Shopping 解説

担当 : 高谷 悠太

問題概要

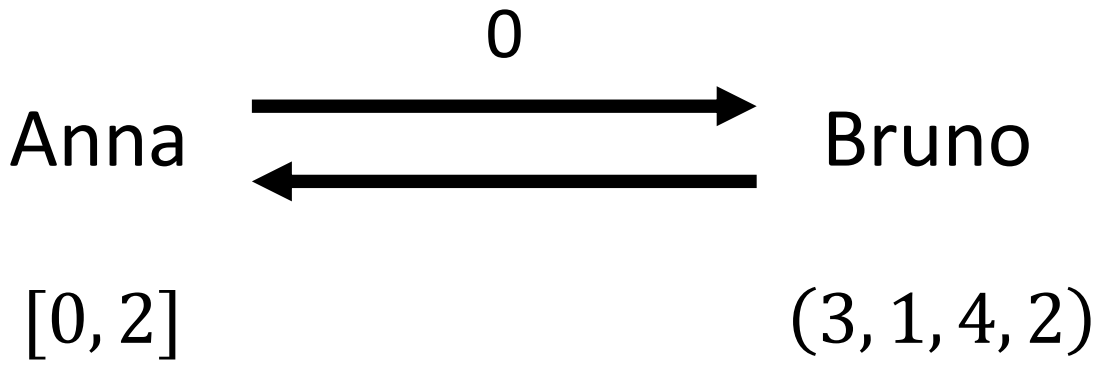
- N 個の商品がある.
- Anna は L 番目から R 番目の商品のうち，最も安い商品を購入したい.
- 各商品の値段を直接知ることはできないため，値段を把握している Bruno と通信を行う.

やり取りの例

$[0, 2]$ $(3, 1, 4, 2)$

$L = 0$ と伝えたい

やり取りの例



$L = 0$ と伝えたい

やり取りの例

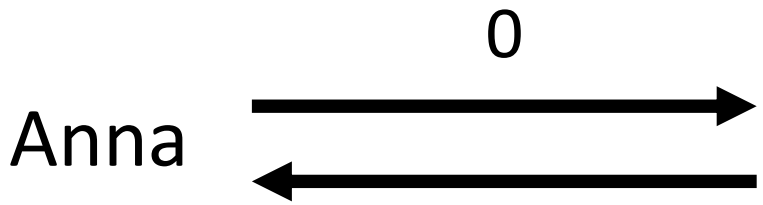
$[0, 2]$

$(3, 1, 4, 2)$

$L = 0$ と伝えたい

1^{st} bit : 0

やり取りの例



$[0, 2]$

$(3, 1, 4, 2)$

$L = 0$ と伝えたい

1^{st} bit : 0

やり取りの例

$[0, 2]$

$(3, 1, 4, 2)$

$L = 0$ と伝えたい

1st bit : 0
2nd bit : 0

やり取りの例

$[0, 2]$

$(3, 1, 4, 2)$

$L = 0$ と伝えたい

1st bit : 0
2nd bit : 0

$L = 0$ と気づく

注意

- 厳密な通信方法を把握するのはとても大切.

- 例えば，双方向の通信といっても
  - 0 と 1 からなる**文字列**を送りあう
  - 0 と 1 の**文字**を送りあう

  では情報の送り方が大きく異なる.

注意

- 今回の通信法は後者.

- つまり，相手が 1 ステップの通信を終えるまで待機し，その後自分が次のステップの通信を行う，などは不可能.

注意

- 双方向通信はあまり見ない形式だが,
  クエリの扱い方だけでなく，クエリ内容まで参加者側に自由度がある通信課題

  といえる.

- クエリの効率的な扱い方を考える一般的な通信課題と違い，どのようなクエリを相手に質問するかが重要となる.

過去の例

- ふたつの交通機関（JOI 2018/2019 春合宿）

準備

- 以下のように定式化しておく.
- Anna は区間 $[L, R]$ を， Bruno は数列 P を持っている.
- Anna は通信を行うことで， $[L, R]$ のうち数列 P において最小値をとる index を求めたい． どの程度の通信量で可能か.

小課題１：$N \leq 1000$

- Bruno が P のすべての値を伝える.
  - 一つの値を伝えるには $\log_2 N$ bit 必要.

- 以上より，以下の通信量で可能.
  - Anna : 0 bit
  - Bruno : $N \log_2 N$ bits

小課題２：$N \leq 10{,}000$

- Anna は L の値を伝える.
  - これは $\log_2 N$ bits 必要.

- Bruno は L 以上の各整数 X について,
  数列 P が $[L, X]$ において X で最小値をとるかどうか伝える.
  - これは N bits 必要.

小課題２：$N \leq 10{,}000$

- $[L, X]$ において X で最小値をとるような X のうち，R 以下最大のものが求める index.

- 以上より，以下の通信量で可能.
  - Anna : $\log_2 N$ bits
  - Bruno : N bits

小課題３：$N \leq 1{,}000{,}000$

- ここでは三つの解法を紹介する.

1. バケット法（32 点）
2. Cartesian Tree 上での重心分解（80 点）
3. 二分探索（100 点）

小課題３：バケット法

- バケットサイズ B を固定する.
  - Anna と Bruno の間でこの値を共有しておく.
- 数列 P を N/B 個の長さ B の数列に分割する.
- Anna は L, R が属するバケットの番号を送る.
  - 合計で $2\log_2\left(\frac{N}{B}\right) - 1$ bits 必要.

小課題３：バケット法

- L, R が属するバケットの番号を l, r とする.

- このとき，求める index は
  - バケット l に属す
  - バケット $l+1, \ldots, r-1$ で最小値をとる index
  - バケット r に属す

  のいずれかを満たす.

