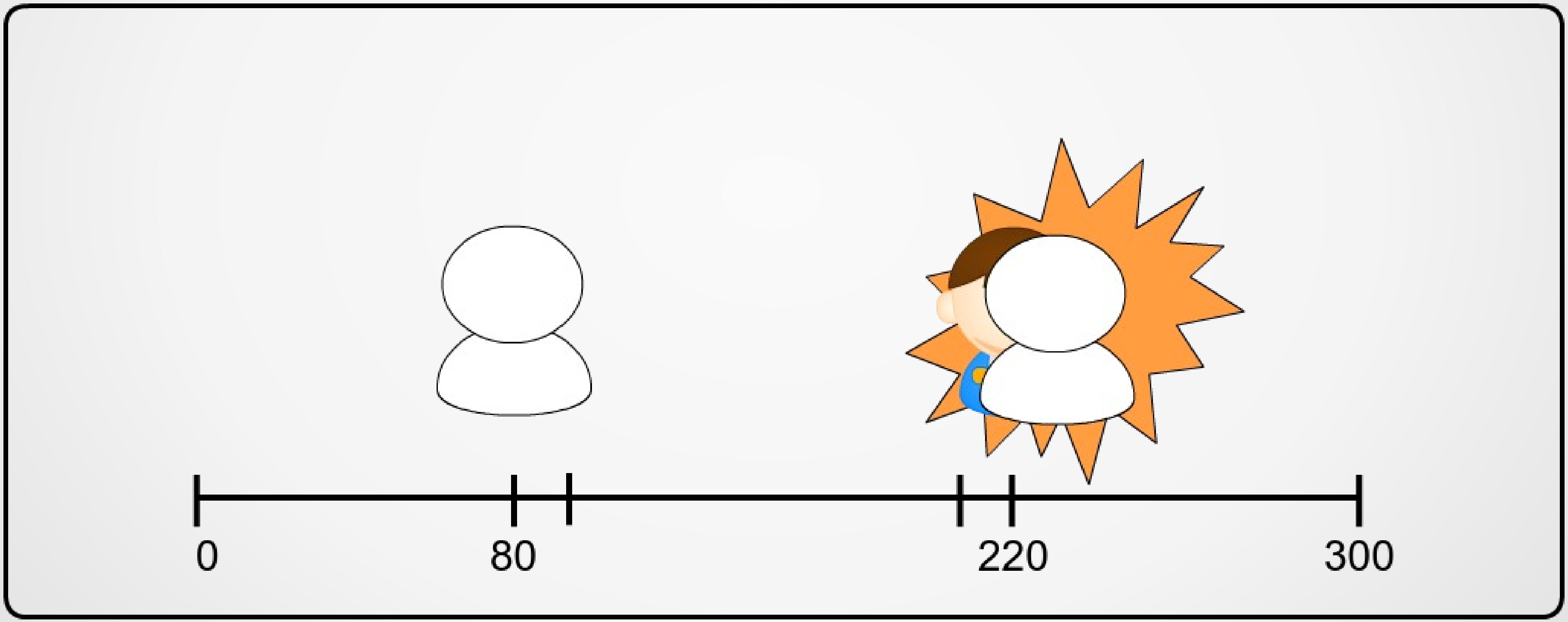
入力例2 (s=8)

- 時刻 9.5(火を移す、JOI君移動再開)



