

2011/05/14 東京大学駒場キャンパス

東京大学プログラミングコンテスト 2011

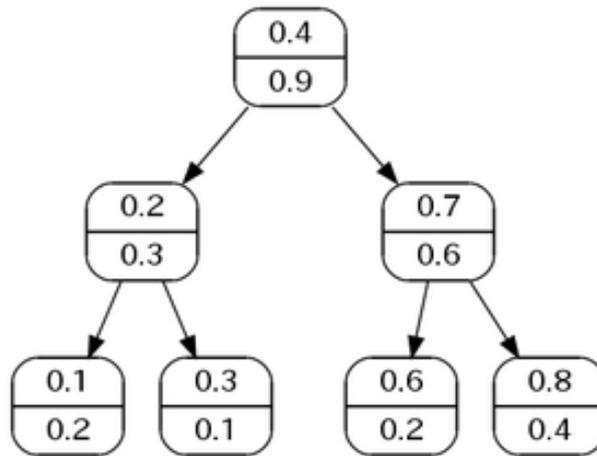
問題 J: 乱択平衡二分探索木

東京大学大学院情報理工学系研究科

秋葉 拓哉

問題概要

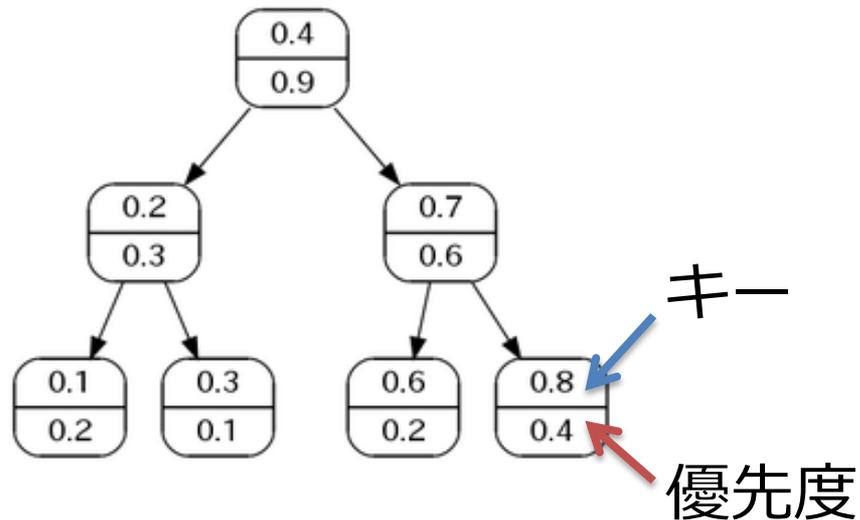
- Treap にランダムな要素を N 個挿入する
- 高さが h になる確率は？
 - $h = 0, 1, \dots, N - 1$ の全てについて出力



