ICPC輪講第五回

理学部情報科学科4年　後藤哲志

- 教科書『算法芸術与信息学競賽』(劉汝佳, 黄
- 亮)の1.4.3節と1.4.4節
  - ハッシュと二分探索木
  - 線分木とトライ木
- 概要と例題の説明

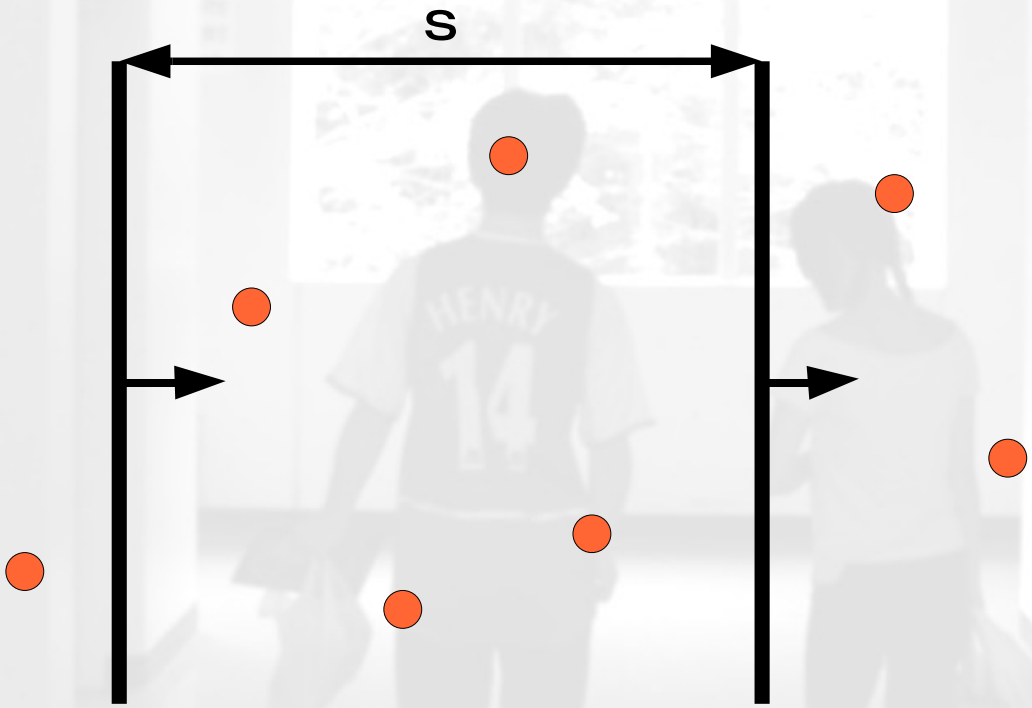
例題3 - Goldmine(1/15)

- 二次元平面上にn個(nは最大15,000)の金塊が埋まっている。幅s、高さwの矩形を平面上のどこに置けば金塊を最も多く矩形内（線上も含む）に入れられるか、その金塊の数の最大値を求めよ。

(Polish Olympiad in Informatics 2001 III)

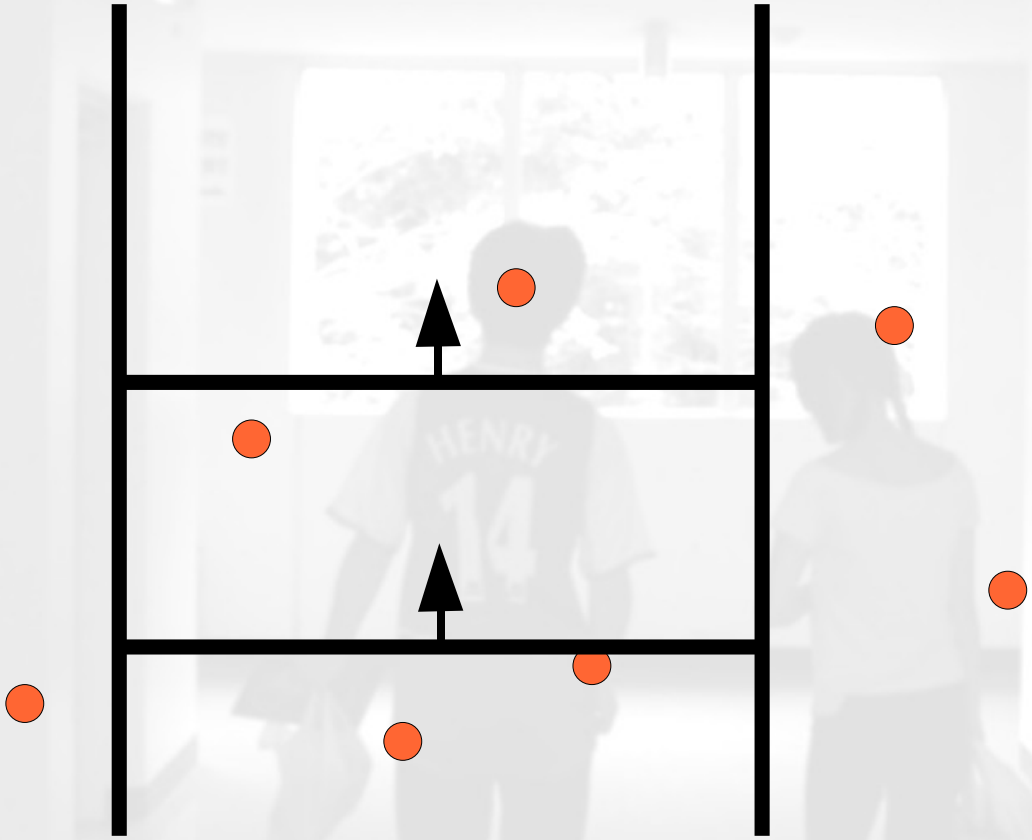
例題3 - Goldmine(2/15)

- 基本的には幅sの枠を左から右に移動させながら最大値を求めていく



例題3 - Goldmine(3/15)

- y軸方向に見てもっとも多く入るy座標を求める



例題3 - Goldmine(4/15)