小課題３：バケット法

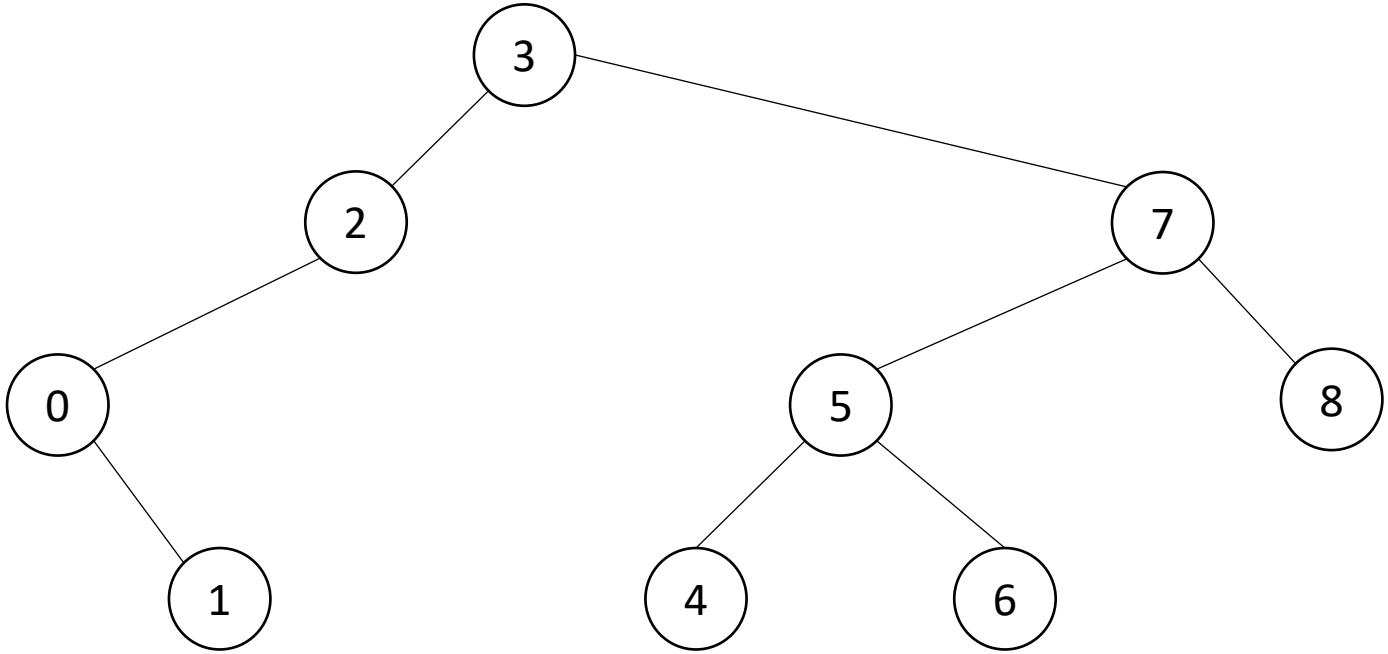
- Bruno はまずバケット $l+1, \ldots, r-1$ で最小値をとる index を送る.
  - これは $\log_2 N$ bits 必要.
- そして，$2B+1$ 個の候補についての Cartesian Tree の情報を送る.

Cartesian Tree

- 数列 Q に対する Cartesian Tree の定義 :

  1. 数列 Q の各 index を頂点とする根付き二分木.
  2. 最小値をとる index を根とする.
  3. 根の左にある部分木は，最小値をとる index より左の部分列に対する Cartesian Tree である.
  4. 根の右にある部分木は，最小値をとる index より右の部分列に対する Cartesian Tree である.

Cartesian Tree



5	9	3	1	7	4	6	2	8

Cartesian Tree の性質

- Cartesian Tree は最小値の情報は覚えていないが, 最小値をとる index は覚えている.

- $[L, R]$で最小値をとる index に対応する頂点は, 頂点 L と頂点 R の LCA である.

- よって，数列全体の情報は必要なく，対応する Cartesian Tree の情報を送れば十分である.

Cartesian Tree の送り方

- 根付き二分木の個数は Catalan 数で表せ，頂点数を M とすると高々 4^{M-1} 個である.

- 実際，Cartesian Tree の構成を考えると，$2M - 2$ bits で送ることが可能.

Cartesian Tree の構成

1. 空の stack S を用意し，Q の要素を先頭から見ていく.

2. 今見ている値より S の先頭の値が大きい場合, S の先頭を pop する.

3. 今見ている値より S の先頭の値が小さい場合, S の先頭に今見ている値を push し，次の要素に進む.

Cartesian Tree の~~構成~~ 送り方

1. 空の stack S を用意し，Q の要素を先頭から見ていく．

2. 今見ている値より S の先頭の値が大きい場合，S の先頭を pop する．このとき 0 を送る．

3. 今見ている値より S の先頭の値が小さい場合，S の先頭に今見ている値を push し，次の要素に進む．このとき 1 を送る．

小課題３：バケット法

- Anna は L, R が属するバケットの番号を送る.
  - 合計で $2\log_2\left(\frac{N}{B}\right) - 1$ bits 必要.

- Bruno はバケット $l+1, \ldots, r-1$ で最小値をとる index を送る.
  - これは $\log_2 N$ bits 必要.
- そして，$2B+1$ 個の候補についての Cartesian Tree の情報を送る.
  - これは $4B$ bits 必要.

小課題３：バケット法

- まとめると，以下で可能.
  - Anna : $2\log_2\left(\frac{N}{B}\right) - 1$ bits
  - Bruno : $4B + \log_2 N$ bits

- $B = 1400$ 程度にすると 32 点が得られる.

