

### 3. domácí úkol | Vilém Zouhar

#### 1

Vyřešeno na papíře se zadáním. ( $a_n = -1 + 3n + 2^n$ )

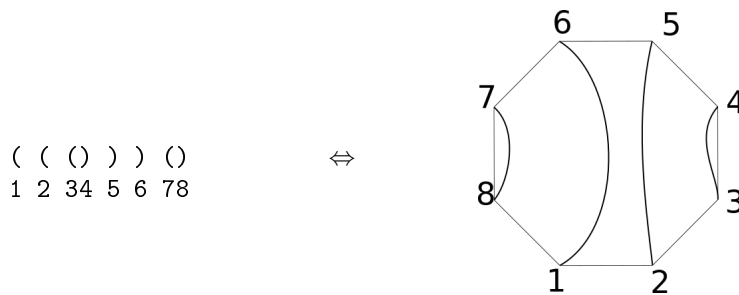
#### 2

Nejprve ukážeme, že počet správných uspořádání u stolu je stejný jako počet správných uzávorkování a pak že počet správných uzávorkování je stejný jako počet binárních stromů.

##### 2.1

**Tvrzení:** Počet správných uspořádání u stolu velikosti  $2n$  je stejný jako počet správných uzávorkování délky  $2n$ . Dobré uspořádání u stolu je takové, že neexistuje čtveřice  $(a, b, c, d)$  sedící u stolu v tomto pořadí, která si chce podat ruce:  $(a, c), (b, d)$ .

**Příklad:**



**Důkaz:** Nejprve ukážeme, že všechna uzávorkování tvoří správné uspořádání u stolu.  $p_o(i) :=$  počet otevíracích závorek v řetězci od 1 po  $i$  (včetně),  $p_u(i) :=$  počet uzavíracích závorek v řetězci od 1 po  $i$  (včetně). Správné uzávorkování (v řetězci délky  $2n$ ) je takové, pro které:

$$\forall i : p_u(i) \leq p_o(i) \quad \wedge \quad p_u(2n) = p_o(2n)$$

V každém bodě se tedy neuzavírá víc závorek než těch, které jsme otevřeli. Proto nikdo nikdy nepodá ruce křížem (neexistuje závadná čtveřice), neboť na to by byla potřeba, aby se uzavřela nějaká závorka dřív, než měla (aby v nějakém bodě bylo víc uzavíracích závorek než otevíracích). To ale nastat nemůže.

Opačně, když máme zajištěno neexistenci závadné čtveřice. Z toho plyne, že pokud máme seřazené hosty  $(a, b, c, d)$  a nahradíme hosta otevírací závorkou, pokud podává ruku někomu novému a zavírací závorkou, pokud podává ruku někomu již objevenému. Nemůže ale nastat, že by bylo víc lidí podávajících ruku zpětně než naopak.

Tedy počet uspořádání u stolu je stejný jako počet správných uzávorkování.

**Tvrzení:** Počet správných uzávorkování délky  $2n$  je stejný jako počet binárních stromů na  $2n$  vrcholech.

**Důkaz:** Vytvoříme zjevnou bijekci pomocí algoritmu. Projdeme řetězec a budeme stavět binární strom. Začneme v kořeni a kdykoliv narazíme na  $($ , vytvoříme v aktuálním vrcholu nového potomka a vnoříme se do něj. Pokud narazíme na  $)$ , vrátíme se o jeden stupeň nahoru a vytvoříme pravého potomka, při dalším návratu půjdeme již o úroveň výš. Pokud dostaneme na vstup správné uzávorkování, nemůže se nám stát, že bychom udělali nepovolenou operaci (zpracovávali bychom  $)$  v kořeni). To nastat nemůže z první vlastnosti správného uzávorkování. Tedy z každého uzávorkování postavíme binární strom. Pokud ale budeme procházet binární strom v nějakém rozumném smyslu (např. DFS), tak úplně stejným způsobem vytvoříme správné uzávorkování (při DFS se nesnažíme vylézt z kořene a končíme v kořeni).