- 幅を固定する都度に、その幅に入っている点について最大値を計算する
- 「特殊」なやり方で求める
  - 幅に入っているy座標の集合{y1,y2,...,yk}
  - ペアを2k個作成する
    - (y1,1),(y2,1),...,(yk,1)
    - (y1+w+1,-1),(y2+w+1,-1),...,(yk+w+1,-1)
  - この2k個のペアから最大値を求めることができる

例題3 - Goldmine(5/15)

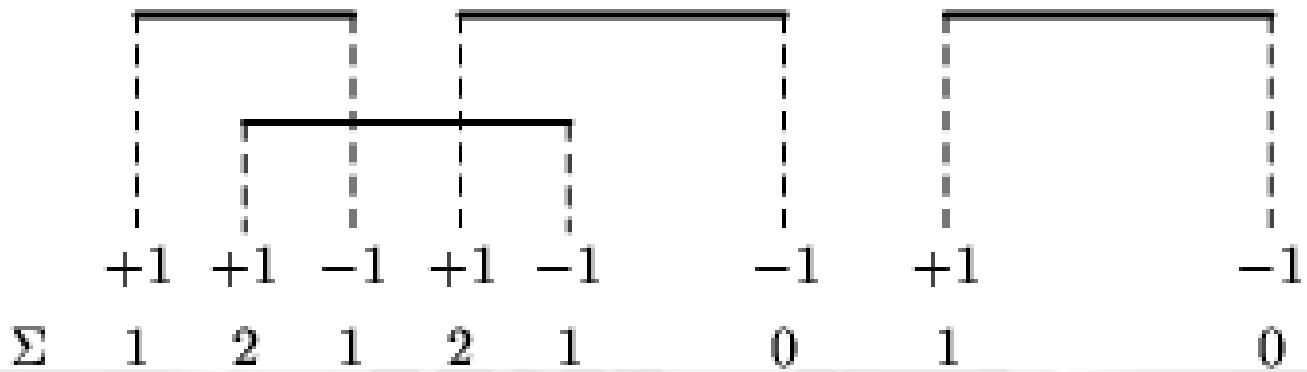
- 領域内の点のy座標を集め、(y,+1),(y+w+1,-1)と変換し、y座標が同じもの（表で数と書かれているもの）はまとめる
- 例：y={5,3,9,1,9},w=2
- ソートして表にしたものが次の通り
- 一番下の「連続和」とは左端から「数」を足していったもの

Y座標	1	3	4	5	6	8	9	12
Δ	1	1	-1	1	-1	-1	2	-2
連続和	1	2	1	2	1	0	2	0

例題3 - Goldmine(6/15)

- 下の表の「連続和」の最大値が求める値
  - 連続和は0未満にはならない

Rys. 2. Przykład zastosowania sum prefiksowych do wyznaczania "maksymalnego" przedziału.



Y座標	1	3	4	5	6	8	9	12
Δ	1	1	-1	1	-1	-1	2	-2
連続和	1	2	1	2	1	0	2	0

例題3 - Goldmine(7/15)

- しかし、ソートだとO(nlogn)かかる
  - 2n個の点が領域を出入りするので、全体ではO(n^2logn)となる
- 二分探索木(Binary Search Tree：BST)を用いることで全体でO(nlogn)で求めることができる

例題3 - Goldmine(8/15)

- 二分探索木
  - 左の子の木<=自分<=右の子の木
  - ノードの値はy座標
- ノードの追加はO(logn)
- 木の高さは平均lognなので、再帰で葉まで探索してもStack Overflowにはならない(と思う)
  - ソートされているデータが一度に来るとやばいかも
- ノードの削除をしないで追加のみで行う

例題3 - Goldmine(9/15)

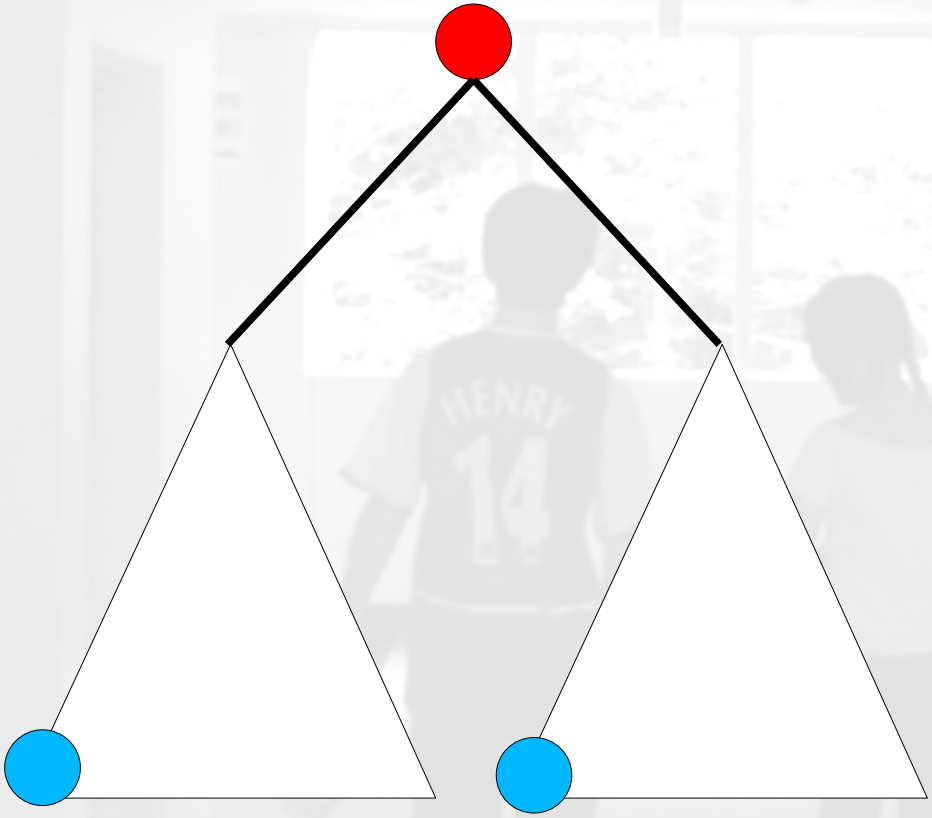
- 各ノードは、
  - SUM
    - 自分自身を根とする木の各ノードのdeltaの総和
  - MAXSUM
    - 自分自身を根とする木の連続和の最大値