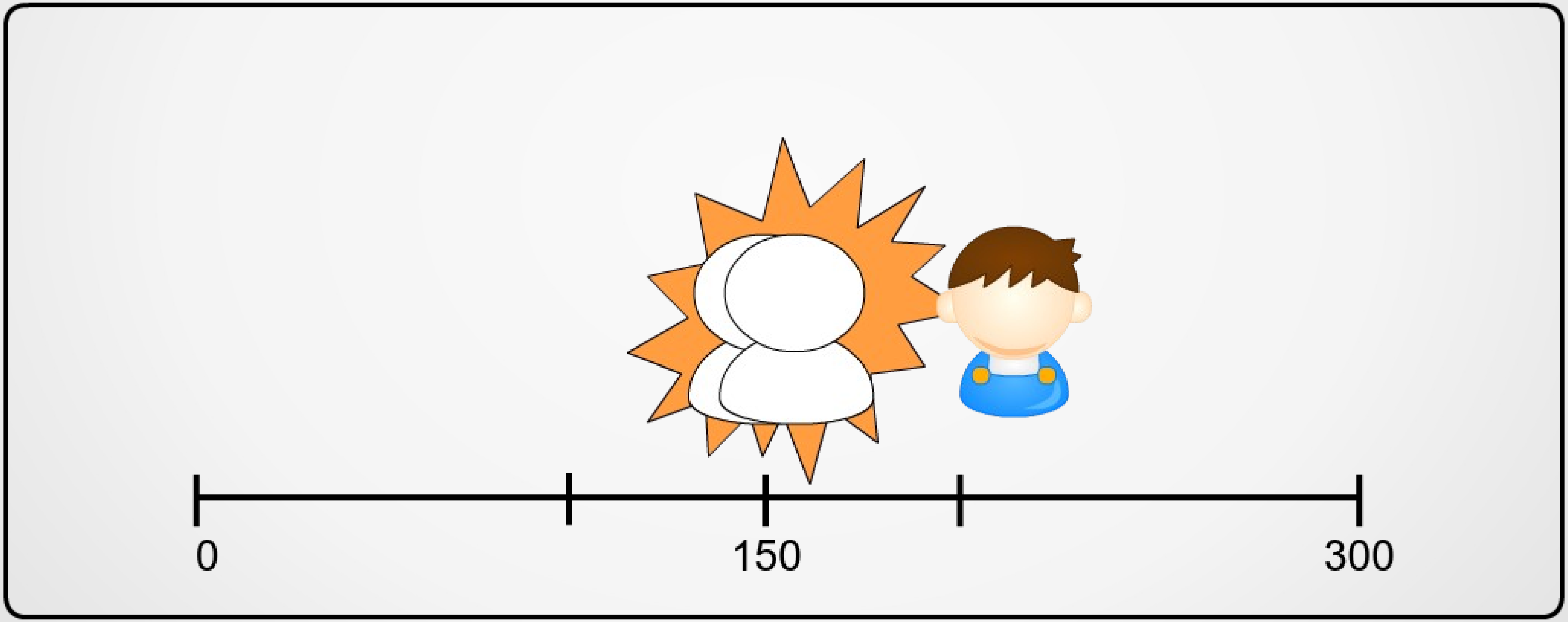
入力例2 (s=8)

- 時刻 10(JOI君燃え尽きる)



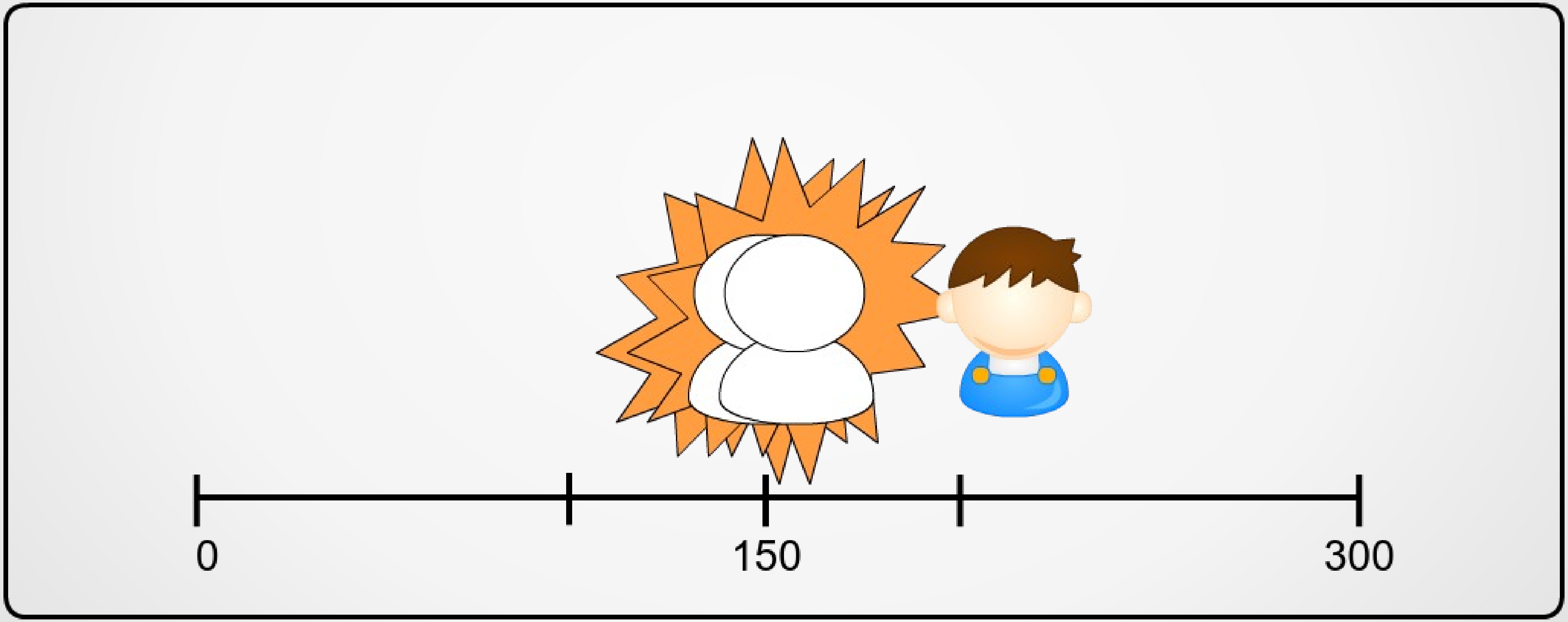
入力例2 (s=8)

