即时战略 (rts)

清华大学 计算机科学与技术系 王逸松

题目大意

- **交互题**

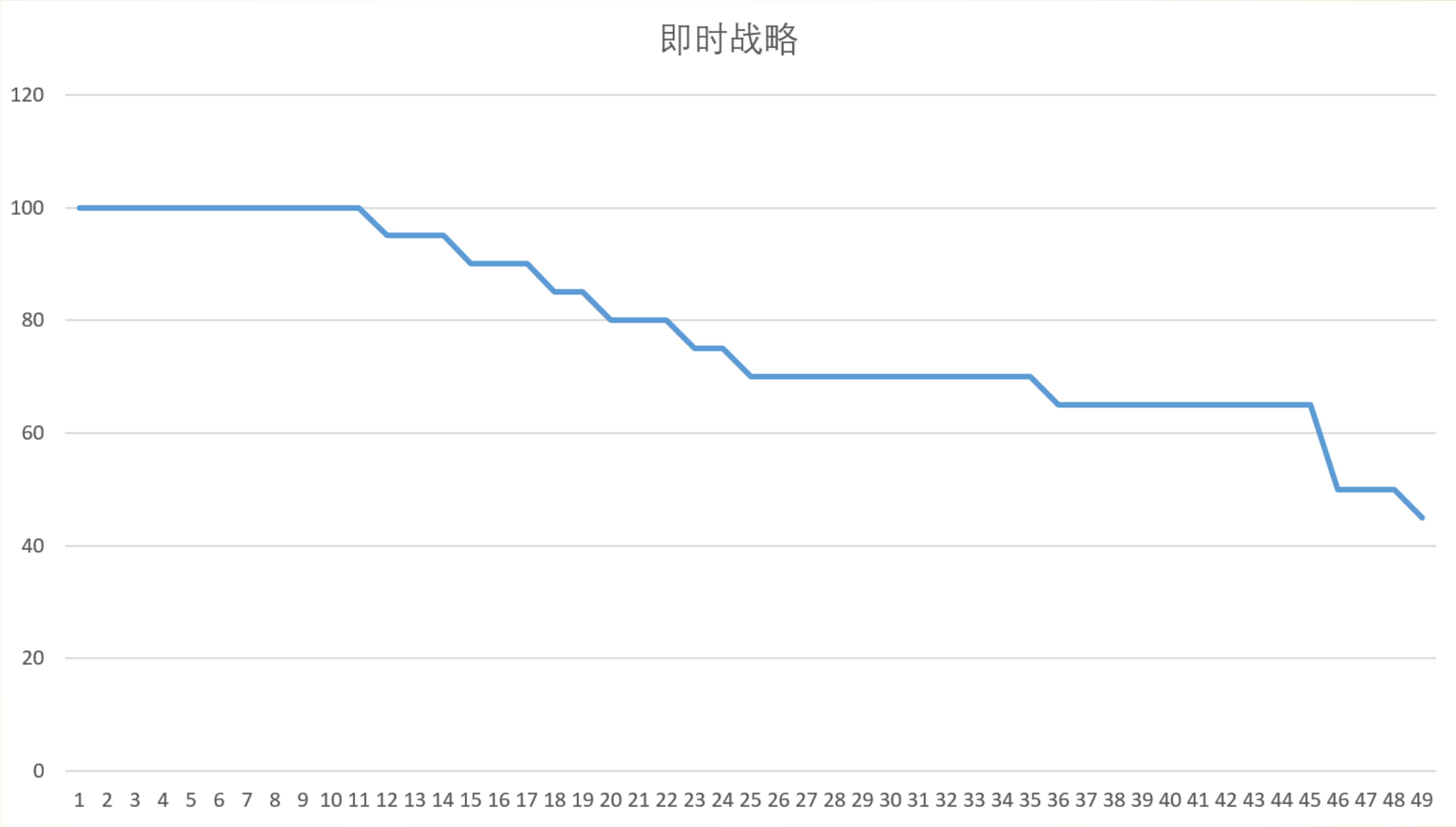
- 一棵 n 个结点的树，不知道树的形态
- 一开始只有 1 号结点是“已知的”
- 每次操作可以选择任何一个**已知的**结点 x，和一个不同于 x 的**任意结点** y
- 将 x 到 y 的路径上第二个结点标记为已知的
- 要求在 T 次操作之内，将所有结点 (1 ~ n) 都标记为已知的

- 操作通过调用交互库的函数来完成

数据范围

- **第一部分（$4 * 5 = 20$ 分）**
  - 树的形态没有限制
  - $n \leqslant 100, T = 10000$
- **第二部分（$3 * 5 = 15$ 分）**
  - 树的形态为完全二叉树，1号结点为根结点
  - $n \leqslant 250000, T \leqslant 500 * 1e4$
- **第三部分（$6 * 5 = 30$ 分）**
  - 树的形态是一条链
  - $n \leqslant 300000$, 最后一个测试点 $T = n + 20$
- **第四部分（$7 * 5 = 35$ 分）**
  - 树的形态没有限制
  - $n \leqslant 300000, T \leqslant 500 * 1e4$