  という二つの値を持つ
- この二つの値は木にノードを追加する度に正しく更新する必要がある

例題3 - Goldmine(10/15)

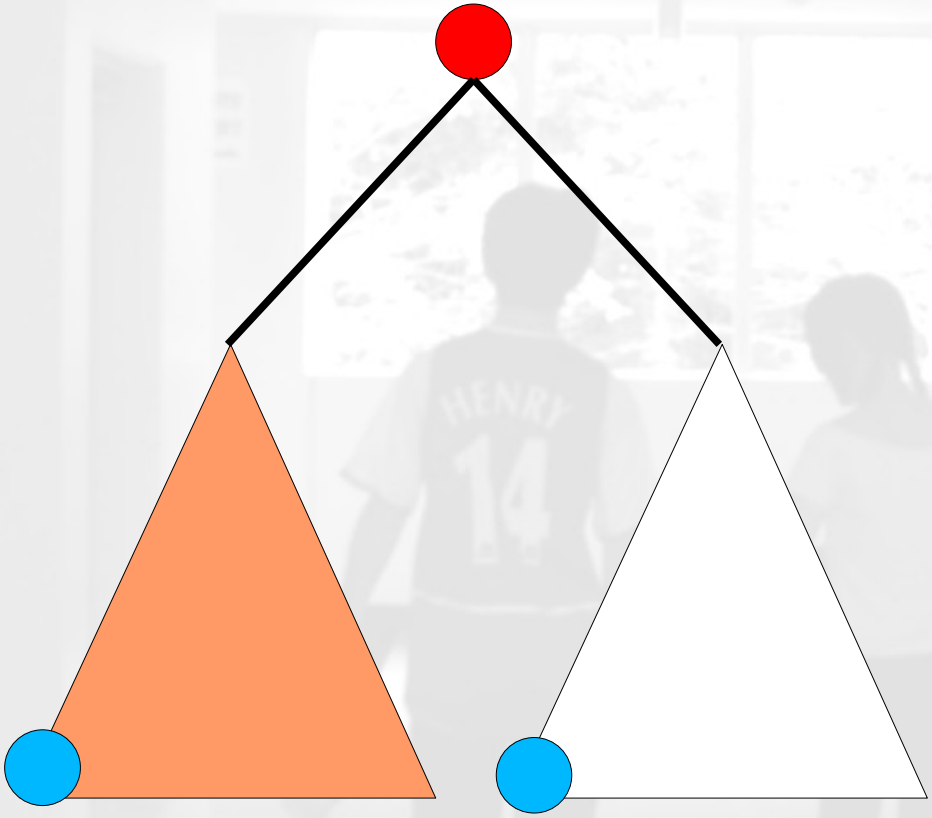
- MAXSUMの求め方
  - 葉ならMAXSUM = SUM
  - 枝なら次の三つのうちの最大値
    - 左の子のMAXSUM
    - SUM-右の子のSUM = 左の子のSUM+delta
    - 左の子のSUM+delta+右の子のMAXSUM

例題3 - Goldmine(11/15)



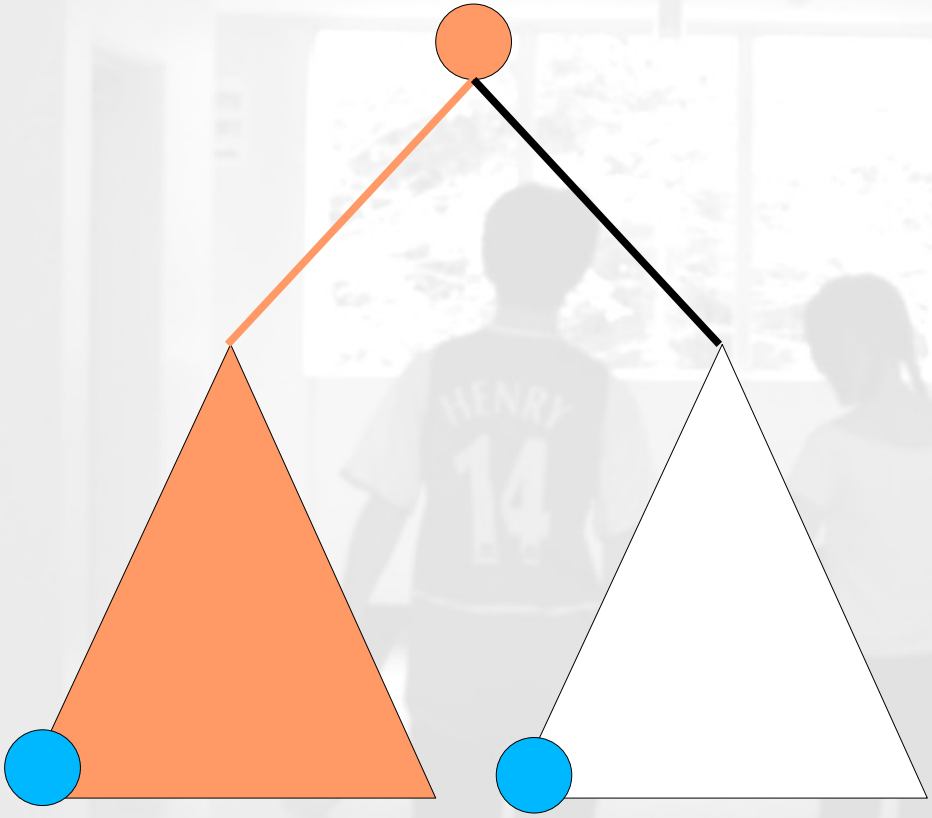
例題3 - Goldmine(12/15)

