解释。

烟花在手

烟火

问题简介。

- N 人排成一条直线观看烟花。
- 烟花只燃烧 T 秒。
- 只有 Hajime JOI 的烟花才能点燃。
- 你们必须以多快的速度移动，才能成功地将火种传递给对方，并点燃每个人的烟花？
- 查找最低限速。
- N≤100,000

问题简介。

- N 人排成一条直线观看烟花。
- 烟花只燃烧 T 秒。
- 只有 Hajime JOI 的烟花才能点燃。
- 你们必须以多快的速度移动，才能成功地将火种传递给对方，并点燃每个人的烟花？
- 查找最低限速。
- N≤100,000

考虑 0

- 如果你可以在给定的速度限制 s 下向所有人传送火力，那么你也可以在任何速度限制大于 s 的情况下向所有人传送火力。
- 找出 s 中能将火传递给每个人的最小值。
- 有一些算法很容易用于这类优化问题。

考虑 0

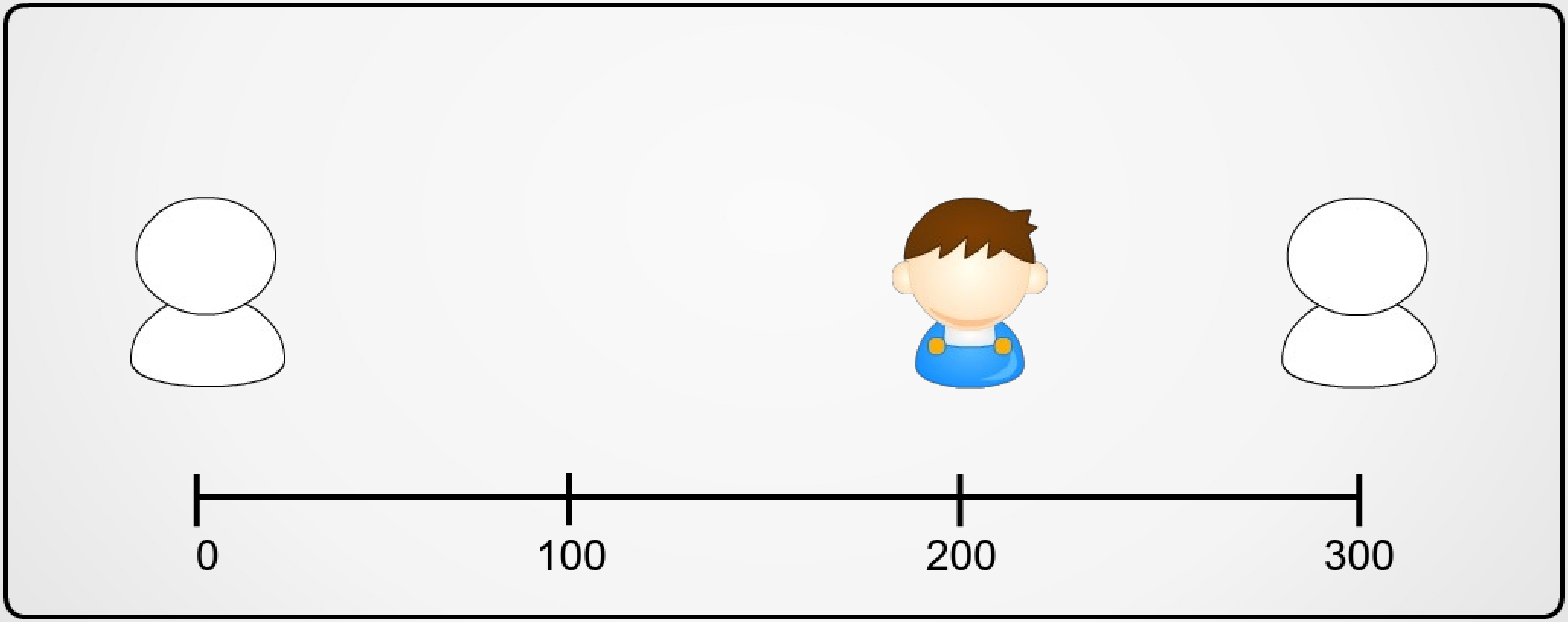
- 如果你可以在给定的速度限制 s 下向所有人传送火力，那么你也可以在任何速度限制大于 s 的情况下向所有人传送火力。
- 找出 s 中能将火传递给每个人的最小值。
- 有一些算法很容易用于这类优化问题。

对立

输入示例

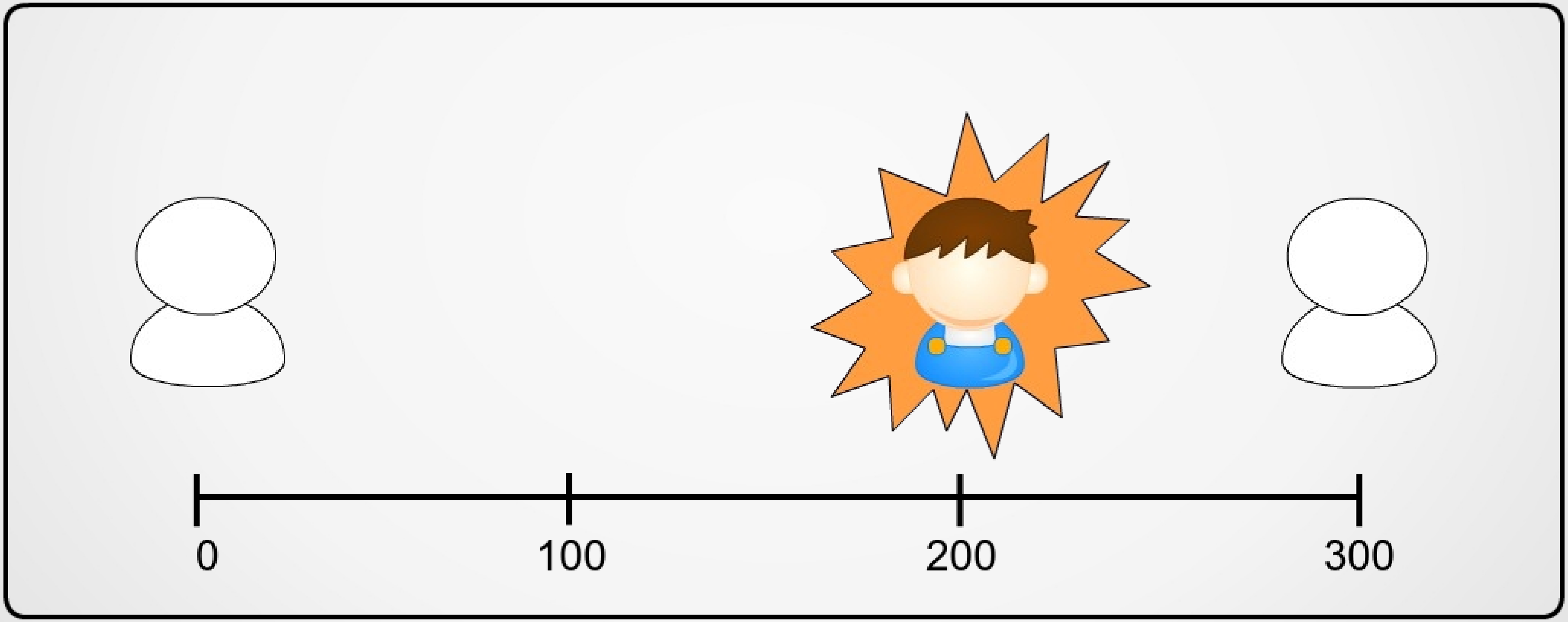
输入示例 1.