小課題３：重心分解

- （再掲）
  $[L, R]$ で最小値をとる index に対応する頂点は，頂点 L と頂点 R の LCA である．

- つまり，問題は以下のように言い換えられる．
  - Anna は二頂点 L, R を，Bruno は二分木を持っている．
  - 通信を行うことで，Anna は Bruno の木における LCA を求めたい．

小課題３：重心分解

- ある頂点 v を根とする部分木に対応する区間を $[l, r]$ とする．Bruno はまず l, r を送る．

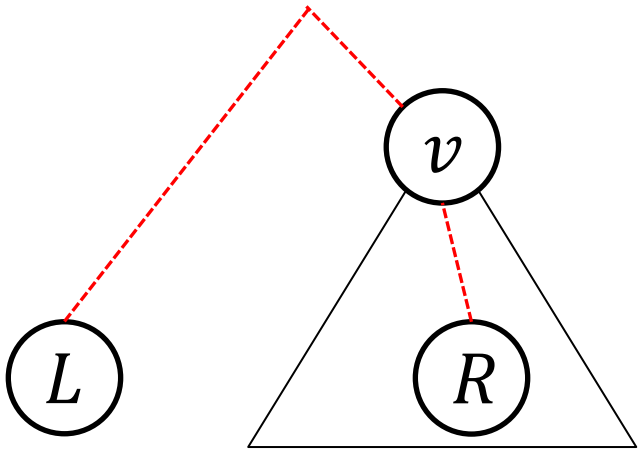
- Anna は以下のようにして 01 を返す．
  - $[L, R]$ が $[l, r]$ に含まれていれば 1 を返す．
  - そうでない場合 0 を返す．

- Bruno は Anna の返答から何が分かるか？

小課題３：重心分解

- $[L, R]$ が $[l, r]$ に含まれていれば，
  当然 L と R の LCA は $[l, r]$ に含まれる．

- そうでない場合，$[l, r]$が頂点 v を根とする部分木に対応する区間であることから，LCA が $[l, r]$ に含まれることはないことが分かる．



小課題３：重心分解

- Bruno は Anna からの返答により，求める答えが $[l, r]$ に含まれるかどうか判定できる.

- できるだけ $[l, r]$ の長さを $N/2$ に近づければ，毎回頂点数が半減するのではないか？
  - 18 回繰り返せば最後 4 頂点しか残らず AC する？

小課題３：重心分解

- Bruno は Anna からの返答により，求める答えが $[l, r]$ に含まれるかどうか判定できる.

- できるだけ $[l, r]$ の長さを $N/2$ に近づければ，毎回頂点数が半減するのではないか？
  - 18 回繰り返せば最後 4 頂点しか残らず AC する？
  - 実はそんなに甘くない.

小課題３：重心分解 (algorithm)

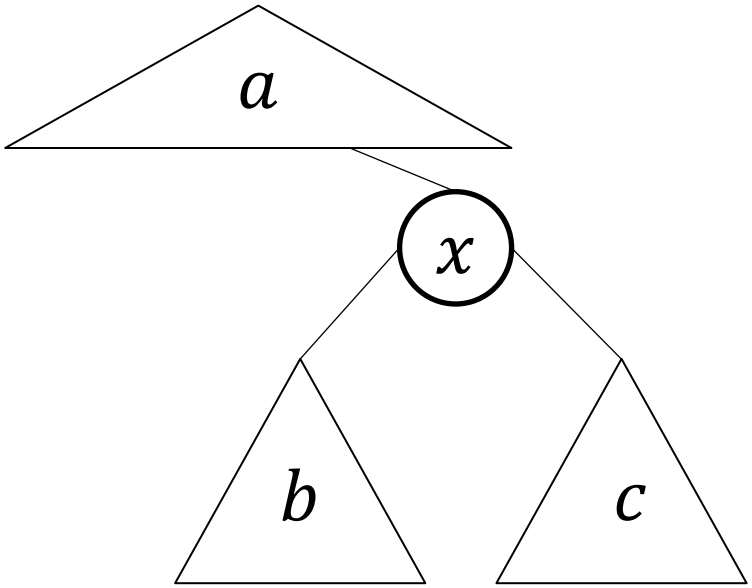
- まず以下を 18 回繰り返す.
  - Bruno は $[l, r]$ を長さが $V/2$ に最も近くなるようにとり，それを送る． V : 残り頂点数
    - これは $2\log_2 V - 1$ bits で可能.
  - Anna は$[L, R]$が $[l, r]$ に含まれるかどうか判定する. そして，不必要な頂点を削除する.

- 最後に残った頂点について， Bruno は Cartesian Tree を送る.
  - これは $2V - 2$ bits で可能． V : 残り頂点数

小課題３：重心分解 (estimation)

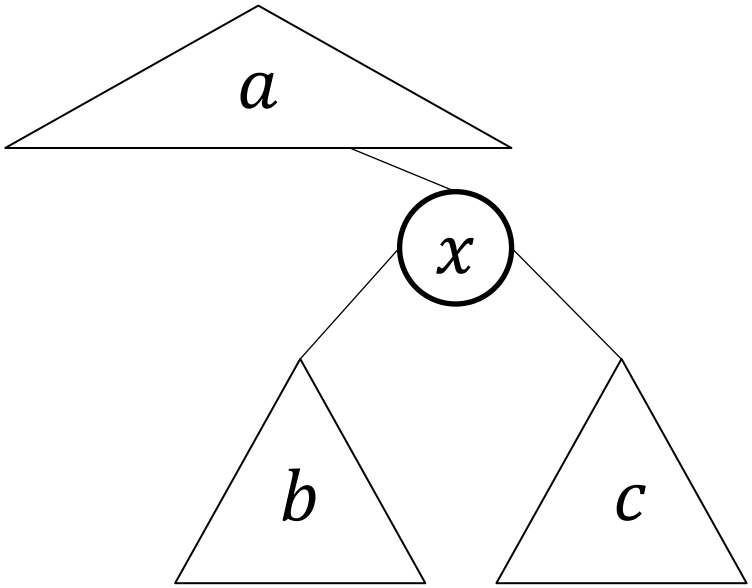
- １回でどの程度頂点数が減るか見積もる.

- x を重心とし，その周りの部分木の大きさを a, b, c とする.