- 時刻 18.75



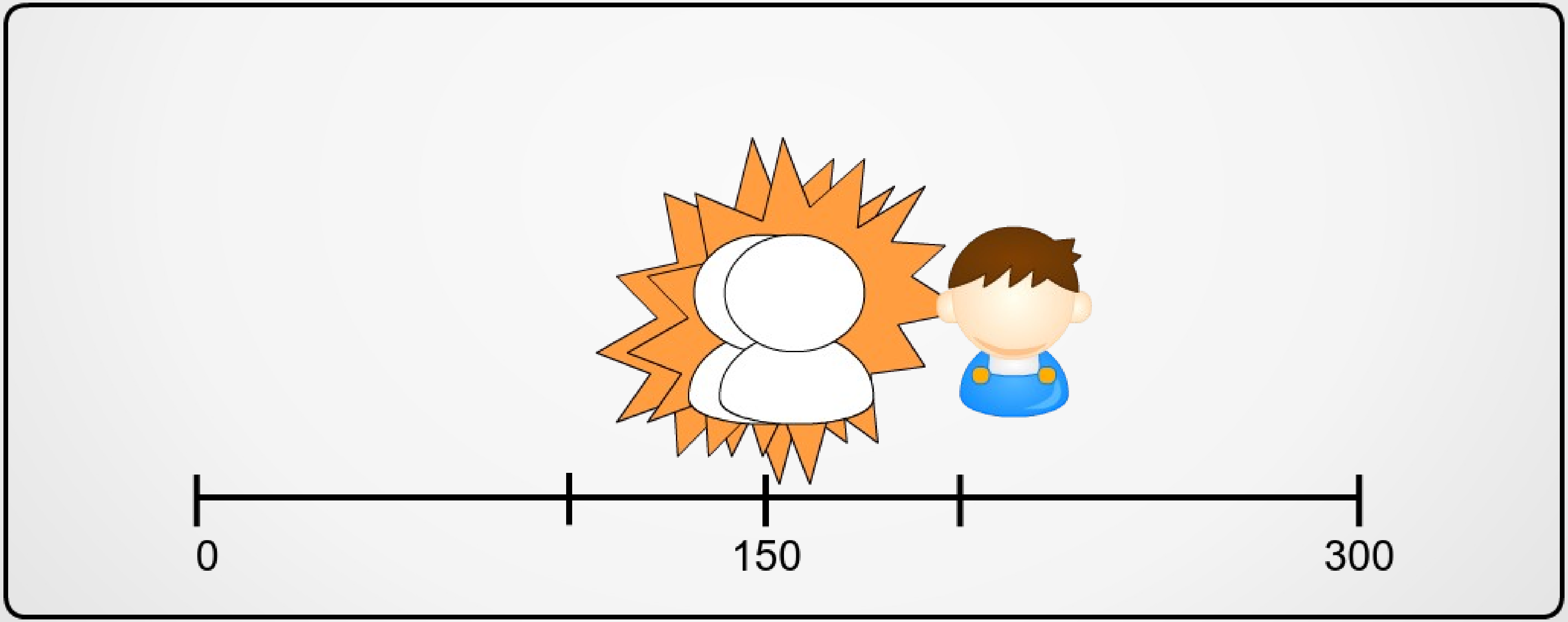
入力例2 (s=8)

- 時刻 18.75(火を移す)



入力例2 (s=8)

- 時刻 18.75(全員に火が移った!)