得分情况 (集训队)



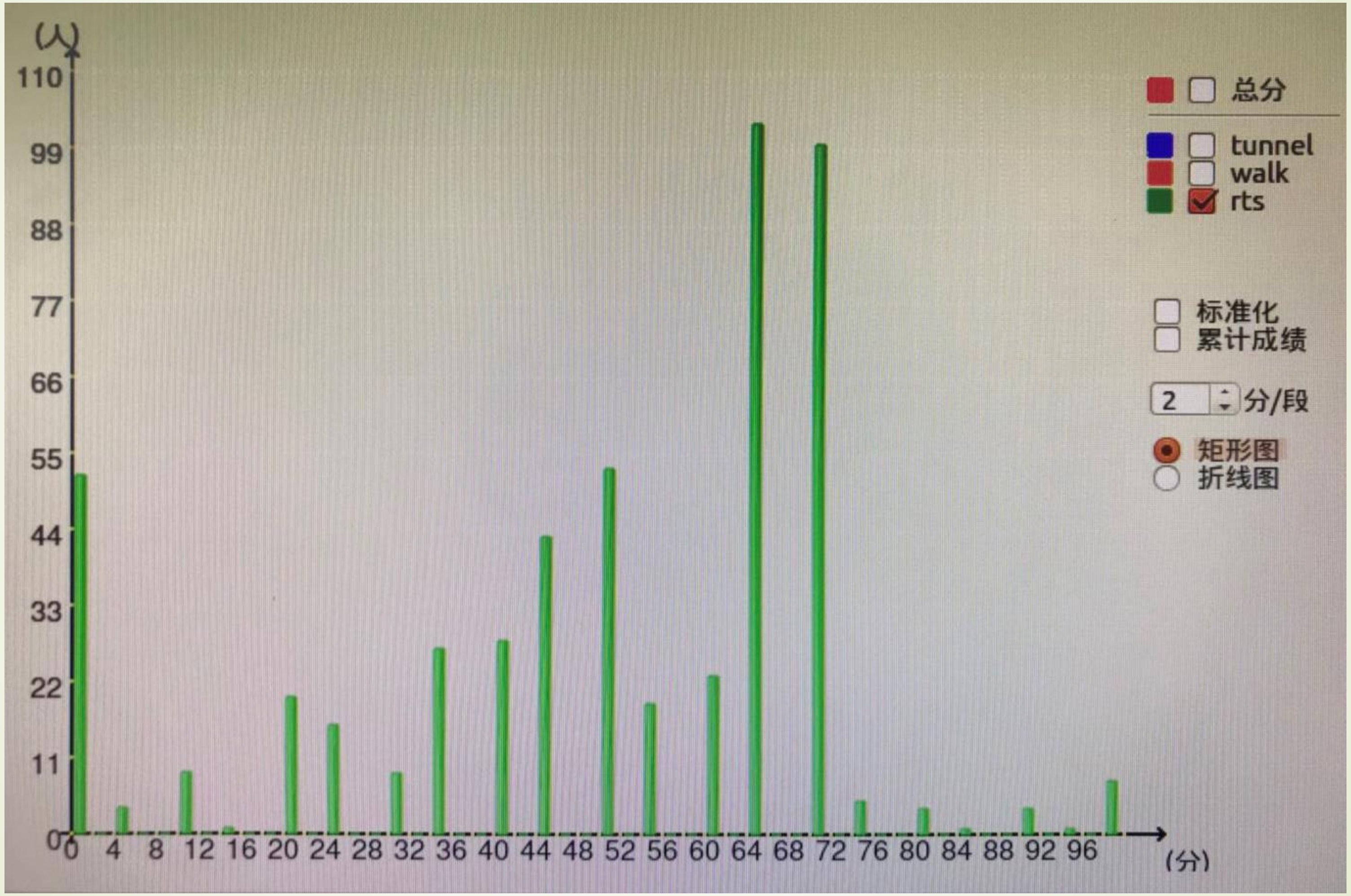
即时战略
120
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49

得分情况 (集训队)

- **100 分：** 11 人
- **≥ 70 分：** 35 人
- **≥ 65 分：** 45 人
- **≥ 45 分：** 49 人

- **中位数：** 70 分
- **平均数：** 78.2 分

得分情况 (非集训队)



(人)
110
99
88
77
66
55
44
33
22
11
0
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84 88 92 96
(分)
总分
tunnel
walk
rts
标准化
累计成绩
2 分/段
矩形图
折线图

得分情况 (非集训队)

- **100 分：** 8 人
- **70 分：** 约 100 人
- **65 分：** 约 100 人

自由讨(tu)论(cao)

