

Daniel Culliver

35-Z2-4 Supervěž

Čtení šachovnice samotné už je N^2 operací, ale bylo mi sděleno velmi dobrý zdroj (Matúšem), že časová komplexita se počítá jako lineární vůči čtení celého vstupu, tedy čtení $N \times N$ šachovnice má časovou komplexitu $O(n)$.

První možnost, která člověku napadne, je převést šachovnici na 2D array 1 a 0. Poté sečíst řádek a sloupec pro každé místo. Toto vyjde na $N^2 * 2N$ operací, což dohromady se čtením šachovnice vyjde na $2N^2 * 2N$, nebo-li $4N^3$, což odpovídá časové komplexitě $O(N^3)$ nebo-li $O(n \log n)$.

Existuje ale optimalizace. Místo sečtení řádku a sloupce pro každý čtvereček, sečteme jenom každý řádek a sloupec a uložíme ho do array. Poté stačí projít Každý čtvereček, ale místo $2N$ operací, to bude jenom 2 operace. Toto vyjde na $2N^2 * 2$ operací, nebo-li lineární časová komplexita vůči vstupu $O(n)$. Vzhledem k tomu, že to je stejná komplexita jako čtení vstupu, pravděpodobně se nedá dál optimalizovat časová složitost.

Druhá část komplexity je paměťová. Uložení array šachovnice zabírá N^2 místa a součty řádků a sloupců je dalších $2N$. Toto vychází na paměťová komplexita $O(N^2)$. Ale my nepotřebujeme array šachovnice potom, co jsme sečetli řádky a sloupce, tudíž můžeme ukládání arraye zanedbat a rovnou sečíst řádky a sloupce při čtení vstupu. Toto zabírá $2N$ místa a uložení čísla N pro for loopů můžeme zanedbat. To vychází na paměťovou komplexitu $O(N)$, tedy $O(\log n)$.

Jako závěr jsme došli k časové komplexitě $O(n)$ a paměťová komplexita $O(\log n)$.

Omlouvám se za veškeré \tilde{t} v dokumentu. Latex píšu dneska podruhé a pravděpodobně je to nastaveným fontem (ještě řeším jak to zpravit).