感想

感想

- 全く同じ配置でも、違う動きをすることがあるようだ
- JOI君たちは立ち止まったり、引き返したりすることがあるようだ
- 同時に複数人の花火に火が付いていることもあるようだ

感想

- 全く同じ配置でも、違う動きをすることがあるようだ
- JOI君たちは立ち止まったり、引き返したりすることがあるようだ
- 同時に複数人の花火に火が付いていることもあるようだ

実は罠！

考察1

- 全く同じ配置でも、違う動きをすることがあるようだ
  - 入力例1は入力例2と同じ動きでも、全員に火を移すことができる

考察2

- JOI君たちは立ち止まったり、引き返したりすることがあるようだ
  - 引き返すことはあるが、立ち止まる必要はない
  - t秒立ち止まりたければ、t/2秒西に進んで、t/2秒東に進めば良い
  - JOI君たちは常に秒速sメートルで進んでよい

考察３

- 同時に複数人の花火に火が付いていることもあるようだ
  - その必要はない
  - 火を移した直後に、火を移しあった2人が同じ方向に進む必要はない
    - それならば別れる時か片方が燃え尽きるときに移せば良い
  - 火を移した直後に、2人が違う方向に進むならば、片方が燃え尽きるまで、火を移すのを延期して良い

教訓

入出力例はたまに意地悪

考察 4(重要)

- 考察3より、着火している花火は、常にひとつである
- まだ着火していない人は、着火している花火に近づかないメリットがない
  - みんな貪欲的に火の付いた花火に近づく
- 火の付いた花火を持ってる人は、次に火を移したい人にむかって近づくのが良い
  - 出会ったら、一緒に並走して、さらに次に火を移したい人に向かって近づくのが良い

考察 4(重要)

- つまり、火をつける順番を決めれば、それぞれの動きも確定できる
- ついに全探索ができるようになった!

小課題0

小課題0 (0点)
$N \leqq 10$

小課題0 (N ≦ 10)　解法

- 火をつける順番を全通り試す
  - O((N-1)!)通りの順番がある
- 各順番については貪欲にシミュレート
  - O(N^2)で容易にシミュレートできる
- あわせてO(N*N!)
- 速度制限に関する二分法をするのでトータルで
  - O(N * N! * log(X))

0点獲得

小課題1 (N ≦ 20)

小課題1 (30点)
N ≦20

小課題1 (N ≦ 20)　考察

- X,Y,Zの順に人が並んでいるときにYを飛ばしてXがZに火を移すようなことは、意味がない
  - 一度Yに火を移してからZに火を移しても、デメリットがない
- よって、着火したことがある人の集合は、開始時の位置について連続した区間となる
- よって一連の操作は、区間を東もしくは西に1つ拡大することの繰り返しなので、火をつける順番は2^N通りしかない

小課題1 (N ≦ 20)　解法

- 2^N通りの火の移し方を全部試す
- あとは小課題0と同様

- トータルの計算量は
  - O(N * 2^N * log(X))

    30点獲得!

小課題2 (N ≦ 1000)

小課題2 (20点)
N ≦1000

小課題2 (N ≦ 1000)　考察

- 着火したことのある人の集合が[L, R]になった直後のことを考える。
- このときL番の人は開始時刻からずっと右向きに動いている
- R番の人も開始時刻からずっと左向きに動いている
- よって、この時刻Tを決定することができる

$$T = (X[R] - X[L]) / 2*s$$

小課題2 (N ≦ 1000)　考察

- このとき[L, R]の人は全員同じ場所にいて並走している
- よって、燃えてる花火のそばには、着火したことがあるかないかを無視すると、全部でR-L+1個の花火がある
- いままで、同時にひとつしか花火は燃えてない条件でT秒やってきたので、これ以上花火が増えないとしたときに、あと何秒くらい持つかも求まる