- ケース1:
  - 左の子の木の連続和の最大値



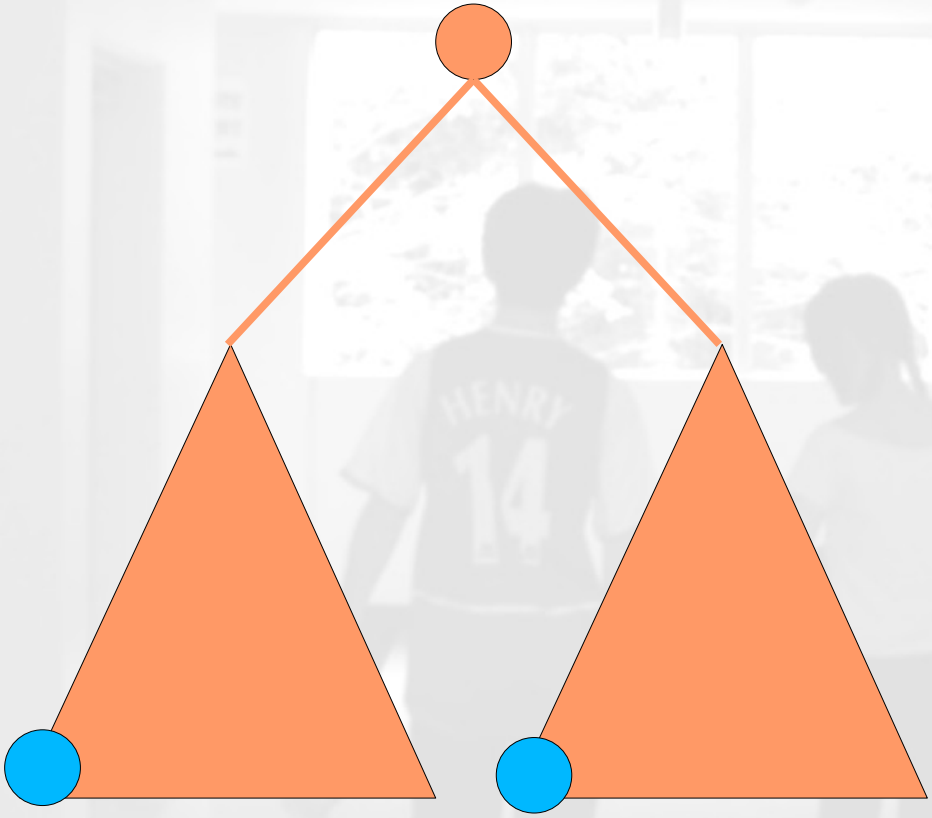
例題3 - Goldmine(13/15)

- ケース2:
  - 左の子の木の連続和+delta



例題3 - Goldmine(14/15)

- ケース2:
  - 全体の木の最大値の和



例題3 - Goldmine(15/15)

- 全てのノードを追加した後、rootのMAXSUMが最大値となる
- 点が領域から出たときの削除はdeltaの符号を逆にしたものを追加することで表現できる

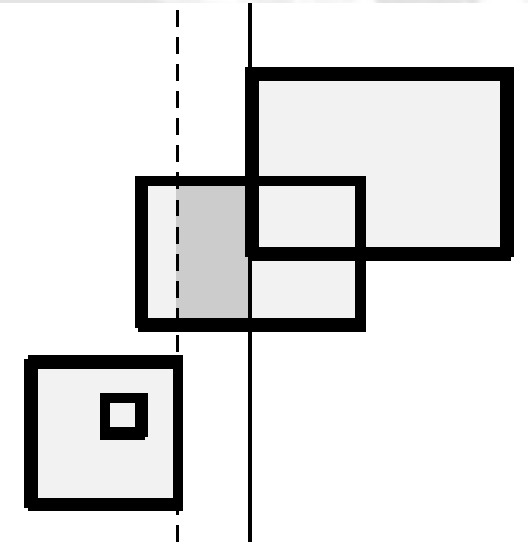
例題4 – Mars map(1/10)

- 二次元平面(0<=x,y<=30,000)上にn個(nは最大10,000個)の矩形が与えられる。このとき線で囲まれた面積の総和を求めよ。

(Baltic Olympiad in Informatics 2001)

例題4 – Mars map(2/10)

- 方法3：スキャンライン
  - x値を左から右に動かしながら、y方向にどれだけ囲まれた中にいるかを調べる。
  - xが変化するとy軸と交わっている矩形が増えたり減ったりする



例題4 – Mars map(3/10)

- 解法
  - 区間木(interval tree)を用いる
    - 二分木の一種
  - 各ノードvはある区間I(v)を表現する
    - 根は全区間([0, n])を表現する
  - あるノードvとその子w1,w2について、
    I(v)=[v1, v2]とすると、子w1,w2には、
    I(w1)=[v1, (v1+v2)/2]
    I(w2)=[(v1+v2)/2, v2]
    が割り当てられる

例題4 – Mars map(4/10)

- 各ノードは自分を根とするサブツリーがもつ区間の共通部分の長さ(length)をもつ
  - 根のlengthをどんどん足していく
- 木に対して区間の追加、削除を行い、そのときにlengthも更新を行う
  - あらかじめ矩形をx軸方向に見てソートしておく

例題4 – Mars map(5/10)