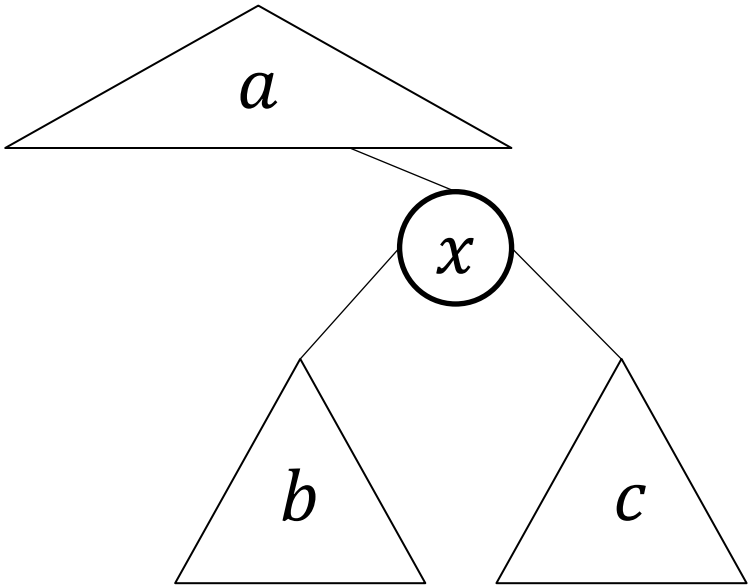
小課題３：重心分解 (estimation)

- x を重心なので，$a, b, c \leq V/2$ である．
  - 特に，次数が 2 以下であれば 1 回で半減する．
    以下，次数は 3 とする．



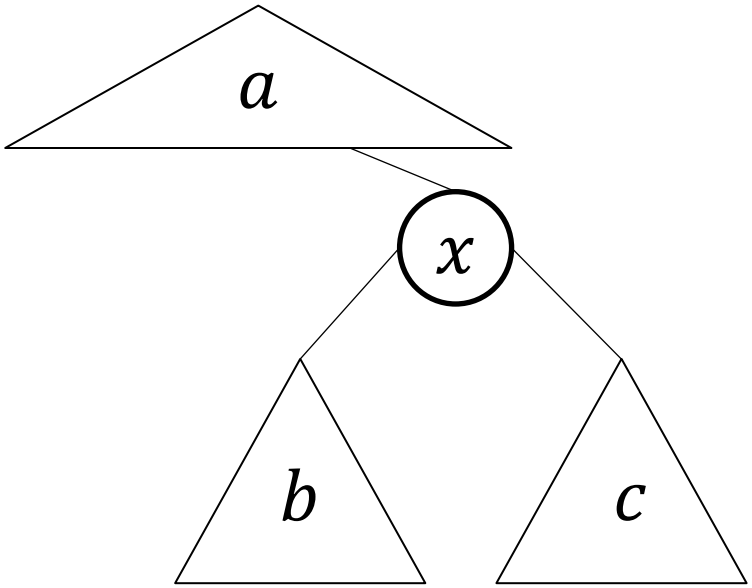
小課題３：重心分解 (estimation)

- 対称性より，$a \geq b \geq c$ とする．
- 1 回行うと，サイズは高々 $b + c$ になり，
  2 回行うと，サイズは高々 b になる．



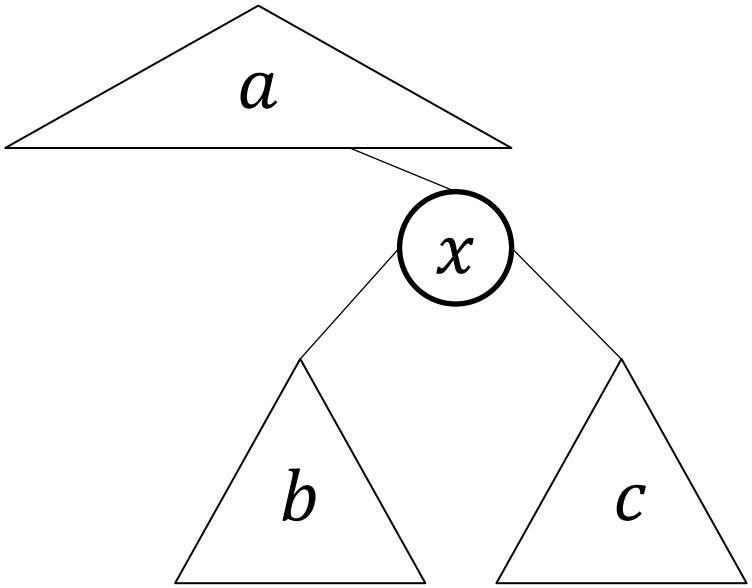
小課題３：重心分解 (estimation)

- α を以下を満たす実数とすれば，（最後の 1 回外）1 回あたり頂点数は α 倍となる．
  - $b + c \leq \alpha V$ もしくは $b \leq \alpha^2 V$



小課題３：重心分解 (estimation)

- $b + c > \alpha V$ かつ $b > \alpha^2 V$ となると
  $$V > a + \alpha V \geq b + \alpha V > (\alpha^2 + \alpha)V$$
  なので．$\alpha^2 + \alpha = 1$ なる α は条件を満たす．
  - 実はαは黄金比の逆数.



小課題３：重心分解 (estimation)