- N=3，K=2，T=50



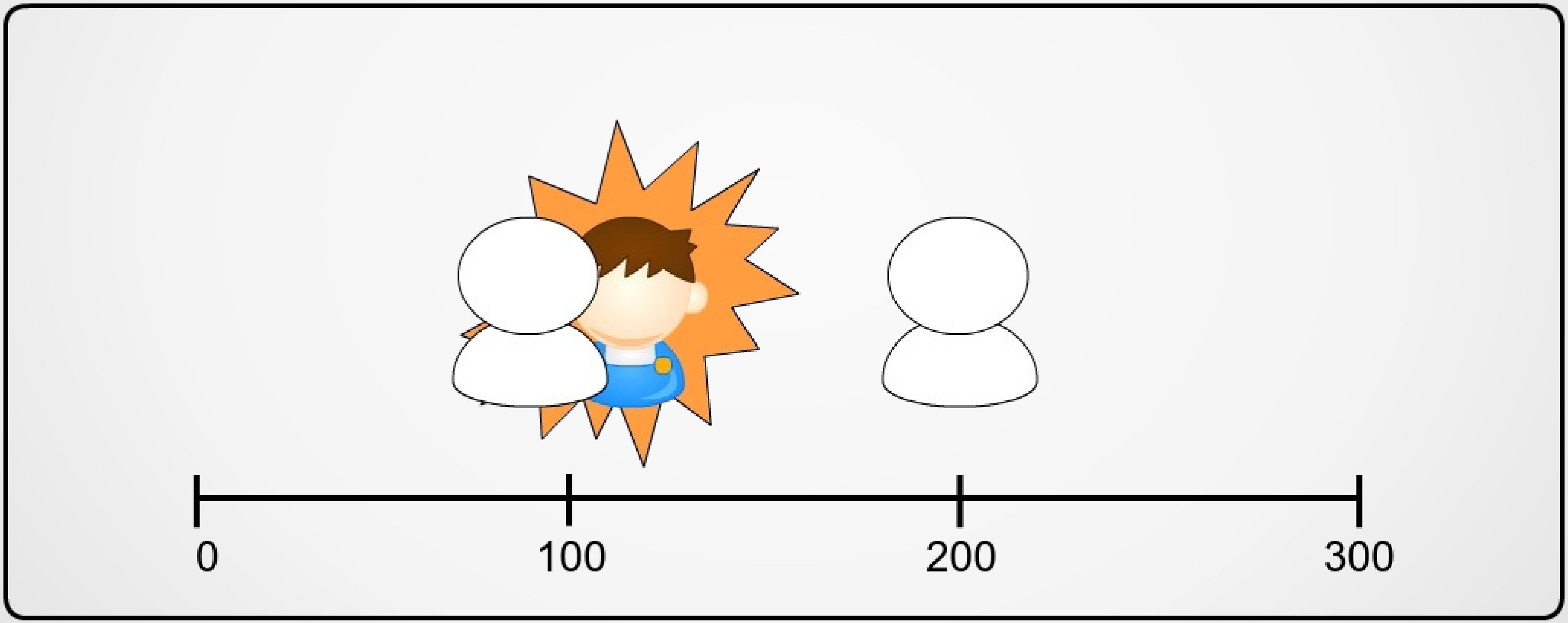
输入示例 1（s=2）

- 时间 0（JOI 君点火）



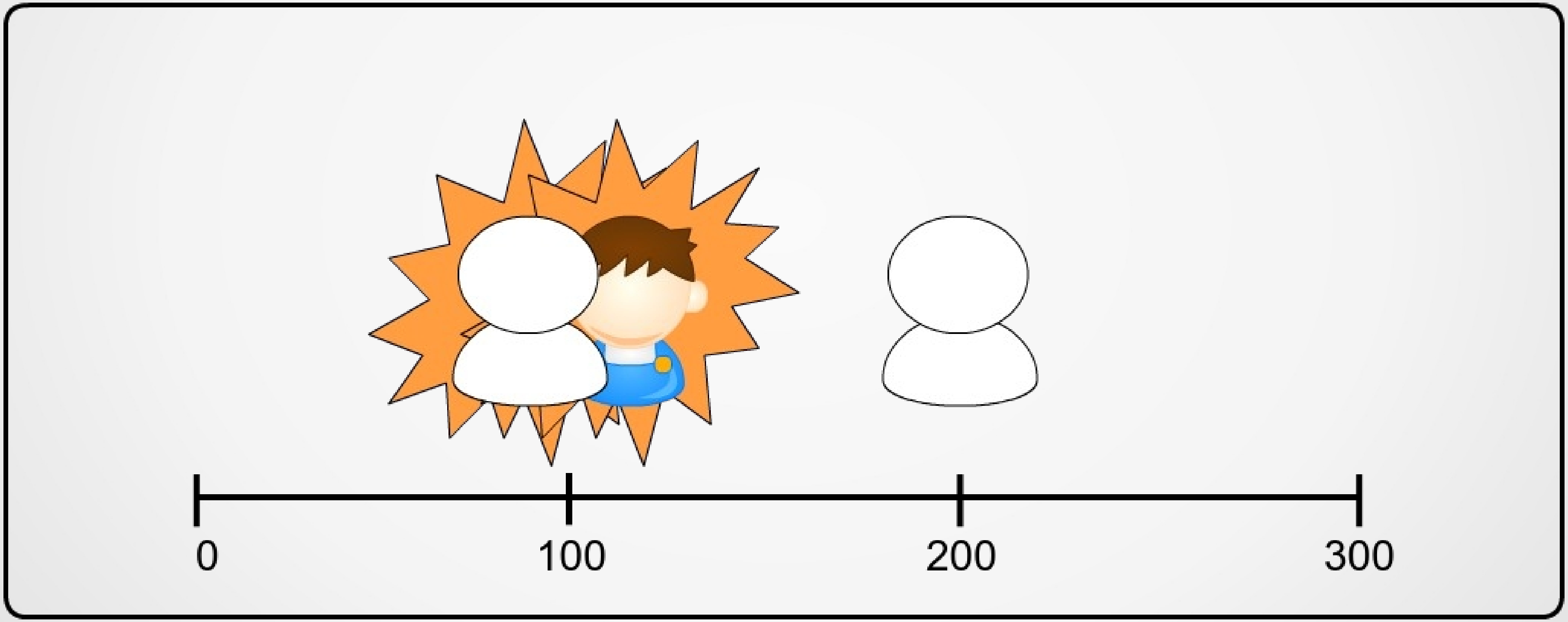
输入示例 1（s=2）

- 时间 50



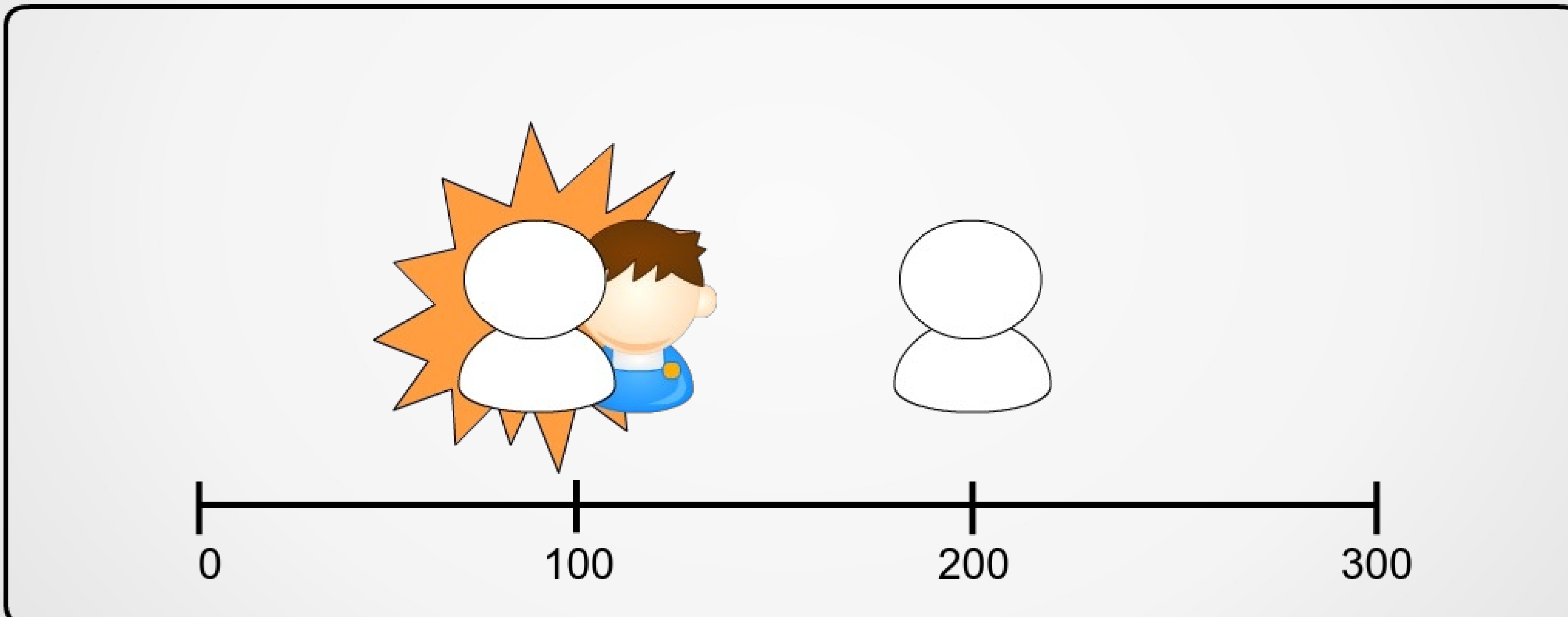
输入示例 1（s=2）

- 时间 50（转移火力）



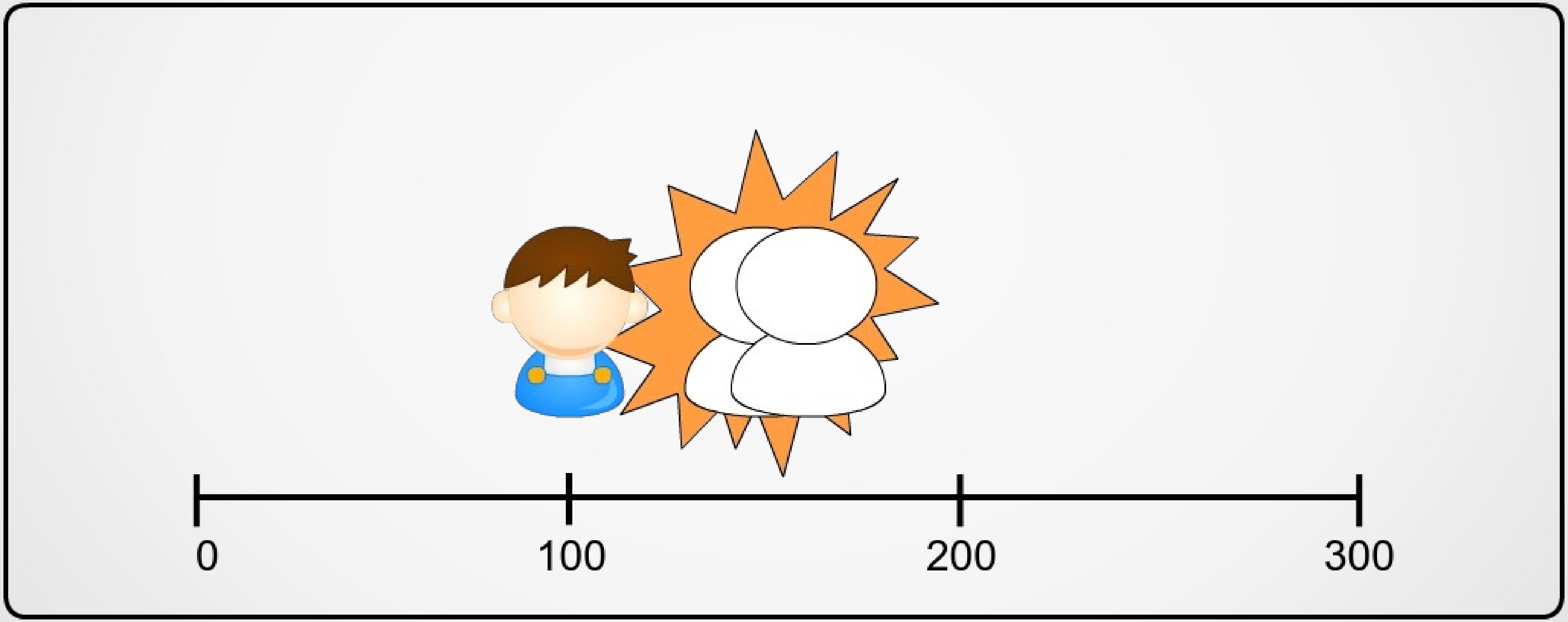
输入示例 1（s=2）

- 时间 50（JOI 君烧尽）



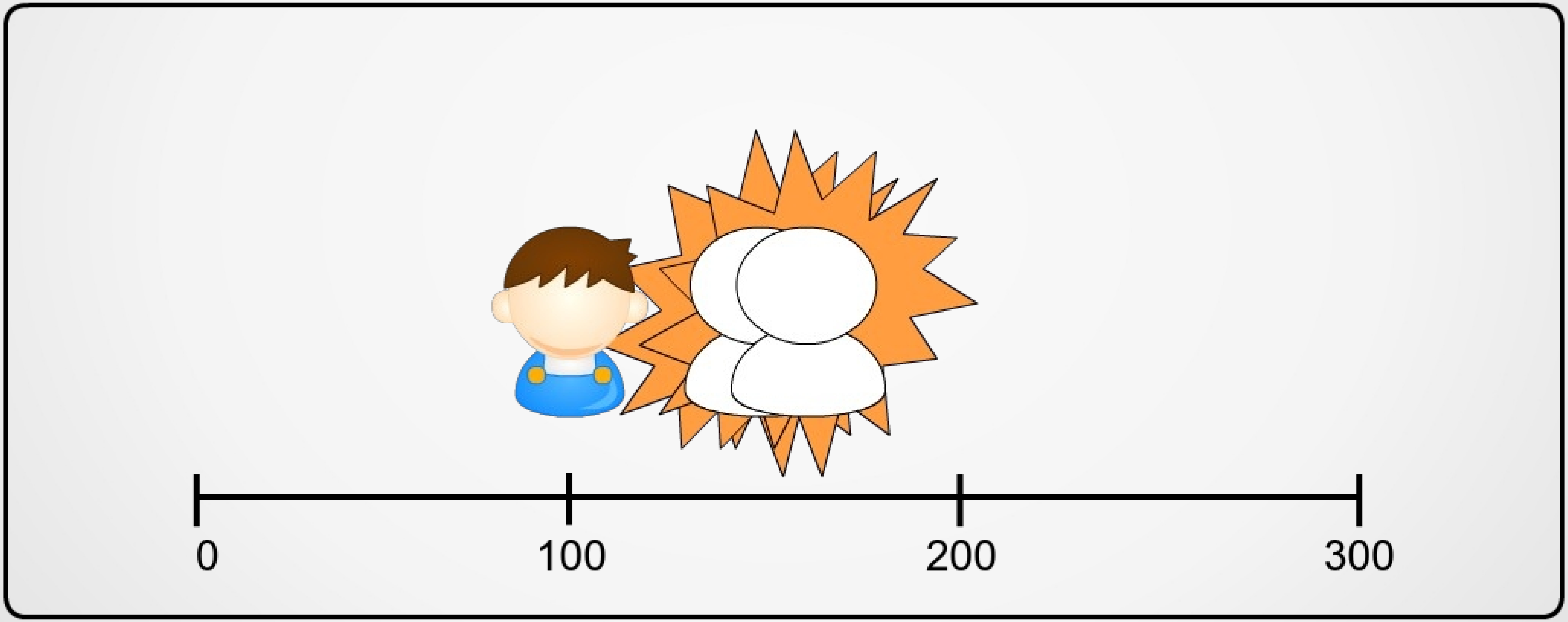
输入示例 1（s=2）

- 时间 75



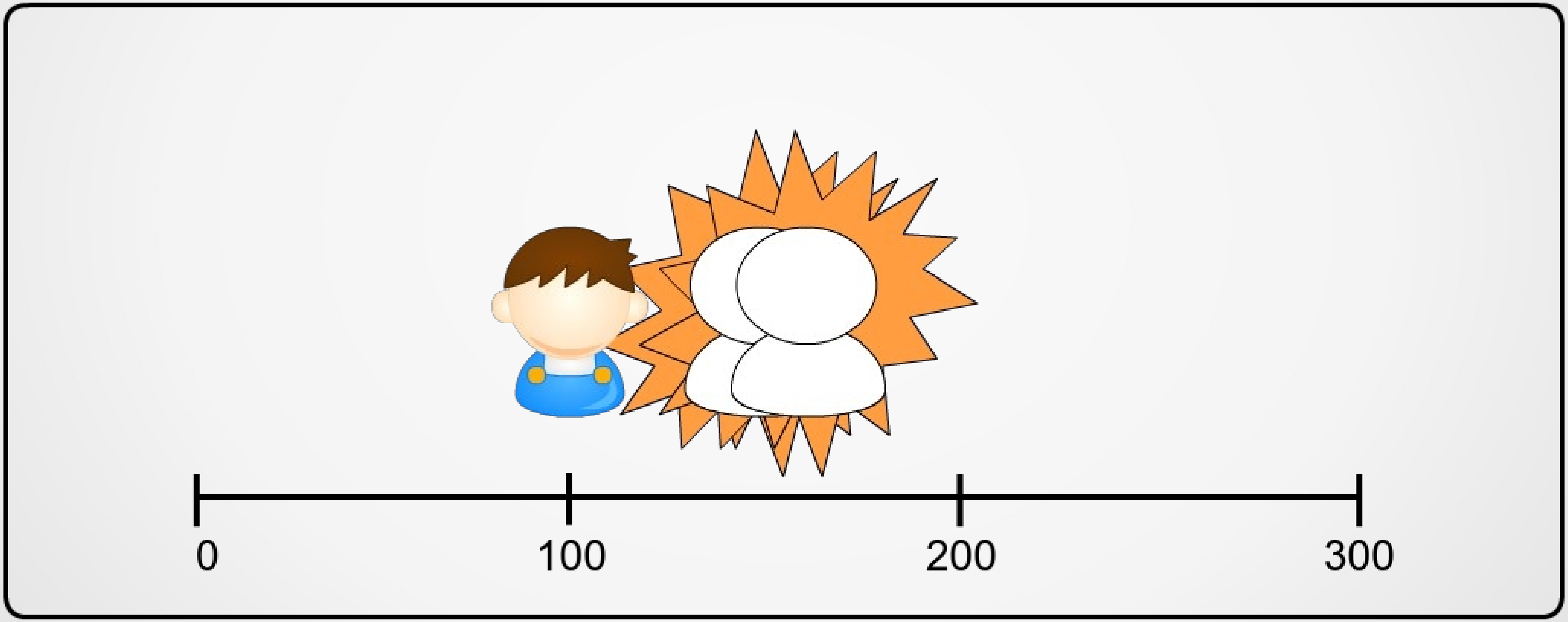
输入示例 1（s=2）

