Cvičenie 5

Príklad 5.1.

Vypočítajte hodnotu kombinačného čísla $\binom{n}{k}$ modulo p, pričom viete, že $n \leq 10^6$ a p je prvočíslo. Ako by ste postupovali, keby ste mali na vstupe q dvojíc čísel a pre každú by ste mali vypočítať dané kombinačné číslo?

Náčrt riešenia 5.1.

Vieme, že $\binom{n}{k} = \frac{n!}{(n-k)!k!}$. Samozrejme, keďže výsledok chceme modulo p, deliť nemôžeme. Namiesto toho však môžeme násobiť inverzným prvkom. Riešenie teda dostaneme tak, že si vypočítame hodnoty $n!$, $(n-k)!$ a $k!$ (modulo p), pre posledné dve hodnoty si vypočítame inverzný prvok, napríklad ich umocnením na $p-2$ a tieto hodnoty vynásobíme.

V prípade, že by sme chceli zisťovať hodnotu viacerých kombinačných čísel, nemôžeme si dovoliť počítať hodnoty faktoriálov zakaždým odznova. Namiesto toho si raz vypočítame všetky hodnoty faktoriálov po $n!$ a uložíme si ich do poľa (uvedomme si, že ich počítame už pri jednom výpočte $n!$), do druhého poľa vypočítame inverzné prvky týchto faktoriálov a následne už počítame kombinačné čísla v konštantnom čase.

Príklad 5.2.

Máme dve čísla – horné a dolné, ktoré sú na začiatku nastavené na 1. Takisto máme dve tlačidlá H a D. Keď stlačíme tlačidlo H, horné číslo sa nahradí súčtom oboch čísel, keď stlačíme tlačidlo D, tak sa súčtom oboch čísel nahradí dolné číslo. Nájdite postupnosť n stlačení, na konci ktorej je väčšie z týchto dvoch čísel rovné S.

Náčrt riešenia 5.2.

Vieme, že väčšie číslo je po n stlačeniach rovné S. BUNV nech je to horné číslo. Nech je dolné číslo rovné x. Ktoré tlačidlo bolo stlačené ako posledné? Ako vyzerali tieto dve čísla pred tým, ako sme ho stlačili?

Keďže $S > x$, tak posledné stlačené tlačidlo muselo byť H, inak by x muselo byť súčet čísla S a toho čo bolo pred tým dole, nevznikajú nám však záporné čísla, takže tento prípad nastať nemôže. Číslo S preto vzniklo ako súčet x a $S-x$. Pred stlačením H bolo teda horné číslo rovné $S-x$. A opäť vieme určiť, ktoré tlačidlo bolo stlačené ako posledné.

Vždy to bude to, kde sa zrovna nachádza väčšie číslo. A takisto ľahko zistíme, ako vyzeral stav pred tým. Prvé riešenie je teda postupne skúšať všetky možné $x < S$, z ktorých už ľahko vyvodíme ako vyzerali všetky stlačenie. Stačí nám overiť, či sa týmto spätným zisťovaním dostaneme do stavu s $(1,1)$ po práve n stlačeniach.

Toto by však bolo ešte príliš pomalé. Pokúsme sa zrýchliť zisťovanie stlačení. Nech je S fakt veľké oproti x. To znamená, že $S-x$, $S-2x$ … sú stále väčšie ako x. Koľkokrát teda stlačíme tlačidlo H, kým bude $S-kx$ menšie ako x? No predsa $S \div x$ krát. A horné číslo bude $S \mod x$. Celé je to preto iba Euklidov algoritmus na výpočet najmenšieho spoločného deliteľa. Vieme teda preskakovať veľa rovnakých krokov a dostaneme zložitosť $O(S \log S)$.

Príklad 5.3.

Mali sme reťazec, ktorý bol tvorení n nulami a j jednotkami. Takisto sme mali funkciu, ktorá vyzerala nasledovne: $f(0,0) = 1$, $f(1,0) = 1$, $f(0,1) = 1$ a $f(1,1) = 0$. Postupne sme túto funkciu aplikovali na náš reťazec: zobrali sme prvé dve cifry a použili funkciu $f()$. Výsledok sme potom vložili do $f()$ s treťou cifrou, výsledok toho so štvrtou atď. Vieme, že na konci nám vyšiel bit b. Ako mohol vyzerať pôvodný reťazec? Koľko máme možností (modulo 10^9+9)?