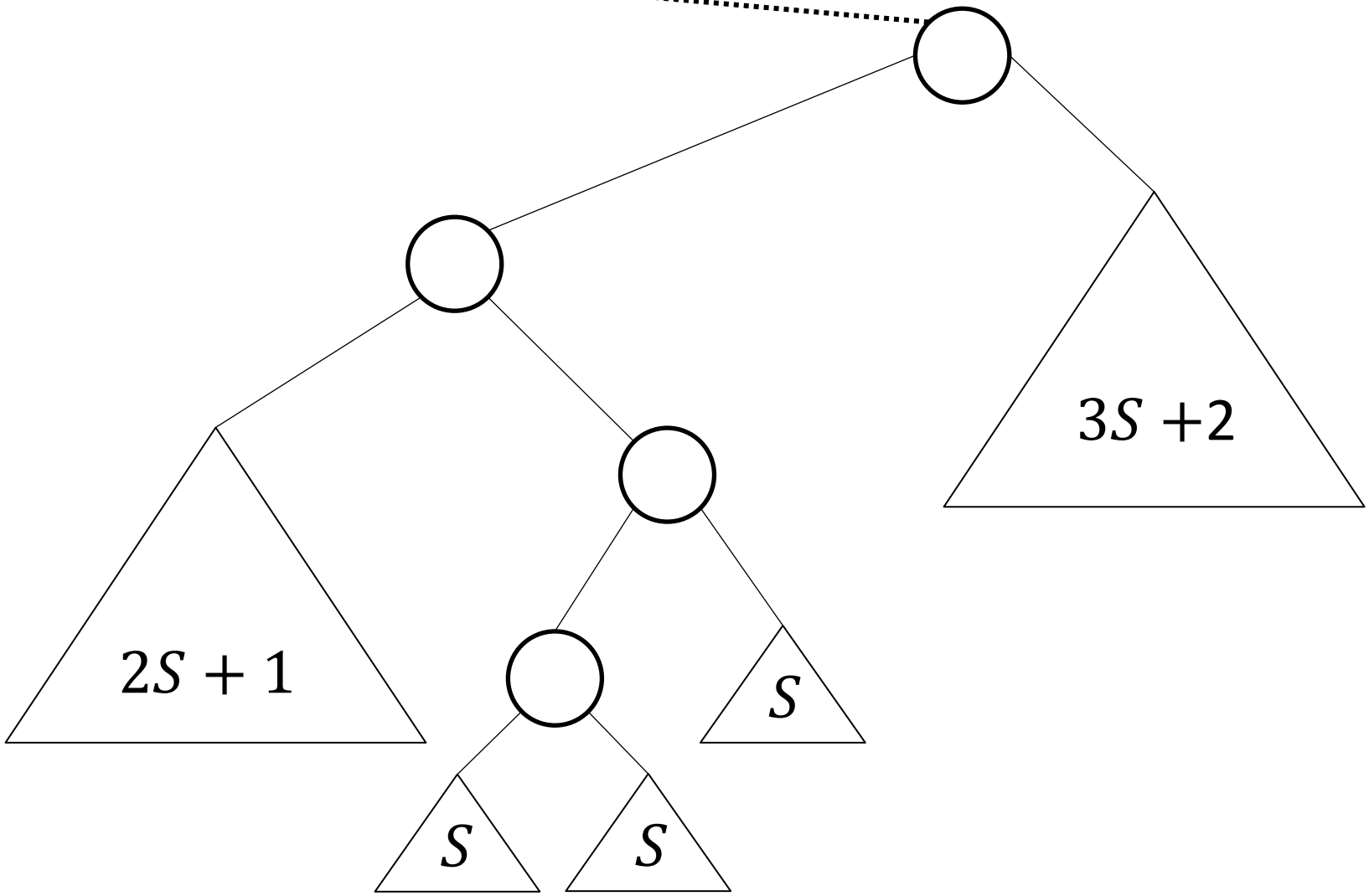
- 最後の 1 回はサイズは高々 $b+c$ にしかならないので，効率は $2/3$ である．

- よって，最後に残る頂点数は高々

$$1{,}000{,}000 \times \alpha^{17} \times \frac{2}{3} \approx 186$$

となる．

小課題３：重心分解 (worst case)



小課題３：重心分解 (estimation)

- 最初の 18 回の通信量 :
  - $2\log_2 V - 1$ の値は 1 回目は 39，18 回目は 16 程度なので，合計 $(39 + 16) \times 18 \div 2 \approx 500$ bits 程度
- 最後の Cartesian Tree の通信量 :
  - $2 \times 186 - 2 = 370$ bits 程度
- 合計 870 bits で可能． 80 点程度.

小課題３：二分探索

- 次のような場合を考える.

  Bruno が $[L, R]$に含まれる index X を一つ知っている.

- この場合，1 回で要素数を半減させることができる.
  - Cartesian Tree における X の親について二分探索を行えばよいが，応用上別の方法を示す.

小課題３：二分探索

- $[X, X]$ から始め，以下のように区間を伸ばし，長さ 1 から N までの区間を得る．

  1. 長さ $n+1$ の区間は，長さ n の区間を左右いずれかに長さ 1 伸ばした区間である．

  2. ただし，追加する値が**大きい**方向に区間を伸ばす．
     - つまり，区間内の最小値ができるだけ大きくなる方向に区間を伸ばしていく．

- 長さ n の区間を I_n とする．

小課題３：二分探索

- $[L, R]$ が I_n に含まれていたとき，当然求める index は I_nに含まれる.

- 一方，I_nに含まれない場合は，求める index は以下のいずれかを満たす.
  1. I_n に含まれない.
  2. I_n で最小値をとる index と等しい.

小課題３：二分探索

- $[L, R]$ が I_l に含まれず，I_r に含まれる，という l, r が求まると，後は以下を送ればよい.

  1. I_l で最小値をとる index

  2. $I_r \setminus I_l$ および 先の値を含めた数列の Cartesian Tree

- これは $\log_2 N + 2(r - l)$ bits 必要.

小課題３：二分探索

- 二分探索を k 回行えば $r - l = N/2^k$ となる.
- 二分探索 1 回には，Anna は 1 bit，Bruno は $\log_2 V - 1$ bits (V : 残り要素数) かかる.
- よって，以下の通信量で可能.
  - Anna : k bits
  - Bruno : $\frac{k(2\log_2 N - k - 1)}{2} + N/2^{k-1}$ bits