- 时间 75（转火）。



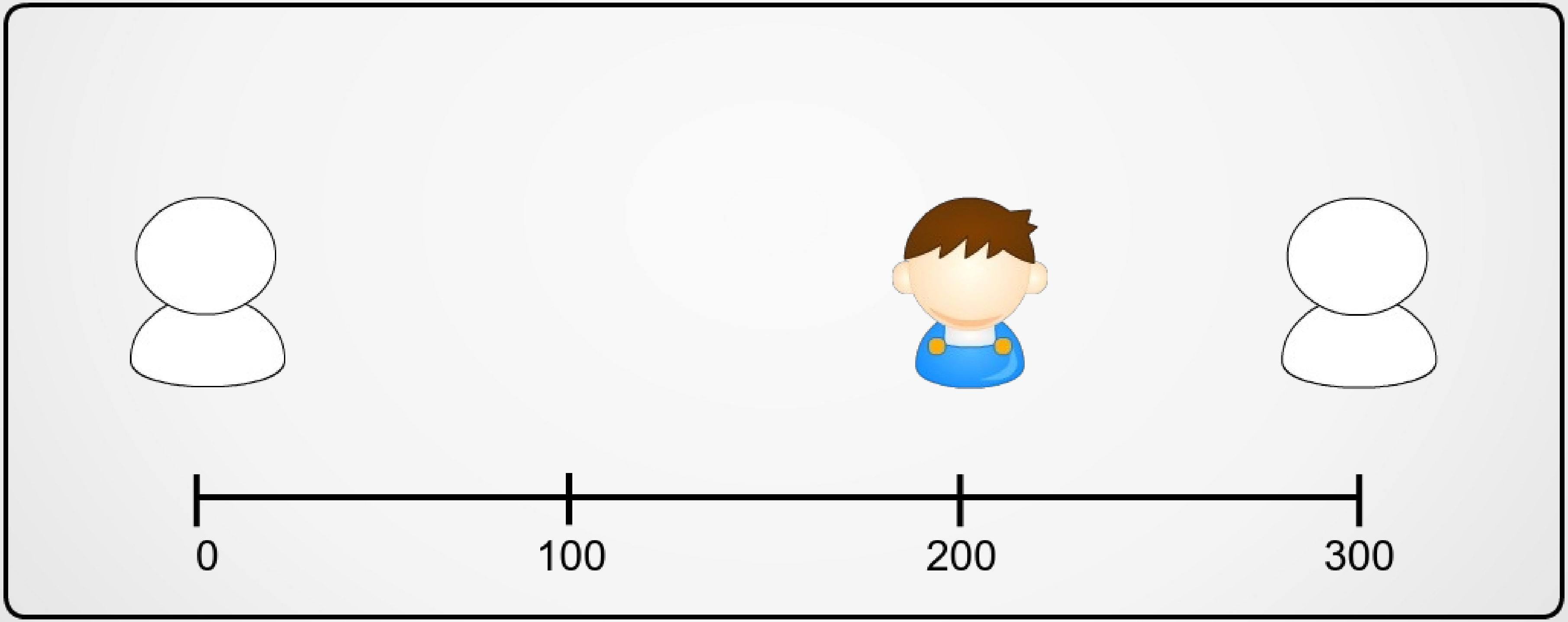
输入示例 1（s=2）

- 时间 75（向所有人开火！）



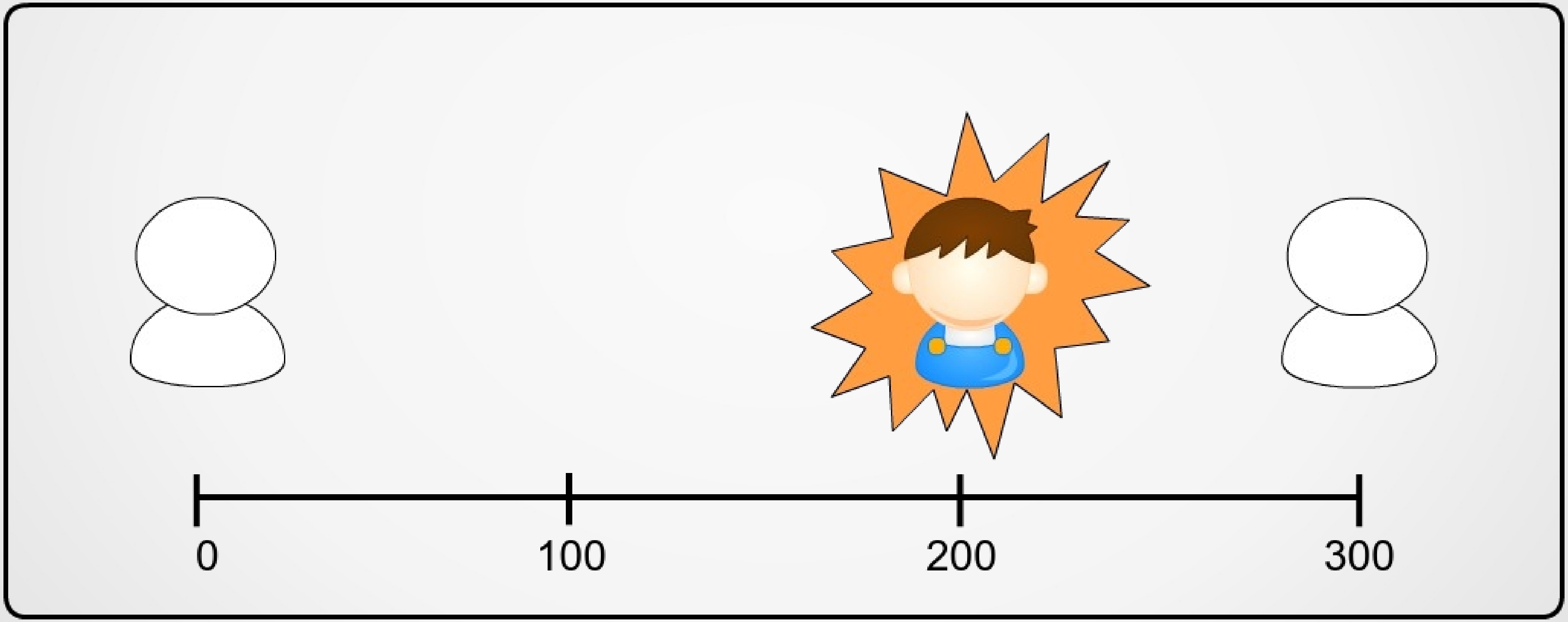
输入示例 2。

- N=3，K=2，T=10



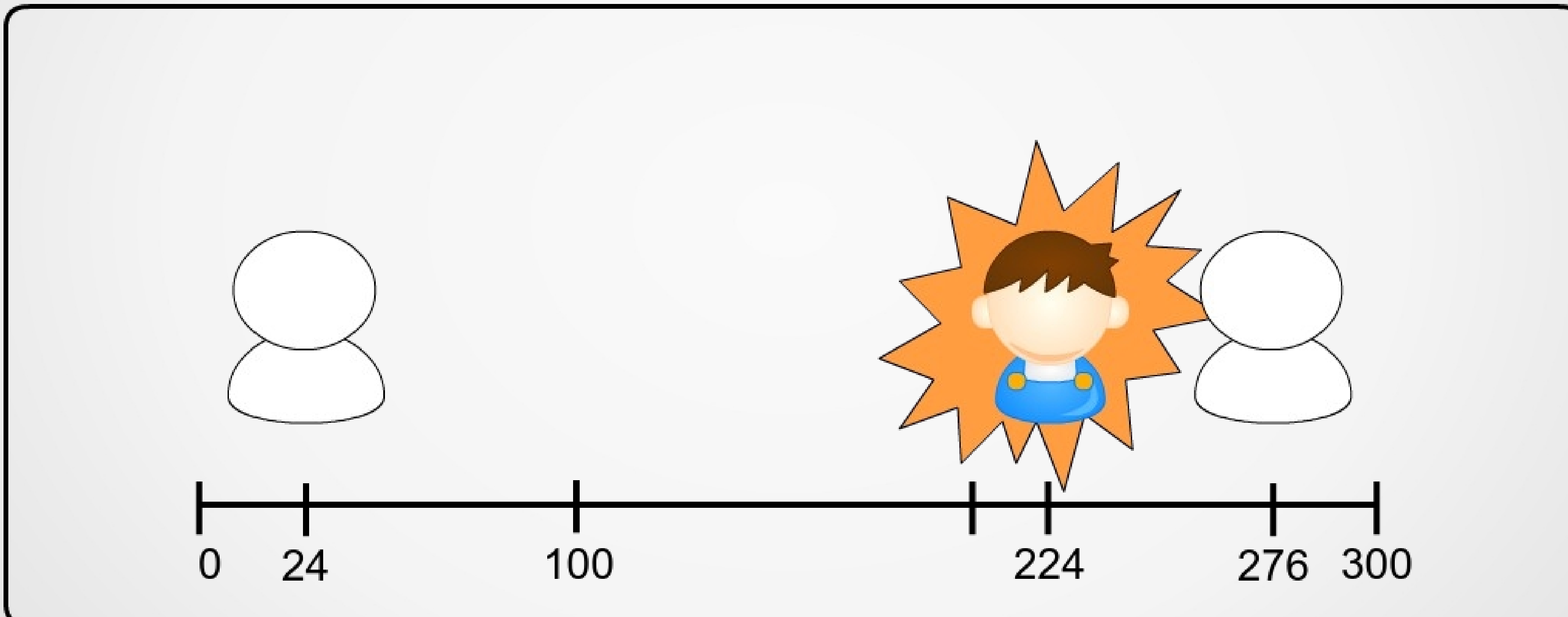
输入示例 2（s=8）

- 时间 0（JOI 君点火）



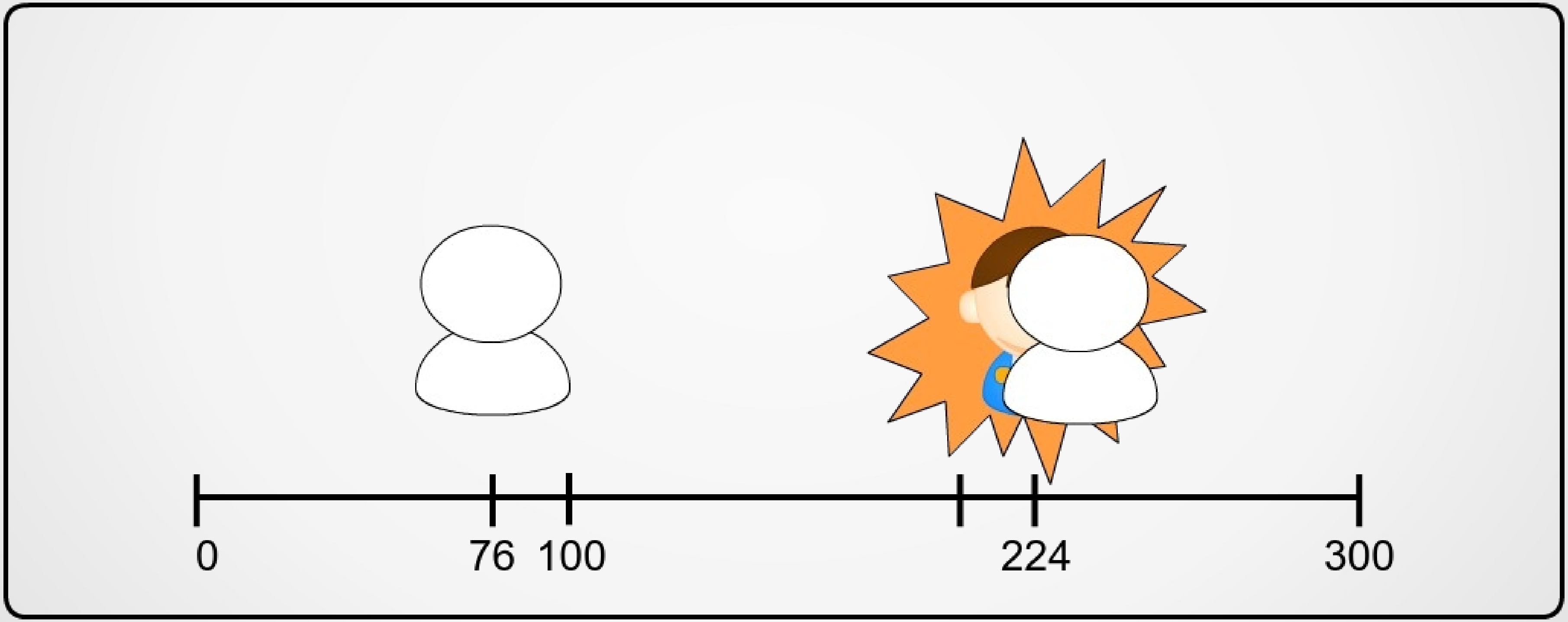
输入示例 2（s=8）

- 时间 3（JOI-kun 停止）



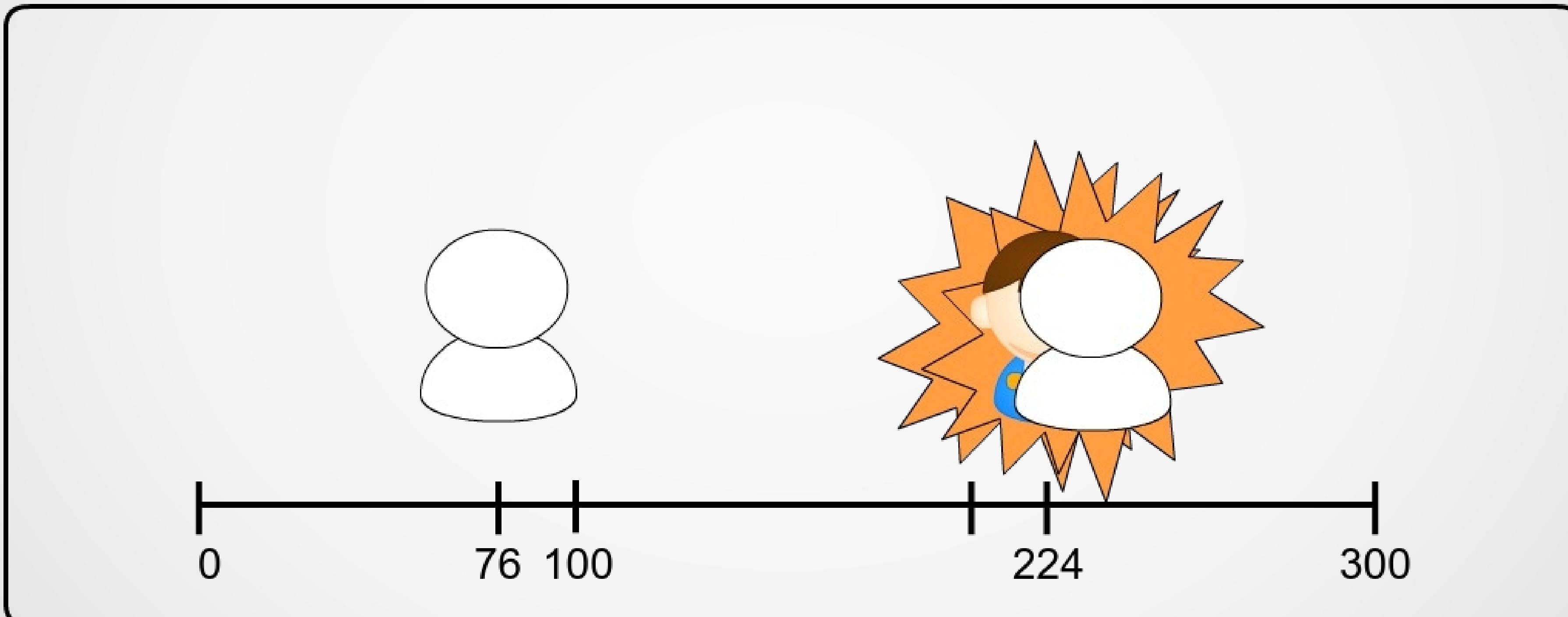
输入示例 2（s=8）

- 时间 9.5



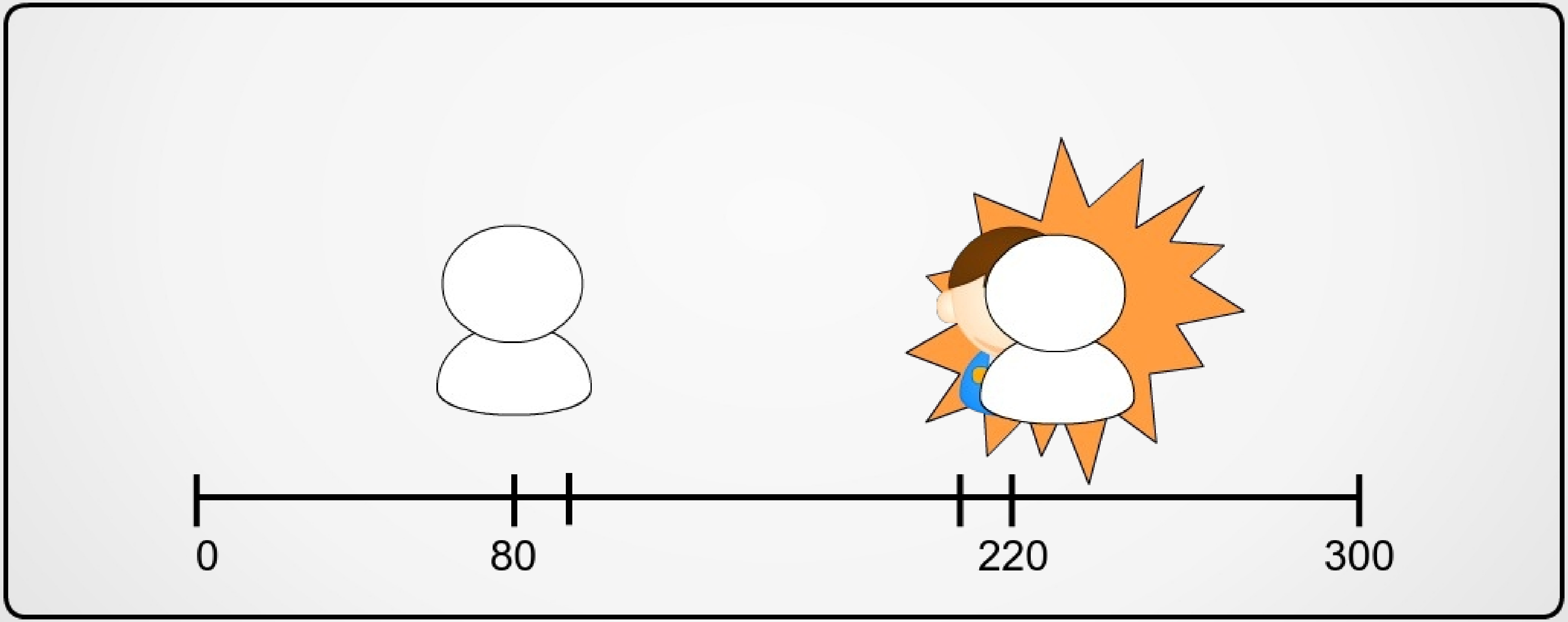
输入示例 2（s=8）