小課題2 (N ≦ 1000)　考察

- よって以上の情報から
- [L, R]の区間の人がJOI君と同じ場所にいるときに（燃えている花火と並走しているときに）
- [L, R+1]や[L-1, R]に区間を拡大できるかどうかを求めることができる

小課題2 (N ≦ 1000)　考察

- 具体的には[L,R]の間の人が一箇所に集まれる時、その集合を
- [L,R+1]に拡大できる必要十分条件は
  - (X[R+1] – X[L]) / 2s ≦ (R-L+1)t
  - (LとR+1が出会う時刻) ≦ [L,R+1]の花火を燃やし尽くす時間
- [L-1, R]に拡大できる必要十分条件は
  - （X[R] – X[L-1]) / 2s ≦ (R-L+1)t

小課題2 (N ≦ 1000)　解法

- この式を元に[K,K](JOI君のそばにJOI君しかいない状態)から[1,N] (JOI君のそばに全員がいる状態)まで遷移できるかDPでチェックすれば良い
- dp[L][R] = [K,K] から[L,R]に遷移できるかどうか
- dp[L][R] は dp[L+1][R], dp[L][R-1]からO(1)で計算可能
- DPテーブルを埋めるのはO(N^2)
- 二分法をつかうとトータルで
  - O(N^2 log X)

    50点獲得

小課題3 (N ≦ 100000)

小課題3 (50点)
N ≦100000

小課題3 ($N \leqq 100000$)　考察

- 遷移が可能かどうかを調べる式を眺めてみる
  $(X[R+1] - X[L]) / 2s \leqq (R-L+1)t$
- $X[R+1] - 2st(R+1) \leqq X[L] - 2stL$
- $x[i] = X[i] - 2sti$とすると
- $[L, R+1]$に遷移できる$\Leftrightarrow x[L] \geqq x[R+1]$
- 同様に
- $[L-1, R]$に遷移できる$\Leftrightarrow x[L-1] \geqq x[R]$

小課題3 ($N \leqq 100000$)　考察

- $x[i] = X[i] - 2sti$とすると
- [L,R]に遷移できる必要十分条件は
  - $x[L] \geqq x[R]$
- つまり以下のように簡単な問題に置き換えることができる

小課題3 (N ≦ 100000)　言い換え

- x[i] = X[i] – 2stiとする
- L=R=kから開始して
  x[L] ≧ x[R]を保ちながら
  以下の操作でL=1, R=Nに遷移できるか？
- 操作
  - Lをデクリメント、あるいはRをインクリメントする

入力例1

- N=3, K=2, T=50
- X = [0, 200, 300]
- s = 2とすると
- x = [0, 0, -100]

- [2,2] → [1,2] → [1,3] や [2,2]→ [2,3] → [1,3] という遷移ができる

入力例2

- N=3, K=2, T=10
- X = [0, 200, 300]
- s = 8とすると
- X = [0, 40 ,-20]

- [2,2]→ [2,3] → [1,3] という遷移ができる

やったぜ！

わかりやすい問題に
言い換えできた！

新たな

考察

考察

- 保ちたいものはx[L] ≧ x[R]
- よってx[L'] ≧ x[L]となるL' > Lがあるとして、[L,R]から[L',R]に直接じゃなくても移動できるなら、移動しても問題ない
- x[R] ≧ x[R']となるR > R'があった時も同様

- この貪欲を続けていくことを考える

考察

- L≧Kの中でx[L]が最大と成るLをGL
  K≧Rの中でx[R]が最小と成るRをGR　とする
- 先ほどの貪欲な操作をしつづけたときに、[GL, GR]に到達したら、もうこれ以上同じ操作はできない
- 逆に[GL, GR]に到達できなかったら？
  - →どのような遷移をしても[1,N]に到達できない

- [K,K]から[GL,GR]に遷移できなければ[1,N]にも到達できない

考察

- 遷移は逆に辿ることができるので、この問題は以下のように言い換えることもできる。
- L=1, R=Nから開始して、以下の操作を繰り返して[K,K]に遷移できるか？
  - Lをインクリメント、あるいはRをデクリメント
- 先ほどと同様の貪欲な操作で同じ議論ができる
- [1,N]から[GL,GR]に遷移できなければ[K,K]まで遷移できない