小課題３：二分探索

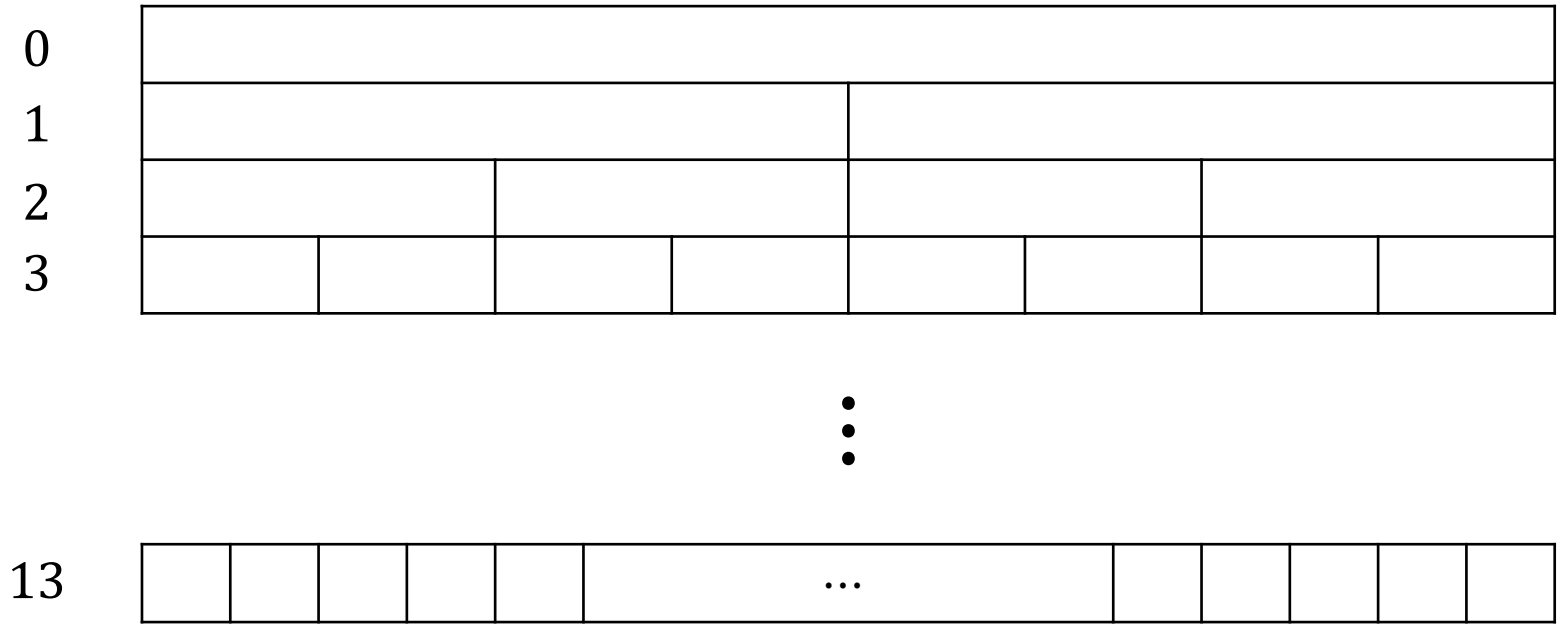
- 最初の仮定：

  Bruno が $[L, R]$に含まれる index X を一つ知っている.

- 実際には，このような仮定は成立しない.

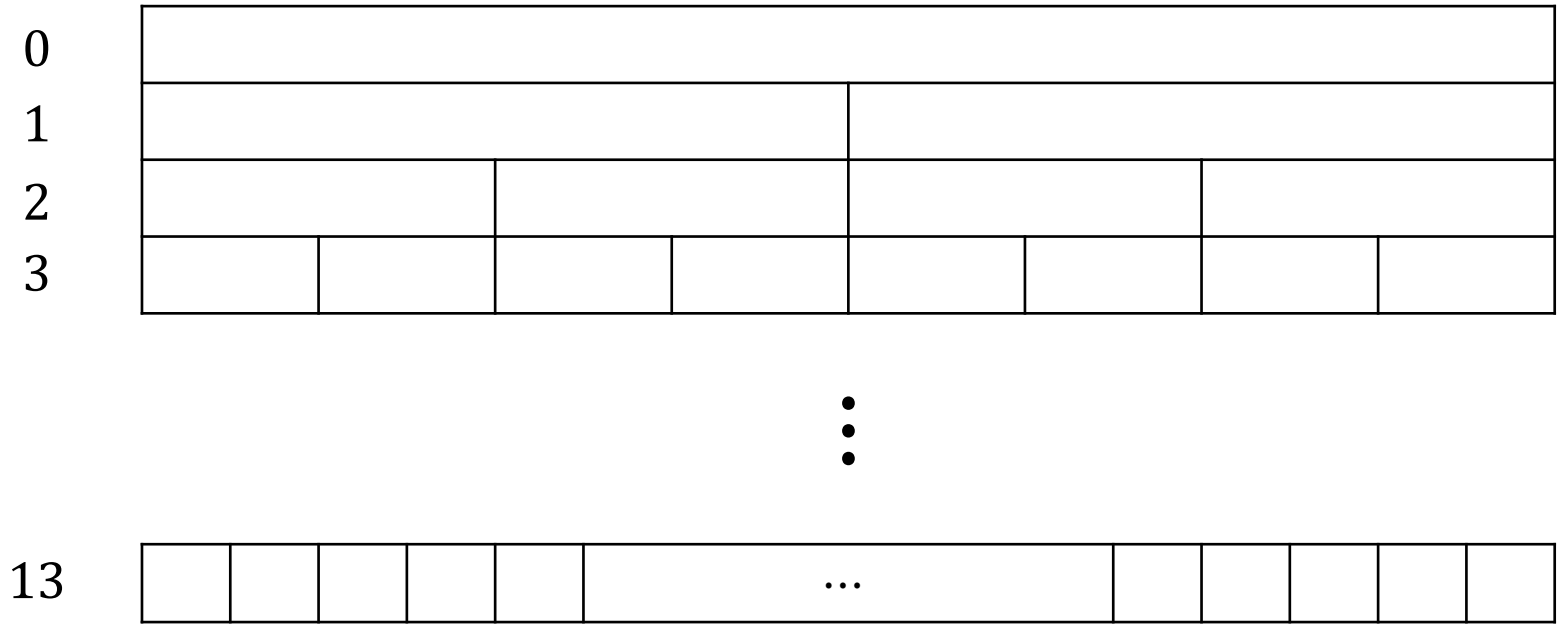
- 今度は，この情報を送ることを考える.