- 时间 9.5（火力转移，JOI-kun 恢复移动）。



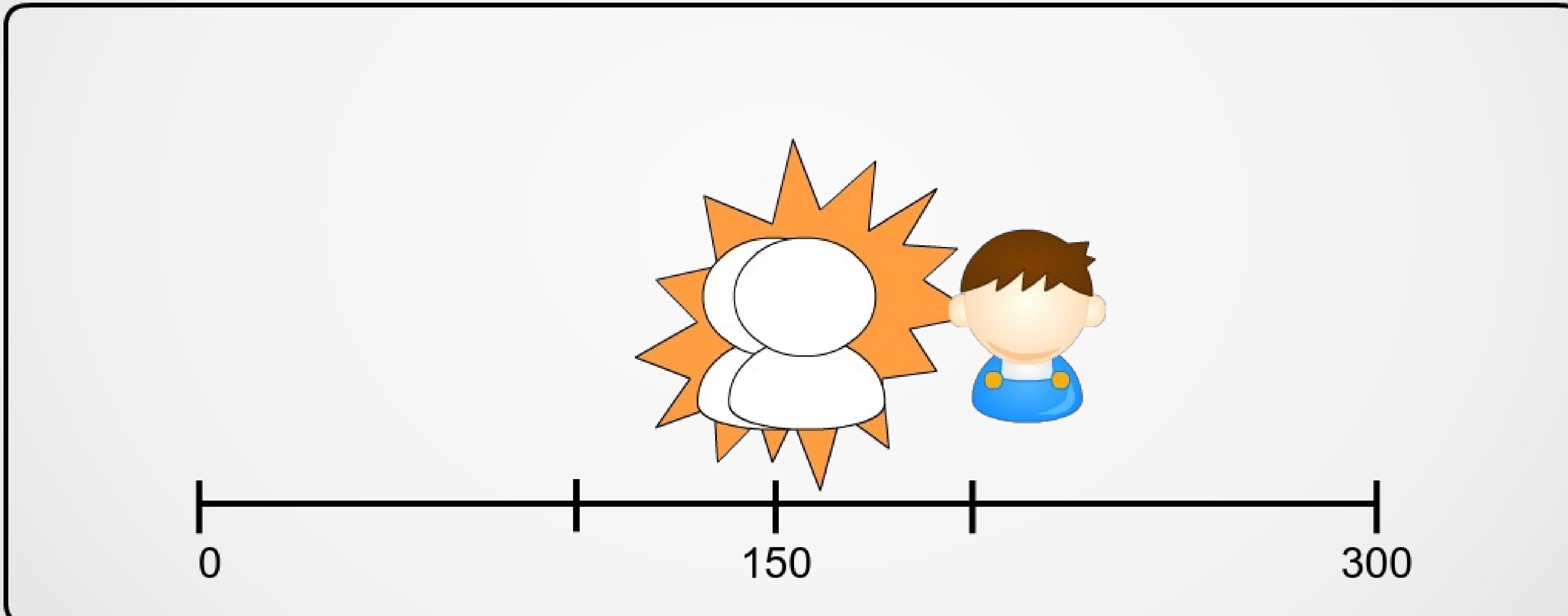
输入示例 2（s=8）

- 时间 10（JOI 君烧尽）。



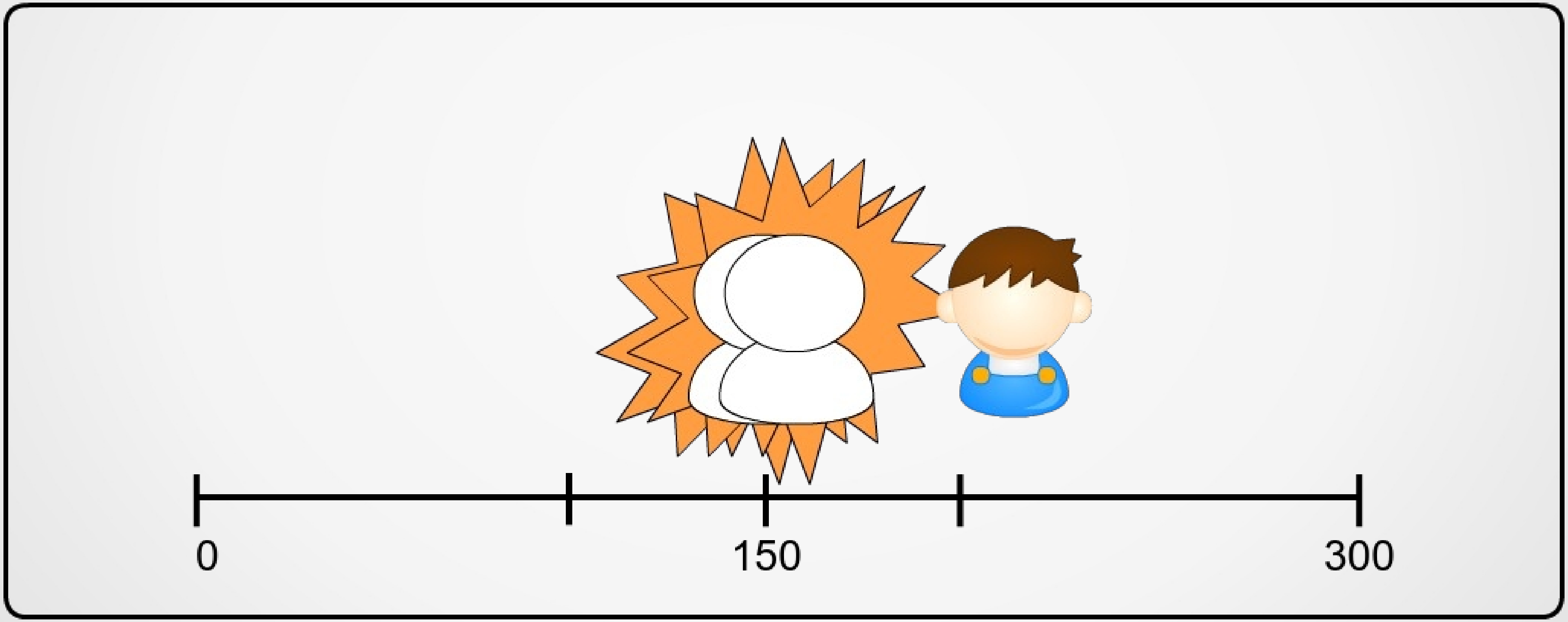
输入示例 2（s=8）

- 时间 18.75



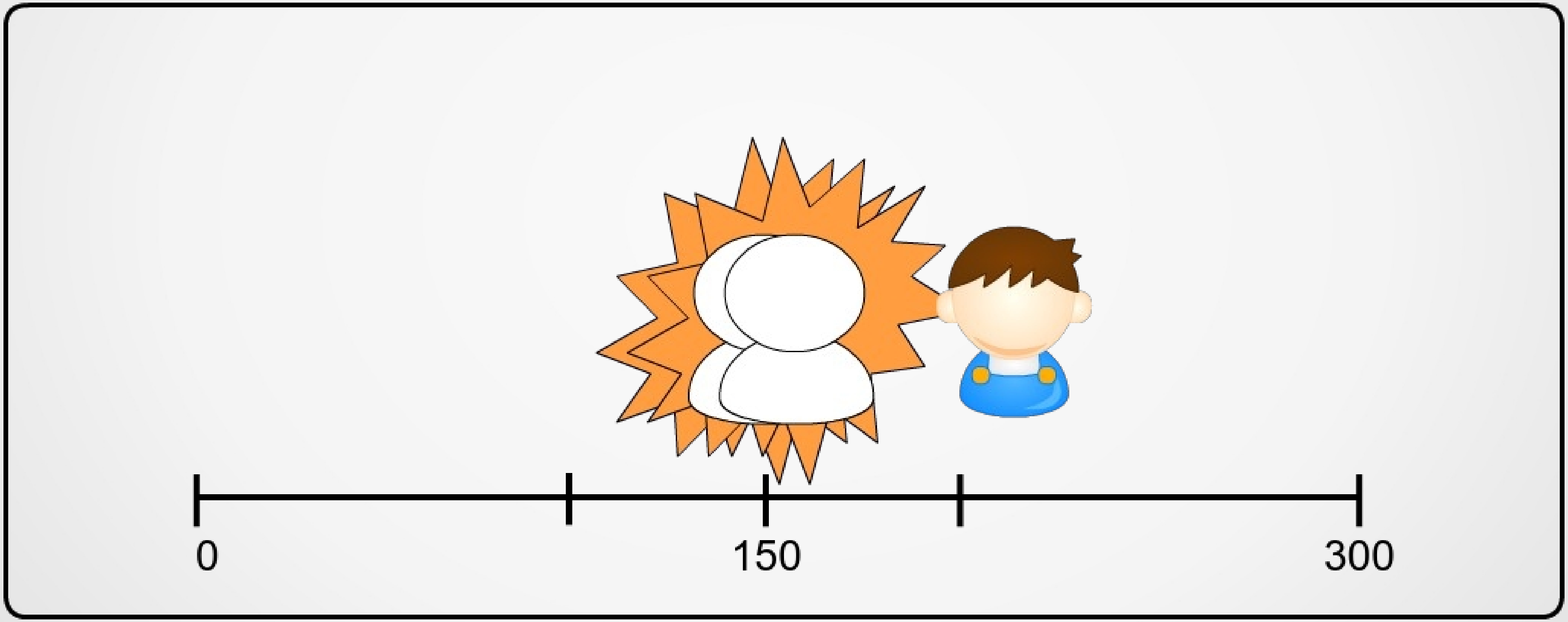
输入示例 2（s=8）

- 时间 18.75 (转火)



输入示例 2（s=8）

- 时间 18.75 (火已转移到所有人手中!)



印象

印象

- 完全相同的安排似乎效果不同。
- 你和你的团队似乎停了下来，转过身去。
- 似乎有时不止一个人的烟花同时点燃。

印象

- 完全相同的安排似乎效果不同。
- 你和你的团队似乎停了下来，转过身去。
- 似乎有时不止一个人的烟花同时点燃。