解题思路 -1

- **放弃法：**
- 交互题太难不做

- **期望得分 0 分**
- 实际得分 0 分

解题思路 0

- **放弃法：**
- 交互题太难不做
- 交一个样例程序试试看

- **期望得分 不知道**
- **实际得分 5 分**

- 这是因为样例程序中有一行 explore(1, 2)

解题思路 1

- **暴力法：**
- for i = 2 to n
- 从结点 1 开始通过调用 explore 函数一路走到结点 i

- 我们知道这个暴力的操作次数上界是 $\frac{1}{2}n^2$
- 所以能通过**第一部分的全部测试点**
- **期望得分 20 分**

- **实际得分：**
- 如果写了类似 if n ≤ 100 then solve() 的语句： 20分
- 如果没有写 if： 35分……（为什么呢？）

解题思路 1

- **暴力法：**
- for i = 2 to n
- 从结点 1 开始通过调用 explore 函数一路走到结点 i

- 注意这个算法的实际操作次数是“每个结点的深度之和”
- 观察第二部分的数据范围，可以发现二叉树的深度是 $\log n$ 的
- 在第二部分的测试点中 $n \log n \le T$
- 所以也能通过**第二部分的全部测试点**
- **期望得分 35 分**
- **实际得分 35 分**

解题思路 2

- **暴力法：**
- for i = 2 to n
- 如果结点 i 已经被访问过了则跳过 (continue)
- 否则：从结点 1 开始通过调用 explore 函数一路走到结点 i

- 这个算法和上一个算法的**期望得分相同，35分**
- **实际得分 45 分……（为啥呢？）**

解题思路 2

- 观察第三部分的数据范围，一条链
- 每次在这条链上找一个未知的结点，并把 1 号结点到它的路径标为已知
- 可以证明，当未知的结点是随机选取的时候，标记的次数的期望为 $2\ln n + 1$ 左右
- $E(n) = 1 + \frac{1}{n}\sum_{i=0}^{n-1} E(i), E(0) = 0$
- $\Rightarrow nE(n) + 1 + E(n) = (n+1)E(n+1)$
- $\Rightarrow E(n+1) - E(n) = \frac{1}{n+1}, E(n) = \sum_{i=1}^{n} \frac{1}{i} \approx \ln n + 0.5$
- 因此操作次数的期望为 $2n\ln n$ 级别
- 能通过第三部分的第一个测试点 (n = 1000, T = 50000)

没有随机排列结点没关系，因为数据就是随机的！

解题思路 3

- **一条链：**
- 维护已知部分的左右端点
- 每次在这条链上找一个未知的结点
- 在 1 号结点询问这个结点是左边还是右边
- 然后从相应的端点开始标记路径

- 每个结点只被标记一次，询问次数等于标记路径的次数，即 $2\ln n$
- **第三部分的期望得分：27 分**
- **第三部分的实际得分：25 分 或 30 分**

今天你中奖了吗（大雾）

解题思路 4

- **一条链：**
- 维护已知部分的左右端点
- 每次在这条链上找一个未知的结点
- 直接猜测左边，从左边端点开始标记
- 如果猜错了就从右边继续标记

- 每个结点只被标记一次，猜错的次数等于标记路径的次数的一半，即 $\ln n$
- **第三部分的期望得分：30 分**
- **第三部分的实际得分：25 分 (144/100000 的概率) 或 30 分**

今天你中奖了吗（大雾）

解题思路 5

- **一棵树：**
- 问题转化：每次找一个未知的结点，要询问出树上离它最近的结点
- 树的点分治：在树上找到一个【删掉它之后最大子树最小】的结点，称作**重心**
- 在重心上调用 explore 函数，确定往哪棵子树走（这叫**树的点分治**）
- 重心的性质：删去它之后最大子树的大小不超过原树的一半
- 所以对于每个未知结点，询问时需要的操作次数为 $\log n$ 次
- 但时间复杂度是 $O(n^2)$ （每次询问 $T(n) = T\left(\frac{n}{2}\right) + O(n) = O(n)$）

- **第四部分期望得分：10分**
- **第四部分实际得分：10分**

解题思路 6

- **树上数据结构：**
- 我们有很多同时支持【在树上加一个叶结点】和【对树进行分治】的数据结构
- **动态点分治：**将点分治的树形结构记录下来，用替罪羊树/treap的思想进行重构
- 询问的时候顺着点分治的结构一层一层询问即可
- 能保证时间复杂度为 $O\left(n\log^2 n\right)$，操作的次数为 $O(n\log n)$
- **Link/cut trees：**一种基于 splay 的动态树链剖分
- 询问的时候顺着 LCT 的树形结构沿着 splay 往下走，走到底之后 access 一次
- 能保证时间复杂度和操作次数都是 $O(n\log n)$ 级别

- **第四部分期望得分：35分**
- **第四部分实际得分：35分**

感谢

- 感谢 CCF 给我此次命题和交流的机会
- 感谢大家的认真聆听