小課題３：二分探索



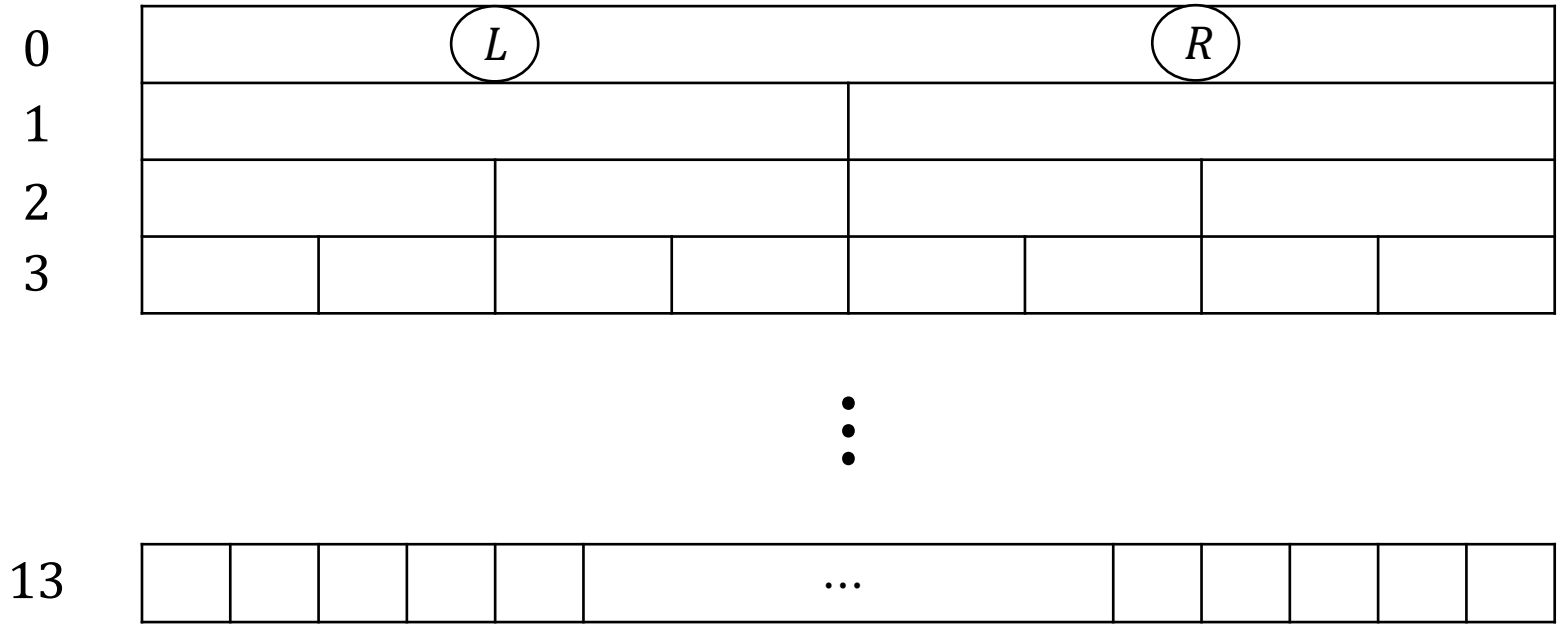
- segment tree （の葉を広くしたもの）を考える.

