3. CVIČENÍ Z DATOVÝCH STRUKTUR 1

BVS vyvažované líně nebo naopak velmi proaktivně (pomocí splay operací)

1. *Dokonalé vyvážení.* Navrhněte algoritmus, který ze setříděného pole vyrobí v lineárním čase dokonale vyvážený BVS (kde pro každý vrchol platí, že počet vrcholů v levém podstromu liší od počtu vrcholů v pravém podstromu maximálně o 1).

Bonus: s logaritmickým prostorem navíc oproti uložení BVS.
Bonus^2: s konstantní pamětí navíc.

2. *Konstrukce optimálního statického BVS.* Mějme množinu klíčů $k_1 < k_2 < \cdots < k_n$, ze kterých chceme postavit statický BVS. Nechť známe dopředu četnosti přístupů $w_1, \ldots, w_n$ k jednotlivým klíčům (které lze také interpretovat jako pravděpodobnosti přístupů). Postavte *optimální BVS*, tedy BVS minimalizující

$$\sum_{i=1}^{n} h(i) \cdot w_i$$

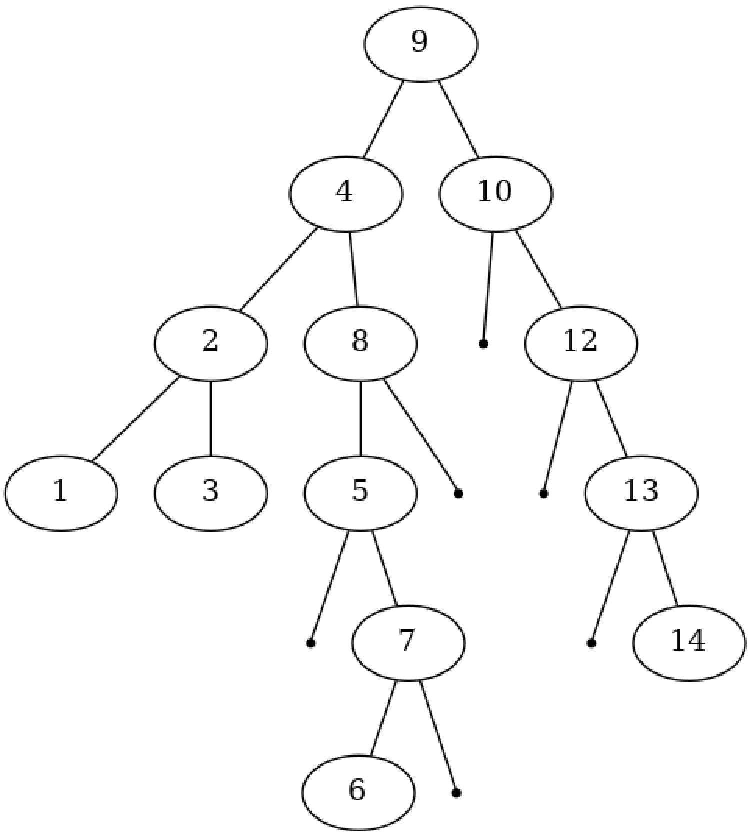
kde $h(i)$ je hloubka klíče k_i v postaveném BVS (kořen má hloubku 0). Jinými slovy provedení w_i přístupů ke klíči k_i pro každé i (beze změny stromu) bude trvat co nejmenší dobu.

Hint: dynamické programování.

3. *Líně vyvažované stromy, neboli BB[α].* Připomeňte si, jak fungují líně vyvažované BVS a jakým potenciálem se analyzují. Jak dlouho může trvat provedení k operací provedených na libovolném BB[α] stromu? Proč je v definici potenciálu výjimka pro rozdíl 1, tedy co by se pokazilo, kdybych ji neudělali?

4. *Splay.* Proveďte operace

1. Splay(14)
2. Splay(6)
3. Insert(11)
4. Delete(6)
5. Delete(10)
6. Delete(4)



5. *Splay naivně.* Co se pokazí, pokud splay stromy implementujeme naivně, tedy jen pomocí jednoduchých rotací jedné hrany?

Hint: zkuste cestu. Jak se chovají dvojité rotace na cestě?

6. *Potenciál pro následníka.* Už jsme si dokázali, že použití n operací následníka na libovolném BVS má složitost $O(n)$ (když začneme ve vrcholu s nejmenším klíčem). Jak to dokázat za pomoci potenciálu, tedy jaký potenciál zvolit?