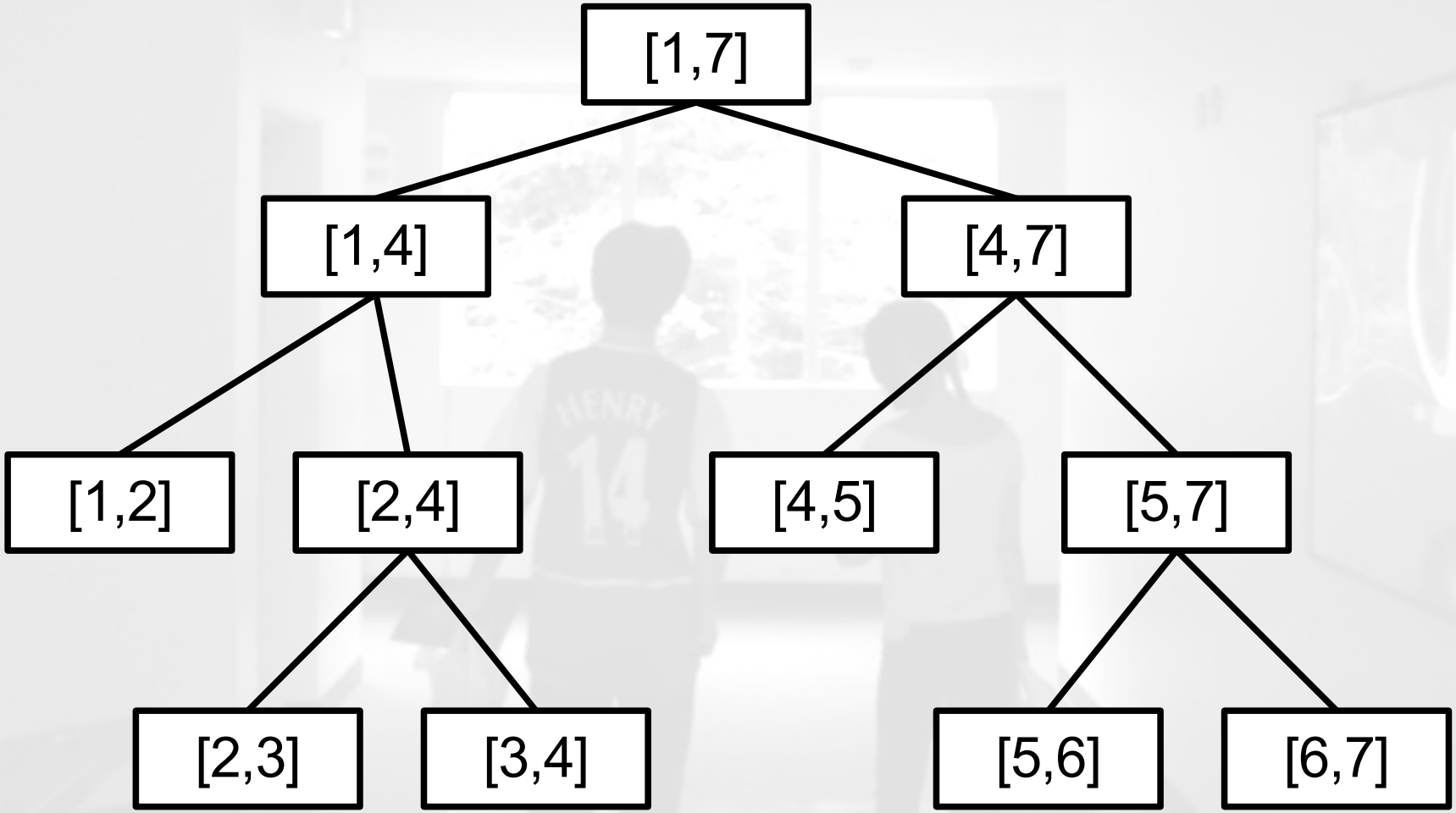
- 準備
  - 全ての矩形の両端点(y座標)をソートする
  - 以降、区間[a,b]とは、a番目の端点を左端、b番目の端点を右端とする区間を意味する
- あとは各ノードのlengthを正しく更新しながら範囲の追加、削除を行えばよい

例題4 – Mars map(6/10)

- 区間の追加、削除
  - 各ノードには何回範囲を割り当てられたかを表す変数がある（初期値は全て0）
    - 追加のときは変数に1を足し、削除の時は1を引く
    - 0から非0、非0から0の時は自身のlengthの更新も必要

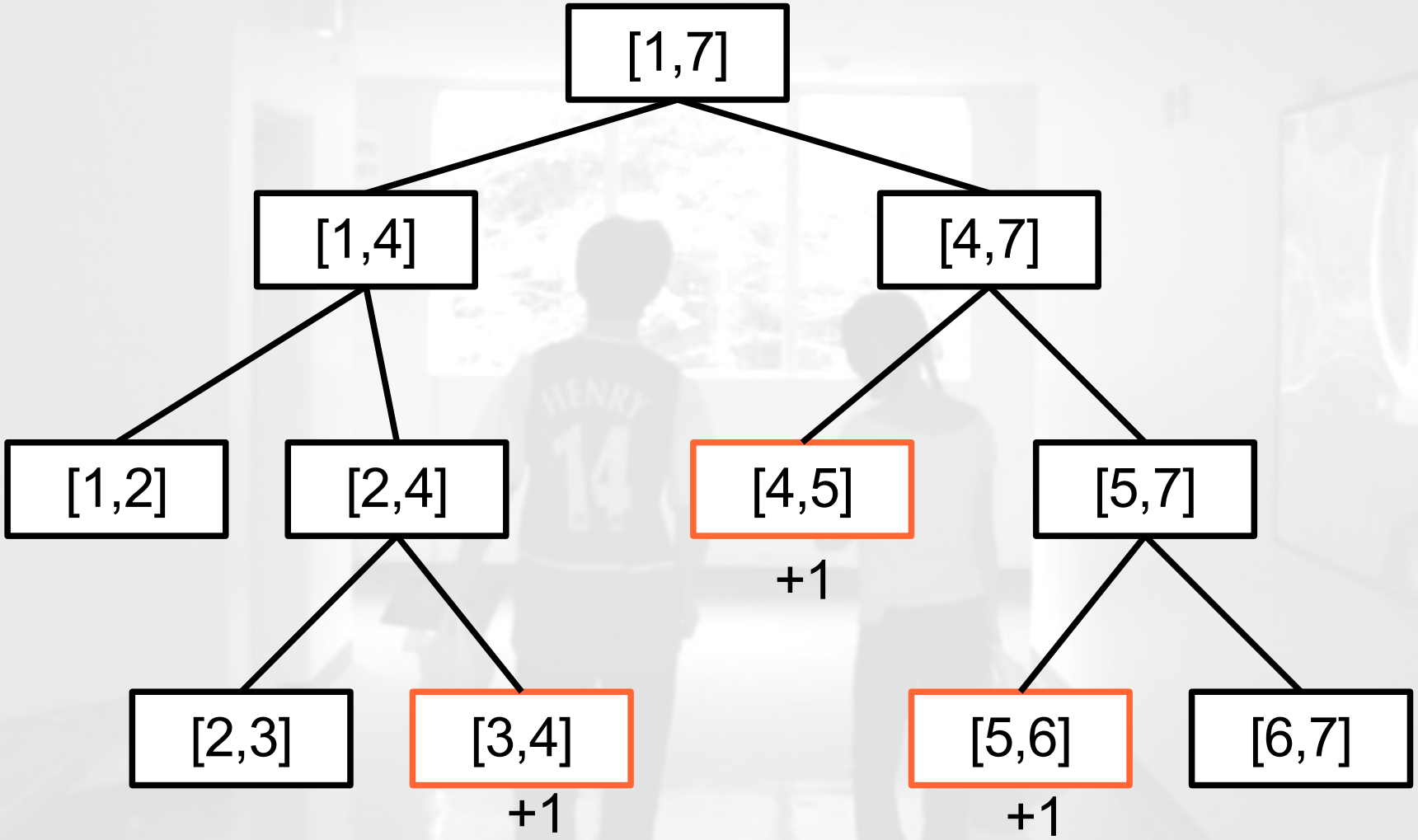
例題4 – Mars map(7/10)