小課題３：二分探索



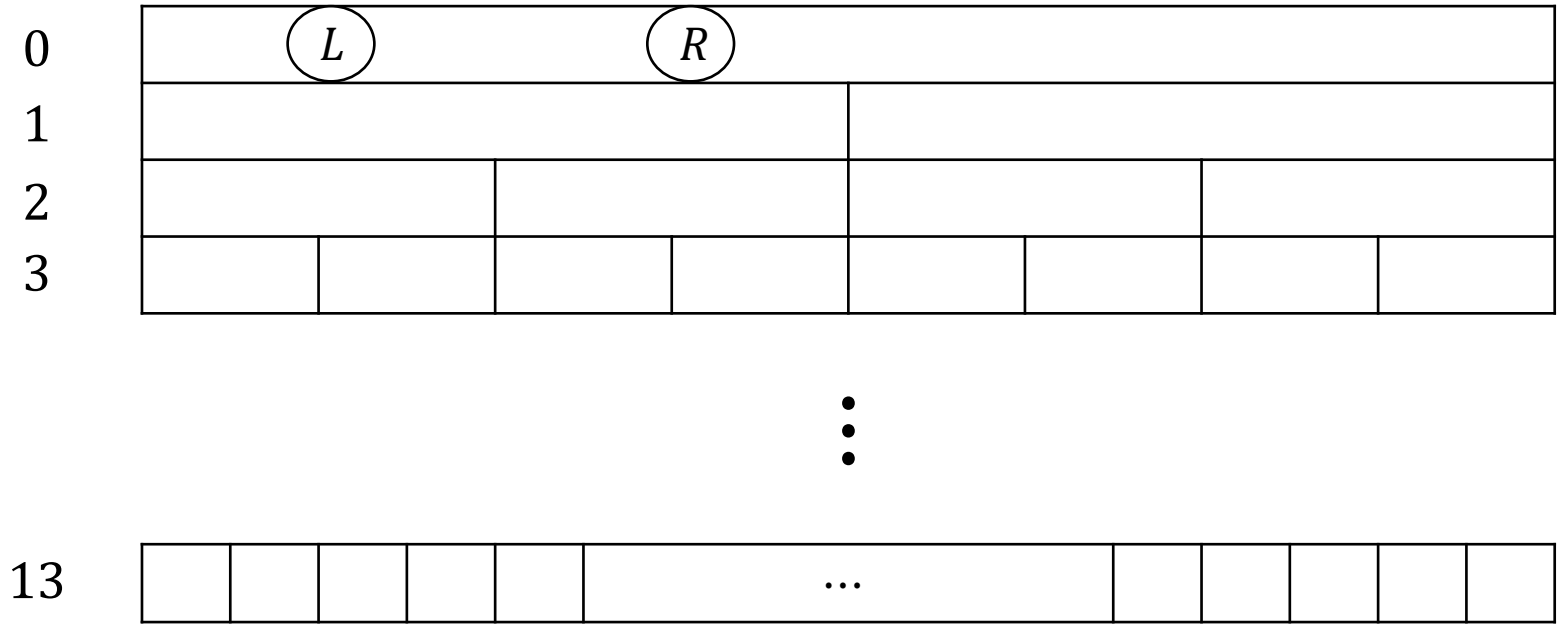
- $[L, R]$ が含まれる頂点のうち，最も深い頂点を送る．

小課題３：二分探索



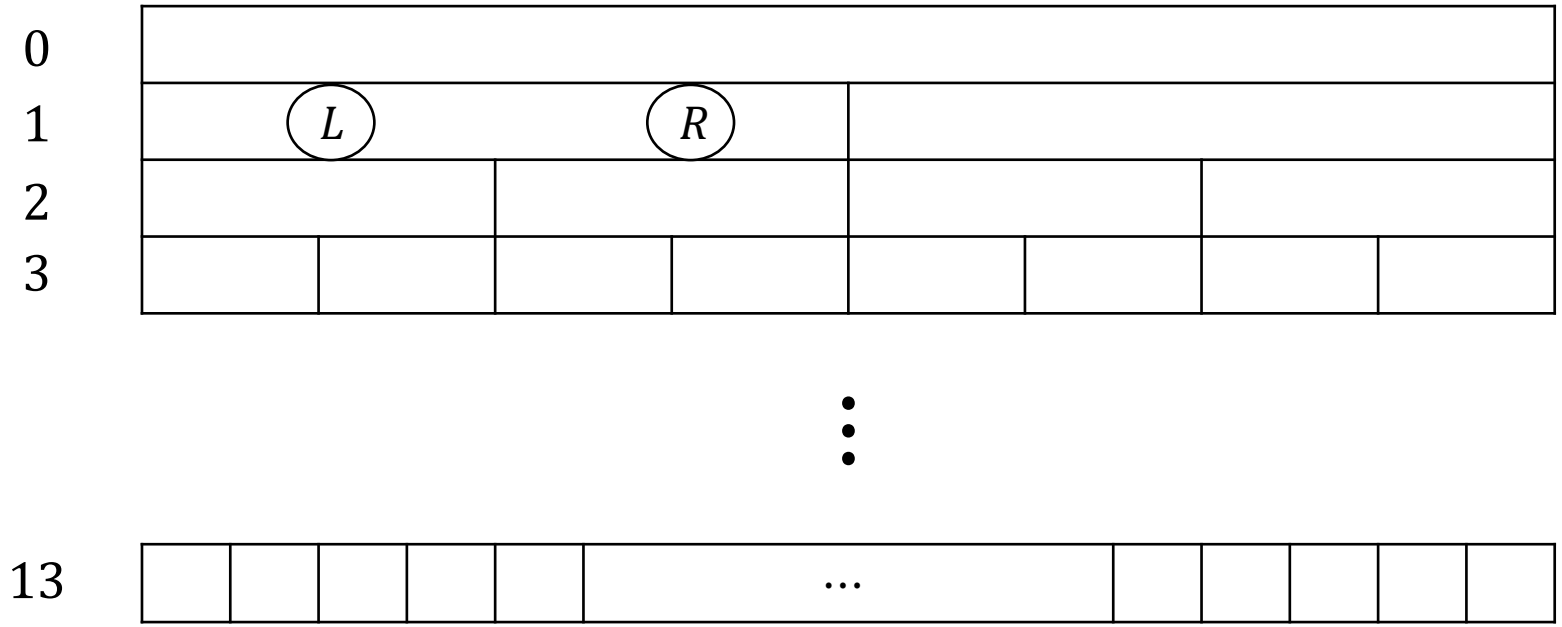
- $[L, R]$ が含まれる頂点のうち，最も深い頂点を送る．

小課題３：二分探索



- $[L, R]$ が含まれる頂点のうち，最も深い頂点を送る．

小課題３：二分探索



- $[L, R]$ を含む頂点のうち，最も深い頂点を送る．

小課題３：二分探索

- 深さが 1 増えるごとに，ありうる範囲が半減している．

- つまり，二分探索を行う回数を減らしても，その分深さが深ければ効率はほとんど同じである．

小課題３：二分探索 (algorithm)

- Anna は $[L, R]$ を含む頂点のうち，最も深い頂点の**深さ**を以下のようにして送る.
  - 深さが $0, 1$ のとき 3 bits 用いて,
    深さが 2 以上 13 以下のとき 4 bits 用いる.
    - $1/8 \times 2 + 1/16 \times 12 = 1$ より可能.
- それから，対応する頂点を送る.
  - その深さを d としたとき，d bits で可能.

小課題３：二分探索 (algorithm)

- Anna に残された通信量を k bits として，Bruno は k 回二分探索をし，Cartesian Tree を送る.
  - Anna : k bits
  - Bruno : $\frac{k(2\log_2 N - k - 2d - 1)}{2} + N/2^{k+d-1}$ bits

- $d = 2$ のときに最大となり,

$$12 \times 23 \div 2 + 1{,}000{,}000/2^{13} \approx 260$$

- 合計 300 bits に収まる．これで 100 点.

小課題３：二分探索 (in general)

- Anna は $[L, R]$ が含まれる頂点のうち，最も深い頂点の**深さ**を送る．
  - これは $\log \log N$ bits で可能．

- それから，対応する頂点を送る．
  - その深さを d としたとき，d bits で可能．

小課題３：二分探索 (in general)

- Bruno は $\log N - \log\log N - d$ 回二分探索を行い，最後に Cartesian Tree を送る.
  - Anna : $\log N - \log\log N - d$ bits
  - Bruno : $\frac{(\log N - \log\log N - d)(\log N + \log\log N - d - 1)}{2} + 2\log N$ bits

- よって，一般に以下の通信量で可能.
  - Anna : $\log N$ bits
  - Bruno : $\frac{1}{2}(\log N)^2 + O(\log N)$ bits

（$\log$ の底は 2 ）

得点分布