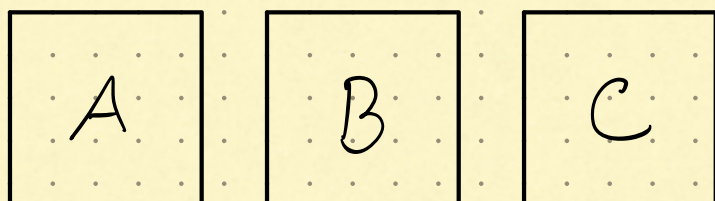


3 nádoby s objemoch a, b, c
počátečnými objemami x, y, z

- chceme počet krokov $\rightarrow$ BFS

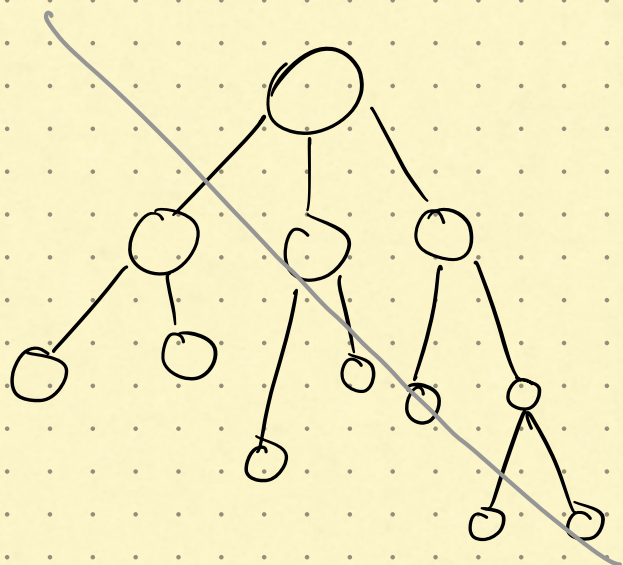


možné objemy = n

- max možný objem = $\min(a, b, c, x+y+z) = n-1$

- nemůžeme hledat specifické hodnoty

$\rightarrow$ chceme nůchť, dokud nenarazíme na již
nalezenou kombinaci, pak zabijeme větev



- chceme jen jeden průchod

- ukládáme do (int, int)

nebo **fronta**

$\rightarrow$ pokud new { přidáme možné hltí }
 ≤ 5

- init for $0 \leq i \leq n$: $\{ (i, j-1) \dots \}$
 ↑
 Sčítanom výsledku

• X, Y, Z : do listen

- rytvarit frontu

- new list zažitých situací $\rightarrow \{ (x, s, z) \}$

class Nadoby (a,b,c)

a. kapacita objemu

ب

C

für $\text{bij}(x, y, z, \text{odhnd-lamm}) \Rightarrow x, y, z$

Ice rozszer frontu $(x, y, z) \leftarrow$ uśrednij możliwe prędkości

main

↳ loop prichodn
fronta

historie statusu
list výsledků

~~rezurzivní průchod~~
průchod samostatných
situací

fronta = { (N^0 přelítí, x, y, z , odhodl-hamw)

fronta.append(rozsir-frontu (x, y, z))

loop : hop = fronta.pop()

preliti = hop[0] ++

new_state = Nadohy.hij (hop[1], ..., hop[4])

if ! new_state in history-of-states

→ { list-vysledku.update (x, y, z , preliti)
historie-statusu.append (x, y, z)
rozsir-frontu (x, y, z)
}

→ konstatni update vysledku (extra Class+fe)

→ pruchod historie na konci (extra pelmel, položek, struktura historie)

(x, y, z) → bool (*int, *int, *int)
upgrade