Náčrt riešenia 5.3.

Dôležité pozorovanie je, že $f(x,0) = 1$ bez ohľadu na to, čo je x. Ak si teda zoberieme nejaký reťazec a pozrieme sa na poslednú 0, všetko čo je pred ňou nijak neovplyvní výsledok. A za ňou sú len samé 1, ktoré znegujú druhý bit. Ak je teda $b=1$, za poslednou 0 je párny počet 1, v opačnom prípade nepárny.

Postupne teda skúšame, ako vyzerá koniec našeho reťazca – koľko jednotiek je za poslednou 0. Označme si toto číslo x. Koľko je potom takýchto reťazcov? Pred poslednou 0 to môže vyzerať ľubovoľne, koľko máme teda možností? No predsa $\binom{n+j-1-x}{n-1}$. Máme totiž ešte $n+j-1-x$ voľných miest, a chceme si vybrať $n-1$ miest kam dáme nuly.

Príklad 5.4.

Majme funkciu $f(x)$, ktorá vracia počet 1 v binárnom zápise čísla x. Je jasné, že keď budeme túto operáciu opakovať, časom dostaneme hodnotu 1, na ktorej sa výsledok ustáli. Našou úlohou je zistiť počet takých čísel z intervalu $< l, r >$ ($l \leq r \leq 10^{18}$), ktoré sa prvýkrát zobrazia na 1 po práve k volaniach funkcie $f()$.

Náčrt riešenia 5.4.

Prvým trikom je odstránenie si intervalu $< l, r >$. Namiesto toho budeme riešiť o niečo všeobecnejšiu úlohu, v ktorej chceme nájsť počet vhodných čísel **menších alebo rovných** ako a. Ak vieme riešiť túto úlohu (a mali by sme, pretože l môže byť 1), vieme riešiť aj konkrétnejší problém, stačí totiž, že vypočítame výsledok pre $a = r$ a $a = l - 1$ a tieto výsledky odčítame.

Následne si môžeme všimnúť, že hodnota čísla x sa pomerne rýchlo zmenšuje. Presnejšie, už po prvom zavolaní bude výsledná hodnota nanajvýš 63. Prvých 63 hodnôt si vieme ľahko predpočítať pomocou dynamického programovania. Presnejšie, pre každé x menšie ako 63 budeme chcieť vedieť, koľkokrát budeme musieť použiť funkciu $f()$ aby sme dostali číslo 1. Je jasné, že táto hodnota je o 1 väčšia ako počet volaní pre číslo $f(x)$.

Tieto malé hodnoty nám prezradia, koľko jednotiek má v binárnom zápise nami hľadané číslo. Ak totiž zistíme, že číslo p sa zmení na 1 po $k-1$ volaniach funkcie $f()$ tak to znamená, že ľubovoľné číslo s p jednotkami v binárnom zápise sa zmení na 1 po k volaniach. My ale hľadáme iba tie čísla s p jednotkami, ktoré sú menšie ako r. Ako na to?

Opäť využijeme dynamické programovanie. Číslo r si zapíšeme do binárnej sústavy. Následne sa budeme rozhodovať, ako vyzerá naše číslo s p jednotkami. Ktorá cifra bude na najvýznamnejšej pozícii (na ktorej je pri čísle r určite 1)? Môžeme tam dať 0. V takom prípade ale všetky čísla, ktoré dostaneme doplnením zvyšných pozícií budú menšie ako r. To znamená, že našich p jednotiek môžeme rozdeliť ľubovoľne, počet možností bude teda jedno kombinačné číslo.

A čo keď sa na prvú pozíciu rozhodneme dať 1? V takom prípade sa musíme pozrieť ďalej, pričom vieme, že použiť môžeme už len $p - 1$ jednotiek. Pri ďalšej cifre čísla r sa opäť môžeme rozhodnúť ak je táto cifra 1. Ak je táto cifra 0 tak nám neostáva nič iné ako tiež použiť 0, pretože použitím 1 by sme číslo zväčšili.