- 例:[1,7]の区間木



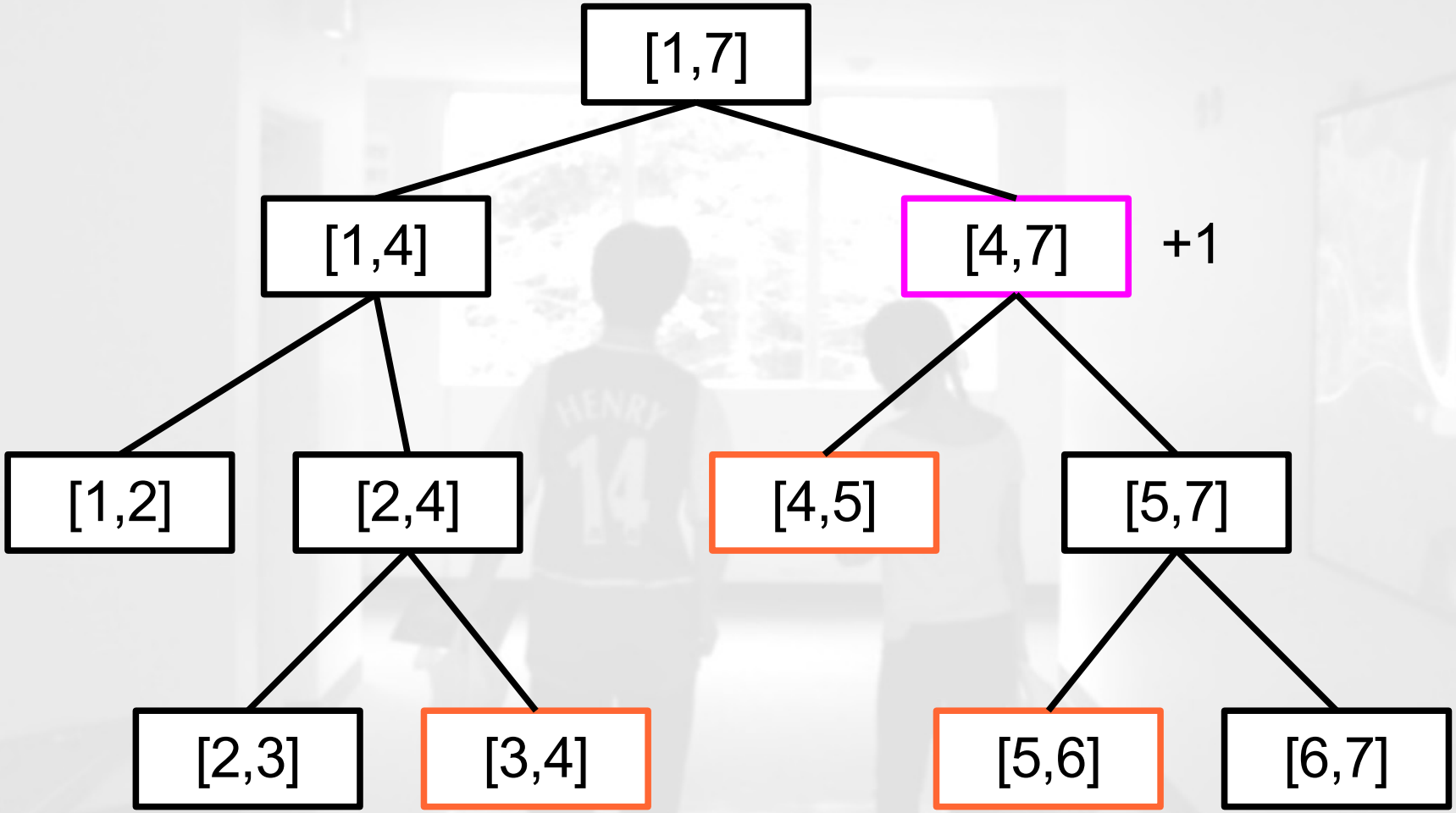
例題4 – Mars map(8/10)

- 例:[3,6]を追加



例題4 – Mars map(9/10)

- 例:[4,7]を追加



例題4 – Mars map(10/10)

- 区間Xを入力
  - まず根に対して処理を行う(v=root)
  - l(v)がXに完全に含まれていたら、vにXを登録（追加、削除に応じて+-1する)
    - l(v)が表す区間の長さL(v)を返す
  - そうでないなら、vの子vl,vrについて、l(vl) (l(vr))がXと交わっていればvl (vr)に対して処理を行う
    - 返値から子を根とするサブツリーの中の範囲の共通部分の長さが判る
    - return L(v)=L(vl)+L(vr);
- rootのlengthが答えとなる

