

Řešení

Tak samozřejmě první řešení které nás napadá je hrubě projít každou mističku pro každého člověka. Toto jednoduché řešení vyjde na složitost $\mathcal{O}(N \cdot K)$, což není až tak hrozné. Ale existuje určitě lepší řešení.

Jaké je tedy to lepší řešení? No, je to pomocí něčemu čemu se říká intervalové stromy, o kterých jsem se dozvěděl z knížky *Průvodce labyrintem algoritmů*.

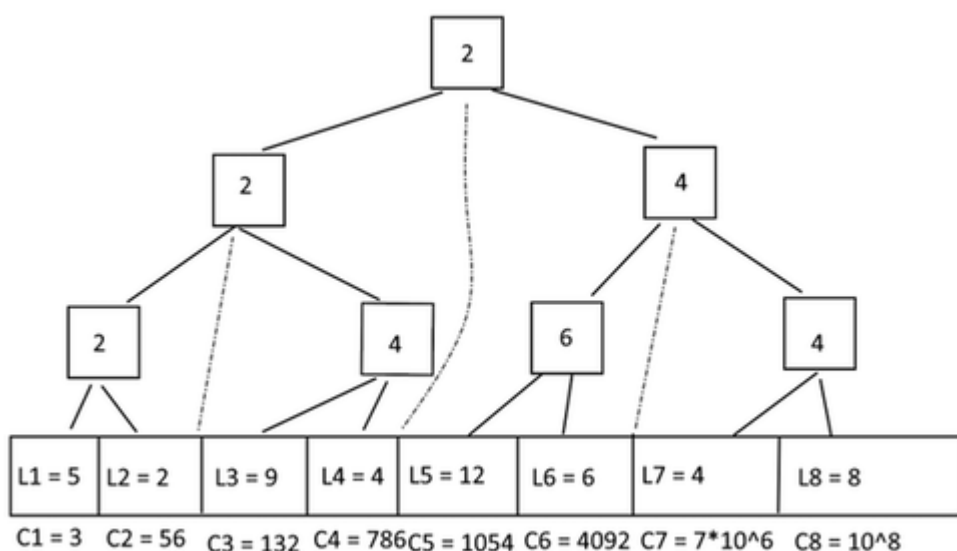
Prvím krokem této řešení je, že si nejprve uspořádáme všechny mističky podle toho, jakou koncentraci čaje vytvoří. Toto bude vyžadovat $\mathcal{O}(K \log K)$ kroků.

Poté vytvoříme tedy ten intervalový strom. Je potřeba uvažovat o mističkách jako objekty, které drží více informací. Přesněji jejich původní pořadí při načtení ze vstupu (což bychom pravděpodobně potřebovali při praktickém řešení úlohy), poté ještě počet listů na vystelení mističky (L_i) a koncentraci čaje, kterou mističkou vytvoříme (C_i).

Mističky máme seřazené podle jejich koncentraci, ale intervalový strom bude ukládat počet minimální počet listů na nějaké intervaly. Strom bude binární, proto bude velmi jednoduché ho uložit do pole. Tvorba stromu vyžaduje $\mathcal{O}(K)$ kroků (přesněji $2K$), protože můžeme začít ze zdola (z původního pole) a postupně ukládat prvky výš ve stromě, tedy blíž k začátku pole do kterého strom ukládáme.

Přes strom budeme tedy schopni dotázat na nejmenší počet listů na vystelení misky pro daný interval v $\mathcal{O}(\log K)$ čase.

Příkladem stromu je tohle (omluvte prosím velmi nízkou kvalitu obrázku):



Protože budeme dotazovat na strom přes koncentraci a ne indexy, každý vrchol nad původním polem. Jednak, každý vrchol bude muset mít uložené hodnoty koncentrace nejvíce vpravo a nejvíce vlevo, aby se mohlo zjistit *identita* intervalů.

Podle příkladu v obrázky by tohle znamenalo, že kdybychom se ptali na interval $\langle 2; 10^9 \rangle$, tak rovnou vrátíme nejvyšší vrchol, protože bude podmnožinou našeho dotazu, ale zároveň bude obsahovat všechny možné mističky pro ten dotaz.

Další hodnoty, které vrcholy budou muset mít uložené (protože jejich uložení zabere konstantní čas při tvoření stromu) budou hodnoty nejbližší ke středu (t.j. ty pod příslušným vrcholem). Toto umožní, abychom mohli správně rozdělovat dotazy na více vrcholů v stromu.

Například dotaz $\langle 4; 10^9 \rangle$ zavolá dvě dotazy, první na levého podstromu $\langle 4; 786 \rangle$ a druhý na pravý podstrom $\langle 1054; 10^9 \rangle$

Ted' už jsme připraveni abychom mohli dotazovat na náš strom v $\mathcal{O}(\log K)$ času. Ještě si uvedeme pseudokód dotazu:

Algorithm 1 DotazIntervalů

Input: index vrcholu v ve stromu, interval vrcholu $\langle a; b \rangle$, hledaný interval $\langle i; j \rangle$

if $i \leq a$ and $j \geq b$ **then**

 return v

end if

$S_l \leftarrow$ hodnota hned vlevo od 'středu' vrcholu \triangleright v příkladu by pro $v=0$ rovnalo 786

$S_r \leftarrow$ hodnota hned vpravo od 'středu' vrcholu \triangleright v příkladu by pro $v=0$ rovnalo 1054

if $j \leq S_r$ **then**

 return DotazIntervalů ($2v$, $\langle a; S_l \rangle$, $\langle i; j \rangle$)

else if $i \geq S_l$ **then**

 return DotazIntervalů ($2v + 1$, $\langle S_r; b \rangle$, $\langle i; j \rangle$)

else

 return DotazIntervalů ($2v$, $\langle a; S_l \rangle$, $\langle i; S_l \rangle$)

 return DotazIntervalů ($2v + 1$, $\langle S_r; b \rangle$, $\langle i; S_r \rangle$)

end if

Output: All intervals that $\langle i; j \rangle$ contains

Jaká je teda složitost tohoto řešení? No, nejprve si vypočítáme časovou složitost. Třídění mističek je $\mathcal{O}(K \log K)$, tvoření stromu je $\mathcal{O}(K)$ a všechny dotazy budou $\mathcal{O}(N \log K)$. Celkově to tedy vyjde na $\mathcal{O}(K \log K + N \log K)$.

Je důležité si ale poznamenat, že toto řešení vlastně ani nemusí být vždycky lepší než jednoduché, hrubé řešení uvedené na začátku. Rovnou změníme obecné $\log K$ na $\log_2 K$, které je přesnější pro tyto výpočty.

$$\begin{aligned} N \cdot K &< K \log_2 K + N \log_2 K \\ 1 &< \frac{\log_2 K}{N} + \frac{\log_2 K}{K} \end{aligned} \tag{1}$$

Už jenom na této nerovnici můžeme upozorovat, že pro $N \leq \log K$ bude intervalový strom horší. Samozřejmě, když se budou obě hodnoty blížit nekonečnu, tak intervalový strom je mnohem lepší, jen je ale důležité si pamatovat, že pro velmi velké K může existovat ne úplně malé N , pro který bychom preferovali jednoduchý, hrubý algoritmus.

A teď si ještě rychle na konci spočítáme prostorovou složitost. Nejvíce paměti bude zabírat náš strom, takže započítáme jenom jeho: $\mathcal{O}(K)$ (přesněji to vyjde okolo $6K$)