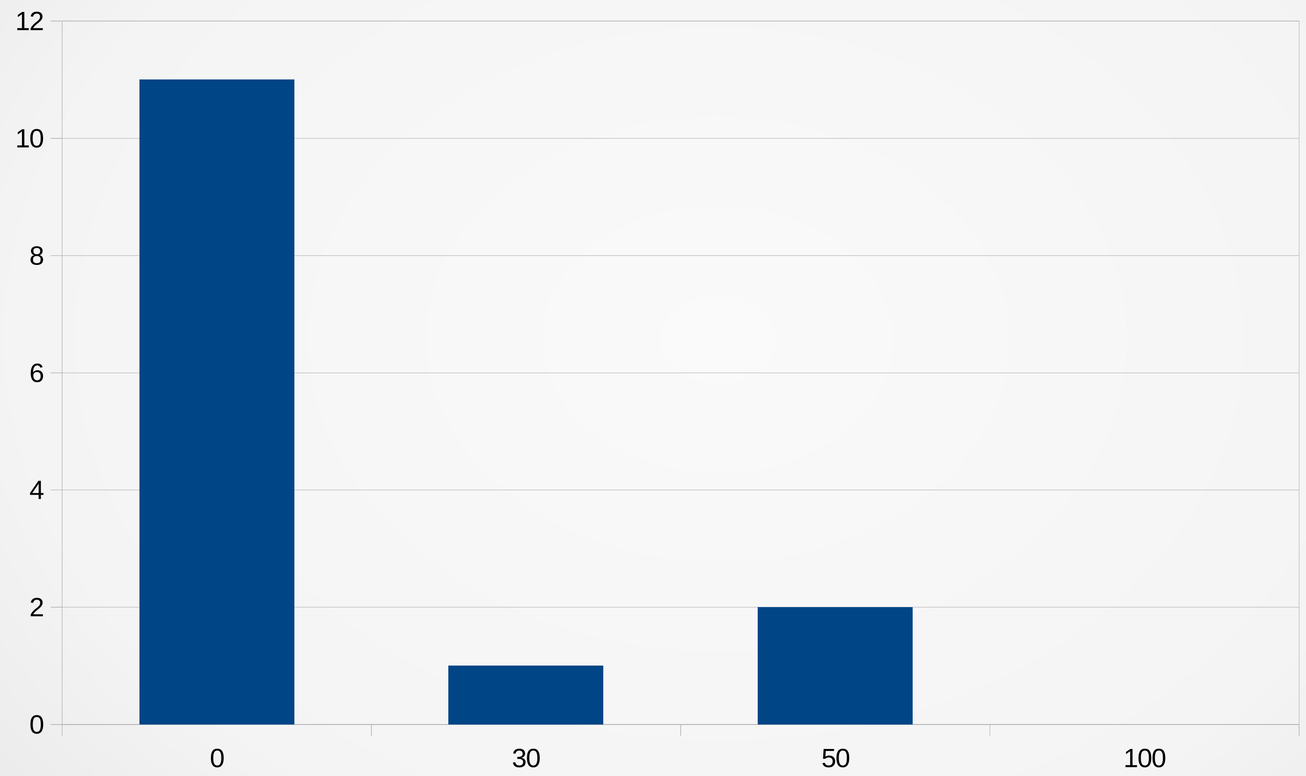
考察まとめ

- Lに関しては増やせる限り増やす、Rに関しては減らせる限り減らすという貪欲な遷移を考える
- 以下の2つは同値
  - [K,K]から[L,R]に行けるかどうか
  - [K,K]からも[1,N]からも貪欲な遷移で[GL, GR]に到達できる
- 後者はO(N)でシミュレートすることができる

満点解法

- 言い換えた後の問題で
- [K,K]と[1,N]の両方から[GL, GR]に遷移できるかどうかを調べる
- それが、元の問題で全員に火を着火できるかと同値
- シミュレーションはO(N)でできる
- 二分法と合わせてトータルで
- O(NlogX)
  - 100点獲得!

得点分布



12
10
8
6
4
2
0
0
30
50
100