例題5 - Viewer’s Prize in F－TV

- 0と1からなる文字列Xが与えられたとき、Xの部分文字列Sで、S=S^R（回文）となる最長のSの長さを求めよ。

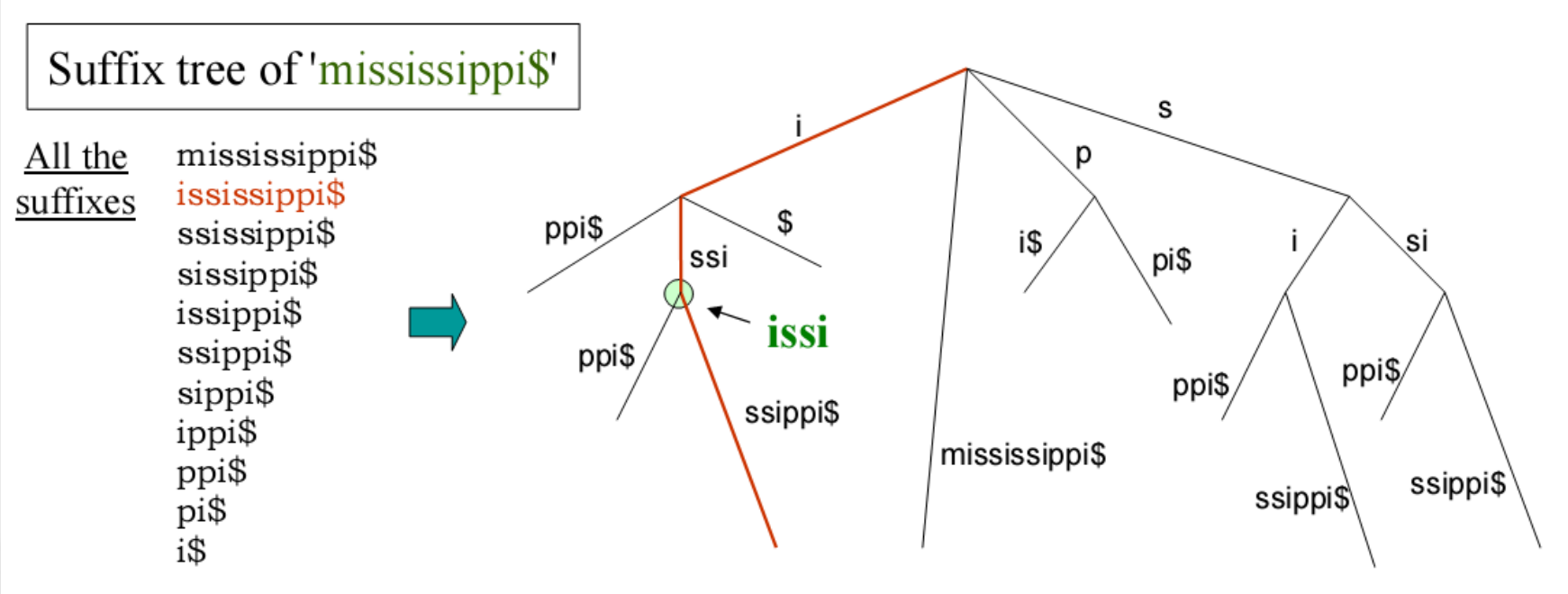
(ACM/ICPC Regional Contest Asia-Kanpur 2001)

Palindrome(1/6)

- 解法
  - 正順と逆順の文字列による接尾辞木(Suffix tree)を作り、もっとも長い共通文字列を求める問題に変形

Palindrome(2/6)

- 接尾辞木(suffix tree)
  - 文字列の全ての接尾辞を表現するtrie木
  - 各枝が部分文字列を表現



Palindrome(3/6)

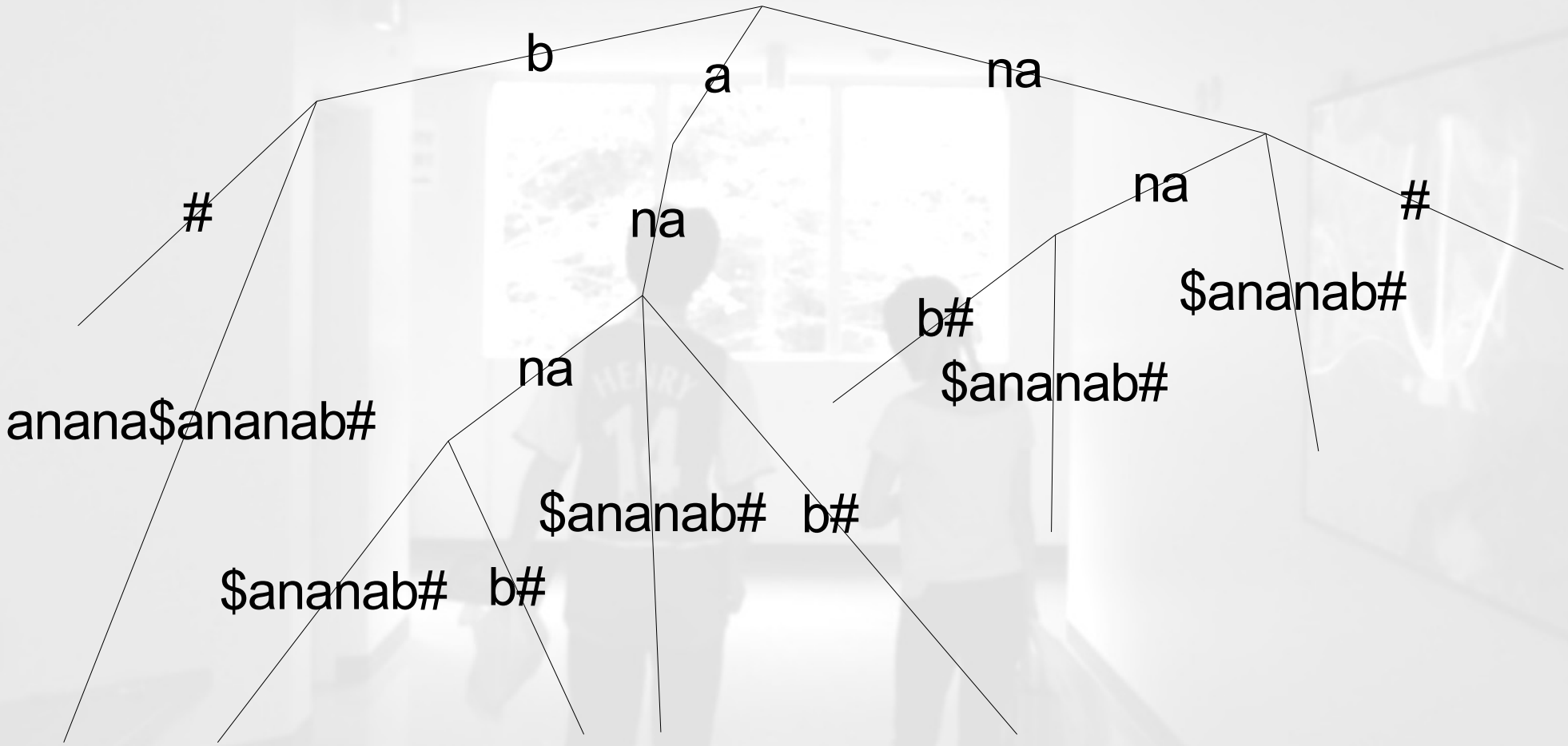
- O(n)でsuffix treeを作るアルゴリズムがある
  - が、かなり複雑
- が、何も考えずに作るとO(n^2)
  - これはまずい
- Weiner
  - O(ns)：sはアルファベットの種類
- Ukkonen
  - O(nlogs)

