1 Jak řešit druhý domácí úkol

- Dvojité rotace implementujte pomocí dvou jednoduchých.
- Při každé z operací `Find`, `Insert`, `Delete` je potřeba provést `Splay` na nejnižší navštívený prvek. Zejména to znamená, že pokud při mazání klíče nahrazujete mazaný vrchol následníkem, pak je potřeba provést `Splay` na předchůdce následníka.

2 Úlohy na dnešek

1. Vložíme do splay stromu prvky od 1 do 10.

   a) Proveďte `Find(1)`, `Find(2)`, ..., `Find(10)`.

   b) Zkuste totéž, ale bez dvojitých rotací. Jaký je rozdíl? Má to nějaký dopad? Nebo jsou dvojité rotace jenom buzzword, který vám má zkomplikovat řešení úloh?

2. Jak byste implementovali v BB[α]-stromech operaci `Delete`? Při zachování logaritmické amortizované složitosti (všech operací).

3. Chceme zkonstruovat binární vyhledávací strom pro prvky $1, \ldots, n$. Známe distribuci vyhledávání, tzn. pravděpodobnost, že budeme vyhledávat prvek $k \in [n]$ je předem známé p_k (a platí $\sum_{i=1}^{n} p_i = 1$). Zkonstruujte binární vyhledávací strom, který minimalizuje střední dobu vyhledávání prvku.

   Doba vyhledání prvku k je úměrná hloubce, ve které leží.

4*. Totéž, ale v čase $\mathcal{O}(n^2)$.

5*. Ukažte, že nemůže existovat algoritmus pracující v čase $\mathcal{O}(n)$.

6*. Bude tedy stačit v lineárním čase *aproximovat* optimální střední hodnotu vyhledávání prvku. Tedy najděte algoritmus, který pro nějaké c vrátí řešení, které je nejhůře c-krát horší než optimální hodnota.