

Léčivá alpa

Bojovníky máme zadané jako pole čísel, ve kterém hledáme nejdelší sekvenci, pro kterou platí, že součet jejích prvků (počtu zranění jednotlivých bojovníků) je dělitelný počtem kapek v ampulce. Nejjednodušší je procházet toto pole od konce v cyklu, protože si můžeme zavést jednu proměnnou dostatečné kapacity, do které si budeme ukládat celkový počet zranění v řadě bojovníků počínající tím, u kterého stojíme a končící posledním tak, že v každé iteraci přičteme počet zranění dalšího bojovníka, ke kterému jsme přišli. Následně projdeme každou sekvenci bojovníků, která začíná tím, u kterého zrovna stojíme a končí nějakým bojovníkem i mezi námi a koncem řady. Protože hledáme nejdelší sekvenci, budeme zase postupovat od konce a pro každého bojovníka i zkontrolujeme, zda náhodou není počet zranění v sekvenci dělitelný počtem kapek. Pokud dělitelný je, našli jsme nejdelší možnou sekvenci pro daný začátek a můžeme ji porovnat s již nalezeným maximem, případně si ji jako nejdelší zapamatovat. Nikdo se nás neptá třeba na index bojovníka, u kterého začínáme, takže nám stačí jediná proměnná pamatující si nejdelší sekvenci, která se na konci algoritmu stane výstupem.

Takovéto řešení má lineární paměťovou náročnost $O(n)$, náročnost časová je na tom však o trochu hůře, ne však příšerně. Pro každého bojovníka i provádíme operaci pro všechny bojovníky j ; $i \leq j \leq \text{početBojovníků}$, což je přibližně $O(n \cdot (n-1)/2)$, což je asymptoticky $O(n^2)$.

Na paměťové náročnosti se nic moc zlepšit nedá, časová náročnost tohoto algoritmu však není ideální, protože se mu opakovaně nepodařilo projít poslední sadou testovacích vstupů. Přestavěl jsem proto několikrát všechny části kódu, nicméně jsem nenalezl žádné významné zrychlení. Víím, že nejnáročnější je část samostatného prohledávání pole bojovníků a “nekonečné” volání modul, neumím však algoritmus zrychlit.