事实上，这是一个陷阱！

考虑 1.

- 完全相同的安排似乎效果不同。
  - 即使与输入示例 2 的动作相同，输入示例 1 也能将火传递给每个人。

审议 2.

- 你和你的团队似乎停了下来，转过身去。
  - 我们可以回头，但不必停止。
  - 如果你想停 t 秒，向西走 t/2 秒，然后向东走 t/2 秒。
  - JOI 你和你的团队在任何时候都可以以每秒 s 米的速度前进。

考虑因素 3

- 似乎有时不止一个人的烟花同时点燃。
  - 没必要
  - 转移火源后，转移火源的两人无需立即朝同一方向前进。
    - 这样，当你们分开或其中一个烧坏时，就可以将它们转移。
  - 如果在转移火种后, 两者立即向不同的方向移动，您可以推迟转移火种，直到其中一个烧尽。

课程

输入/输出示例有时很讨厌。

考虑因素 4（关键）。

- 从考虑 3 开始，总有一个烟花被点燃
- 尚未点燃烟花的人没有远离点燃烟花的好处。
  - 每个人都贪婪地靠近点燃的烟花。
- 拿着点燃的烟花的人应该走近下一个想要转移火种的人。
  - 当你遇到他们时，你可以与他们并肩奔跑，然后向下一个你想转移火种的人靠近。

考虑因素 4（关键）。

- 换句话说，一旦确定了点火的顺序，就可以固定各自的动作。
- 最后，就可以进行整个搜索了！

分任务 0.

分任务 0（0 分）

$N \leq 10$

分任务 0（N ≤ 10）解决方案。

- 试试点火的整个顺序。
  - O((N-1)!)有一个街道顺序。
- 贪婪地模拟每一笔订单。
  - 只需 O(N^2) 便可轻松模拟
- 加上 O(N*N!)。
- 这完全是因为速度限制的二分法。
  - O(N * N! * log(X)) 0 积分获得

子任务 1（$N \le 20$）

子任务 1（30 分）

$N \le 20$

分任务 1（N ≤ 20） 审议。

- 当人们按照 X、Y、Z 的顺序排列时，为了让 X 把火传给 Z 而跳过 Y 是没有意义的。
  - 火势一旦转移到 Y，再转移到 Z 就没有任何不利之处了。
- 因此，点过火的人的集合是一个相对于起始位置的连续部分
- 因此，操作顺序是向东或向西扩大一个部分的重复，所以只有 2^N 个可能的开火顺序。

子任务 1（N ≤ 20）解决方案。

- 尝试所有 2^N 种传递火种的方法。
- 其余部分与子任务 0 相同。

- 总计算量为
  - O(N * 2^N * log(X))
    获得 30 个积分！

子任务 2（$N \le 1000$）

子任务 2（20 分）
$N \le 1000$

子任务 2（N ≤ 1000） 审议。

- 考虑一下一组曾经点燃过 [L, R] 的人的直接后果。
- 此时，人 L 从开始时间起就一直在向右移动。
- 从开始计时起，R 上的人也一直在向左移动。
- 因此，可以确定此时的 T = (X[R] - X[L]) / 2*s