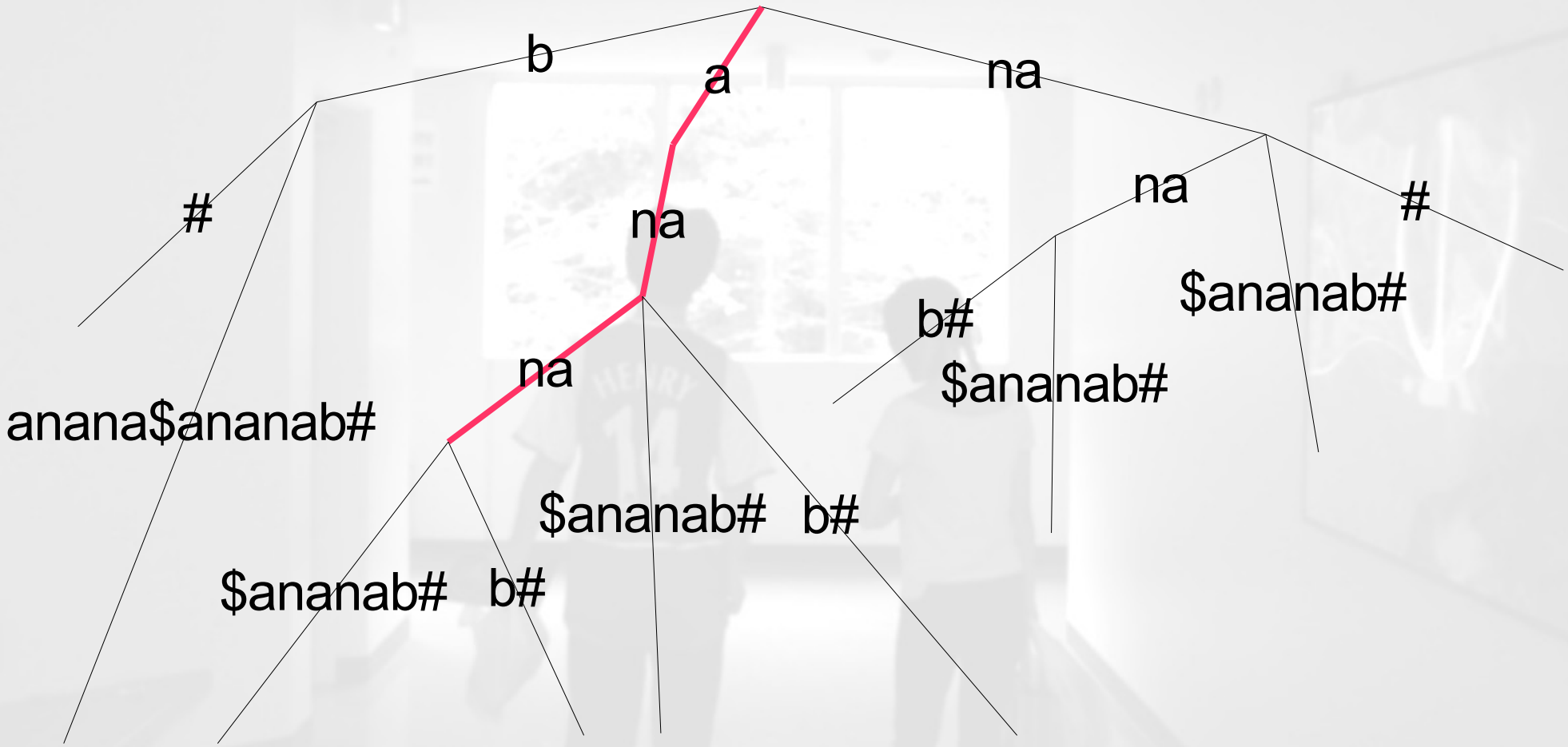
Palindrome(4/6)

- 最長回文(longest palindrome)
  - 文字列Sを逆から読んだものをS'とする
  - S$S'#という文字列のsuffix treeを作り、
  - 子に$と#が現れるノードの中でもっとも深いものを探索する

Palindrome(5/6)

- 例:banana



Palindrome(6/6)

- 例:banana