Intuitivnější argument: správné uzávorkování délky  $2n$  reprezentuje aritmetické výrazy složené z  $n$  binárních operací. Tento výraz můžeme snadno reprezentovat i binárním stromem a naopak.

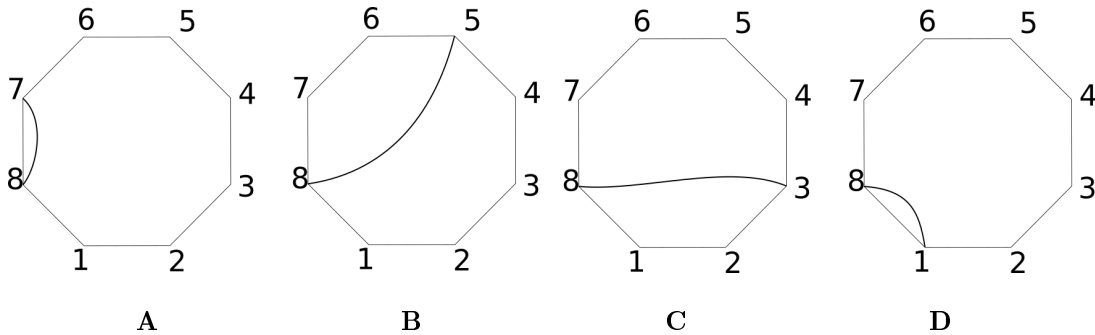
Každopádně z každého binárního stromu vytvoříme správné uzávorkování a binárních stromů na  $n$  vrcholech (každý uzel stojí dvě závorky) je stejně jako správných uzávorkování délky  $2n$ .

Z přednášky víme, že binárních stromů je  $\frac{1}{1+n} \binom{2n}{n}$ . Tolik je tedy i dobrých uzávorkování a správných rozesazení kolem stolu.

## 2.2

Počet se dá vypočítat i přímo za pomoci nástrojů z přednášky. Záleží nám jen na počtu párů, proto tedy naše funkce  $(p)$  bude závislá na  $n$  (počet párů) a nikoliv na  $2n$  (počet lidí), oproti původní funkci  $q$ . Zjevně totiž platí, že  $p_n = q_{2n}$ , ale toto bude zjednodušení.

Pokud přidáme dva lidi  $(8, 7)$  ke stolu, tak:



- A. podají si ruce 8, 7 a ostatní možnostmi jak předtím  $(p_1 \cdot p_3)$
- B. podají si ruce 8, 5 a ostatní možnostmi jak předtím  $(p_1 \cdot p_1 \cdot p_2)$
- C. podají si ruce 8, 3 a ostatní možnostmi jak předtím  $(p_1 \cdot p_2 \cdot p_1)$
- D. podají si ruce 8, 1 a ostatní možnostmi jak předtím  $(p_1 \cdot p_3)$

Celkově  $p_1 \cdot p_3 + p_1 \cdot p_1 \cdot p_2 + p_1 \cdot p_2 \cdot p_1 + p_1 \cdot p_3 = \sum_{i=0}^{4-1} p_i \cdot p_{4-1-i} = p_4$ . To lze zobecnit na:  $p_n = \sum_{i=0}^{n-1} p_i \cdot p_{n-1-i}$ , pro přehlednost  $p_{n+1} = \sum_{i=0}^n p_i \cdot p_{n-i}$ . Intuitivně ještě  $p_0 = p_1 = 1$ . Tato posloupnost je zjevně důsledek násobení (konvoluce) dvou stejných vytvářejících funkcí, proto musí platit:

$$p(x) = x \cdot p^2(x) + 1 \Rightarrow p(x) = \frac{1 \pm \sqrt{1-4x}}{2x}$$

$\frac{1 - \sqrt{1-4x}}{2x}$  konverguje pro  $x$  v okolí nuly, což se nám hodí a proto tento kořen použijeme