子任务 2（N ≤ 1000） 审议。

- 此时，[左、右]中的所有人都在同一地点，并且平行奔跑
- 因此，如果我们不考虑是否曾在燃烧的烟花附近点燃过烟花，那么总共有 R-L+1 枚烟花
- 到目前为止，只有一个烟花同时燃烧了 T 秒钟，所以如果没有更多的烟花，我们就可以知道还剩多少秒钟。

子任务 2（N ≤ 1000） 审议。

- 因此，根据上述信息
- 当[L，**R**]部分中的某人与 JOI 先生在同一地点时，[L，**R**]。(在燃烧的烟花旁奔跑时)
- 可以问区间是否可以扩展到 [L, R+1] 或 [L-1, R]。

子任务 2（N ≤ 1000） 审议。

- 具体来说，当 [L,R] 之间的人可以聚集在一个地方时, 集合为
- 扩展到 [L,R+1] 的必要条件和充分条件是
  - (X[R+1] - X[L]) / 2s ≤ (R-L+1)t
  - (L 和 R+1 相遇的时间） ≤ [L,R+1] 烟花燃尽的时间。
- 扩展到 [L-1，R] 的必要条件和充分条件是
  - (X[R] - X[L-1]) / 2s ≤ (R-L+1)t

子任务 2（N ≤ 1000）解决方案。

- 根据这个等式，民主党可以检查是否可以从 [K,K]（只有 JOI 靠近 JOI）过渡到 [1,N]（所有人都靠近 JOI）。
- dp[L][R] = 能否从[K,K]过渡到[L,R]
- dp[L][R] 可以通过 O(1) 从 dp[L+1][R], dp[L][R-1] 计算出来。
- O(N^2) 填充 DP 表。
- 二分法共提供了
  - O(N^2 log X)
    奖励 50 分

子任务 3（$N \le 100000$）

子任务 3（50 分）
$N \le 100000$

子任务 3（$N \leq 100000$） 审议。

- 查看等式，检查是否可能发生转换 $(X[R+1] - X[L]) / 2s \leq (R-L+1)t$
- $X[R+1] - 2st(R+1) \leq X[L] - 2stL$
- 如果 $x[i] = X[i] - 2sti$，那么
- 可以过渡到 $[L, R+1] \Leftrightarrow x[L] \geq x[R+1]$。
- 类似于
- 可以过渡到 $[L-1, R] \Leftrightarrow x[L-1] \geq x[R]$。

子任务 3（$N \leq 100000$） 审议。

- 如果 x[i] = X[i] - 2sti，那么
- 过渡到 [L,R] 的必要条件和充分条件是

  - $x[L] \geq x[R]$.
- 这就意味着可以用下面的简单问题来代替它

子任务 3（N ≤ 100000） 意译。

- x[i] = X[i] - 2sti。
- 从 L=R=k 开始

  同时保持 x[L] ≥ x[R]。

  能否使用以下操作过渡到 L=1，R=N？
- 操作

  - 减少 L 或增加 R。

输入示例 1.

- N=3，K=2，T=50
- X = [0, 200, 300].
- 假设 s = 2。
- x = [0, 0, -100].
- [2,2] → [1,2] → [1,3] 或 [2,2] → [2,3] → [1,3]。

  过渡过程如下

输入示例 2。

- N=3，K=2，T=10
- X = [0, 200, 300].
- 假设 s = 8。
- X = [0, 40 ,-20]
- 可以进行 [2,2] → [2,3] → [1,3] 的转换

我做到了

对于简单易懂的问题。

可以转述！

新。

审议

审议

- 你想要保留的是 x[L] ≥ x[R]。
- 因此，如果存在一个 L' > L，使得 x[L'] ≥ x[L], 并且如果我们可以从 [L,R] 直接移动到 [L',R], 但是
- 如果存在一个 R > R'，使得 x[R] ≥ x[R']，情况也是如此。
- 考虑继续这种贪婪。

审议

- 为 L 的 GL，使得 x[L] 在 L≥K 中最大

  让 GR 让
- 如果继续进行前面描述的贪心操作，到达 [GL，GR]，就不能再进行同样的操作了。
- 反过来，如果 [GL, GR] 无法到达呢？
  - → 无法通过任何过渡到达 [1,N]。
- 如果无法从 [K,K] 过渡到 [GL,GR]，则无法到达 [1,N]。

审议

- 由于过渡是可以倒推的，因此问题可以重新表述如下。
- 从 L=1，R=N 开始，能否重复以下操作过渡到 [K,K]？
  - 递增 L 或递减 R

- 同样的论点也可以用之前的贪婪操纵来解释。
- 如果无法从 [1,N] 过渡到 [GL,GR]，也就无法过渡到 [K,K]。

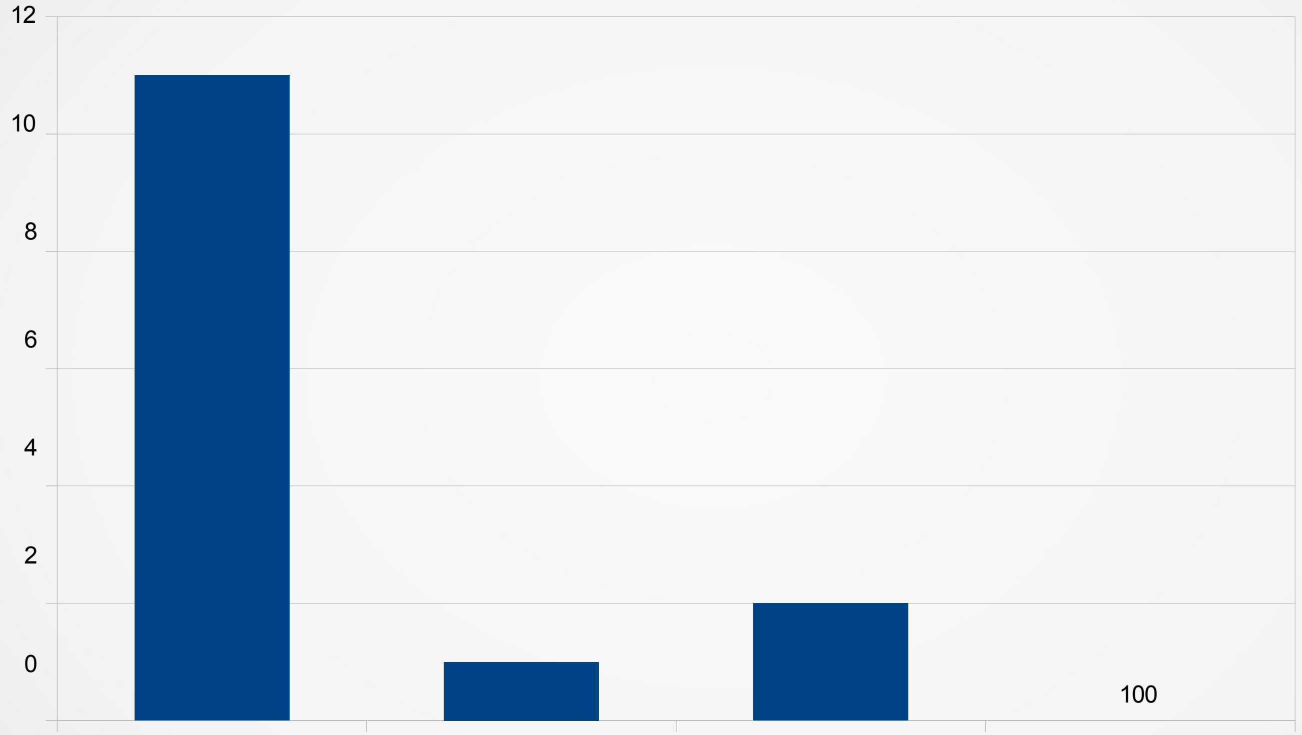
考虑因素摘要

- 考虑尽可能增加 L 和尽可能减少 R 的贪婪过渡。
- 以下两个是等价的
  - 你到底能不能从 [K,K] 到 [L,R]？
  - 从[K，K]和[1，N]都可以通过贪心转换到达[GL，GR]。
- 后者可通过 O(N)

(得满分的方法

- 关于转述后的问题。
- 检查是否可以从 [K, K] 和 [1, N] 过渡到 [GL, GR]。
- 这就相当于能够在原问题中为每个人点燃一把火。
- 仿真可在 O(N) 内完成。
- 总之，加上二分法
- O(NlogX)
  - 得 100 分！

得分分布


12
10
8
6
4
2
0
100