Treap の例

Treap

- 乱数を用いる平衡二分探索木
- 各ノードは, キーの他, 優先度を持つ
 - 優先度が挿入時にランダムで決まる

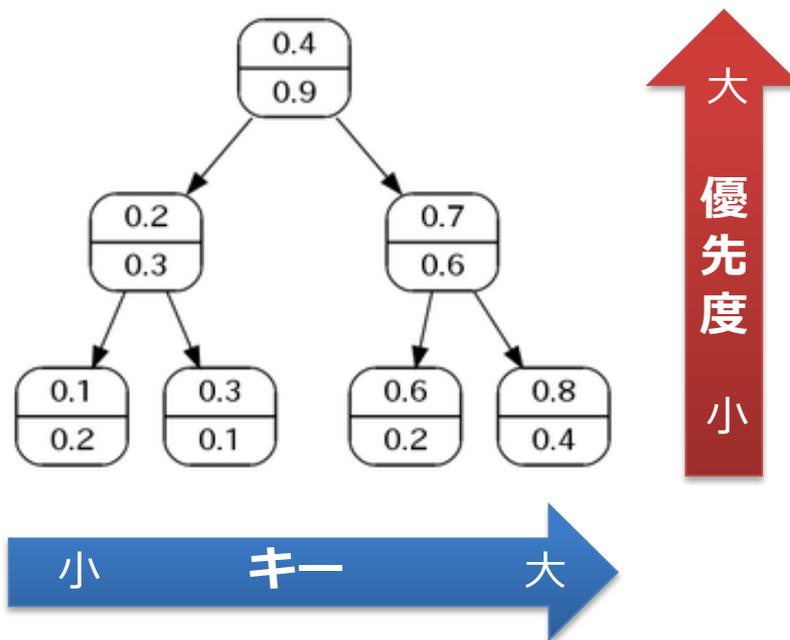


Treap

以下の2つの条件を常に保つ

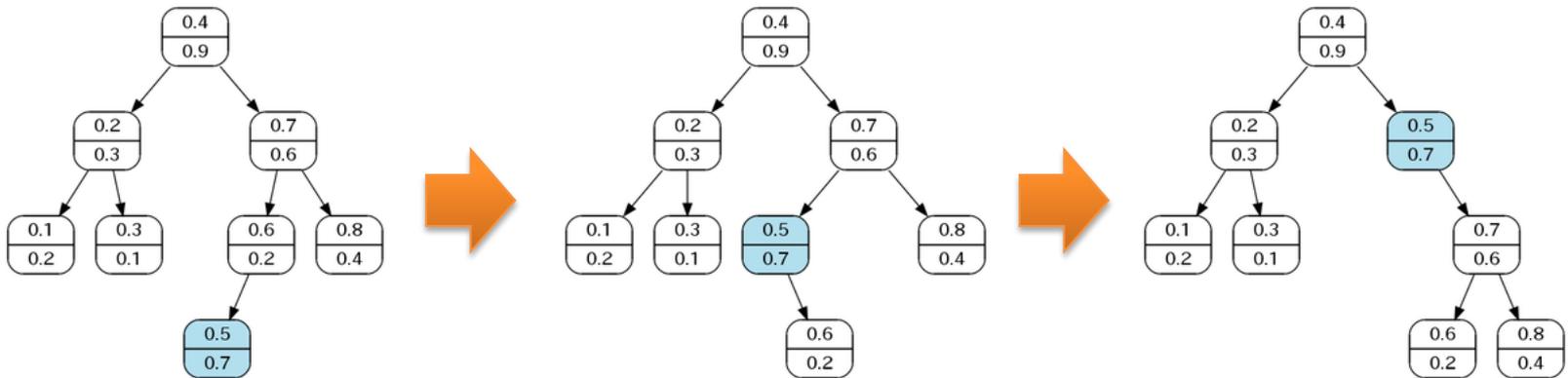
1. キーを見ると二分探索木
2. 優先度を見ると二分ヒープ

(Tree + Heap なので Treap という名前らしい)



Treap での挿入

-
- 実はこの問題ではどうでも良い！
- **そもそも実は Treap とかどうでも良い！**



重要な考察

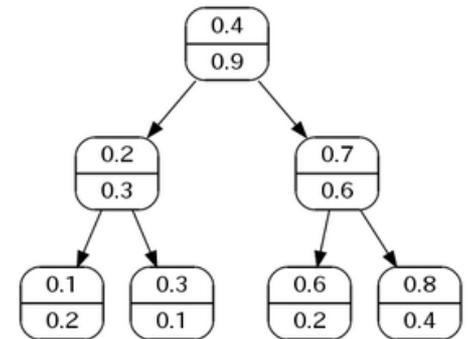
優先度が高いノードから順に、

普通の二分探索木に

挿入していくと考えても同じ。

(普通の二分探索木 = 平衡操作など一切しないやつ)

- Treap
 - 優先度の高いノードが常に上
- 普通の二分探索木
 - 先に挿入したノードが常に上



重要な考察

- 本問題では、挿入されるキーはランダム
- 順番を並び替えても意味ない

よって、

普通の二分探索木

で考えればOK

$O(n^3)$ 時間の DP

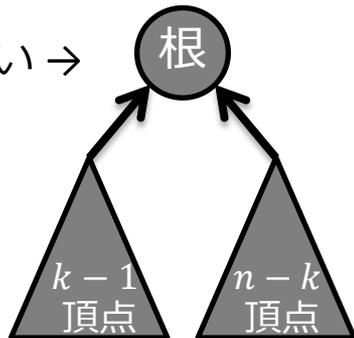
$P(n, h) := n$ 頂点の部分木が高さ h 以下になる確率

$$P(n, h) = \frac{1}{n} \sum_{k=1}^n P(k-1, h-1) P(n-k, h-1)$$

↑
根のキーは等確率で
 k 番目に小さい

↑
左右の部分木が
高さ $h-1$ 以下の確率

k 番目に小さい \rightarrow



木の高さが h 以下 \Leftrightarrow

- 左の部分木の高さが $h-1$ 以下, かつ
- 右の部分木の高さが $h-1$ 以下

$$O(n^3) \rightarrow O(n^2 \log n)$$

Treap は平衡二分探索木



高さが大きくなる確率は小さい

- 許容誤差 10^{-5} より割とすぐに小さくなる
- 計算を打ち切って、あとは 0.0 を出力

$$O(n^2 \log n) \rightarrow O(n \log^2 n)$$

$Q(n, h) := n P(n, h)$ とおくと,

$$Q(n, h) = \sum_{k=1}^n P(k-1, h-1) P(n-k, h-1)$$

- どう見ても**畳込み**
- 高速フーリエ変換で計算すれば良い

$$\begin{array}{ccccccc} & \text{FFT} & & & \text{FFT} & & \\ P(*, 0) & \rightarrow & Q(*, 1) & \rightarrow & P(*, 1) & \rightarrow & Q(*, 2) \rightarrow \dots \\ & O(n \log n) & & O(n) & & O(n \log n) & \end{array}$$

別解 (?)

- 判定で使われる入力 (N の値たち) を特定する
 - 今回用いたシステムでは, 全ケースに対する判定を見ることができた
 - WA, RE, TLE, OLE を値に応じて意図的に引き起こせば, $\lceil \log_4 30000 \rceil = 8$ 回で全ての入力が特定可
- $O(n^2 \log n)$ の DP を実装する
- 手元で実行し, 答えを埋め込んで提出

提出状況

- 最初の提出: omeometo さん (3:15:58)
 - アイディアは正しかった様子
- 最初の正解: rng_58 さん (3:42:59)
- 提出件数: 18
- 正解件数: 1