za pomoci zobecněné binomické věty:  $\sqrt{1-4x} = \sum_{i=0}^{\infty} \binom{\frac{1}{2}}{i} (-4x)^i$

nyní bychom normálně hledali koeficient u  $x^i$ , ale v předpisu dělíme tuto řadu  $-2x$ , proto nás zajímá  $x^{i+1}$

$$\begin{aligned} \binom{\frac{1}{2}}{i+1} (-4)^{i+1} &= \frac{\frac{1}{2} \cdot \frac{-1}{2} \cdot \frac{-3}{2} \cdot \dots \cdot \frac{1-2i}{2}}{(i+1)!} (-1)^{i+1} 2^{2i+2} = \frac{1 \cdot 1 \cdot 3 \cdot 5 \cdot \dots \cdot (2i-1)}{(i+1)!} (-1)^1 \cdot 2^{i+1} = \\ &= - \frac{1 \cdot 1 \cdot 3 \cdot 5 \cdot \dots \cdot (2i-1) \cdot i!}{(i+1)! \cdot i!} 2^{i+1} = (\text{vynásobíme každý člen faktoriálu v čitateli dvojkou}) = -2 \cdot \frac{(2i)!}{(i+1) \cdot i! \cdot i!} = \\ &= - \frac{2}{i+1} \binom{2i}{i}, (\text{nesmíme zapomenout na dělení } -2 \text{ z původního výrazu}) \rightarrow p_n = \frac{1}{n+1} \binom{2n}{n} \end{aligned}$$

Dospěli jsme ke stejnému závěru jako v minulém postupu.

### 3

#### 3.1

$$\begin{aligned} A(x) + x^{n/2}(C(x) - A(x) - B(x)) + x^n B(x) = \\ D_P(x)D_Q(x) + x^{n/2}(D_P(x)D_Q(x) + H_P(x)H_Q(x) + H_P(x)D_Q(x) + D_P(x)H_Q(x) - D_P(x)D_Q(x) \\ - H_P(x)H_Q(x)) + x^n H_P(x)H_Q(x) = \\ D_P(x)D_Q(x) + x^{n/2}(H_P(x)H_Q(x) + H_P(x)D_Q(x)) + x^n H_P(x)H_Q(x) = \\ (D_P(x) + x^{n/2}H_P(x))(D_Q(x) + x^{n/2}H_Q(x)) = P(x)Q(x) \end{aligned}$$

Ověřeno.

#### 3.2

Počet násobení pro výpočty jednotlivých pomocných polynomů:

$$A \mapsto 1 \cdot N(n/2), B \mapsto 1 \cdot N(n/2), C \mapsto 1 \cdot N(n/2), PQ \mapsto 3 \cdot N(n/2)$$

$$\Rightarrow N(n) \leq 3N(n/2) \text{ (některá násobení se nám můžou opakovat, proto horní odhad)}$$

#### 3.3

Pro zjednodušení zavedeme funkci  $T : \mathbf{N} \mapsto \mathbf{N} : T(i) := N(2^i) \leq 3N(2^i/2) = 3T(i-1)$

$$\Rightarrow T(i) \leq 3T(i-1) \Rightarrow T(i) \leq 3^i$$

$$(n = 2^i \rightarrow i = \log_2(n)) \Rightarrow N(2^{\log_2(n)}) = N(n) \leq 3^{\log_2(n)}$$

$$N(n) \leq (2^{\log_2(3)})^{\log_2(n)} = n^{\log_2(3)} \Rightarrow N(n) \leq n^{\log_2(3)}$$

#### 3.4

$$(2^{10})^2 = 2^{20} \gg 2^{10 \log_2(3)} = (2^{10})^{\log_2(3)} \quad (\log_2(3) \approx 1.58)$$

$$\frac{2^{20}}{2^{10 \log_2(3)}} \approx 2^{4.15} \approx 17 \Rightarrow \text{alternativní metoda je zhruba 17x úspornější}$$