Finále soutěže ACM

Filip Hlásek

Abstrakt. Tento příspěvek je věnován úlohám, které se vyskytly na světovém finále soutěže ACM-ICPC konaném v marockém městě Marrákéš od 15. do 20. května 2015. Při řešení několika z nich bylo užitečné využít matematické znalosti a odhalit zajímavé zákonitosti. Přesně na ty úlohy se podíváme a pokusíme se je vyřešit.

Zkratka ACM je nesprávné označení jedné z nejstarších programátorských soutěží, jejíž celé jméno je ACM International Collegiate Programming Contest (ACM-ICPC) a je každoročně pořádána pod záštitou americké společnosti Association for Computing Machinery. Dále stojí za zmínku americká univerzita Baylor, na které soutěž v roce 1977 založil její dosavadní prezident profesor William B. Poucher. Oficiálně se mohou účastnit tříčlenné týmy studentů z jedné univerzity a během pětihodinového klání na ně čeká 8 až 13 úloh. K jejich řešení mohou využít jediného počítače, což vždy vyžaduje ohromnou dávku spolupráce, vstřícnosti a porozumění. Odevzdaný program je přijat za správný pouze tehdy, vydá-li pro všechny připravené vstupy správnou odpověď v daném časovém limitu.

Každý rok se na konci jara nebo na začátku léta koná mezinárodní finále, kam postoupí několik nejlepších týmů z každého geografického regionu. Naše týmy soupeří vždy na podzim se svými konkurenty ze střední Evropy obvykle o tři až pět míst (přesný počet závisí na umístění týmů v předchozím finále). Týmu z Karlovy univerzity se podařilo v roce 1998 v americké Atlantě porazit ve finále všechny ostatní a odvézt si domů obrovský pohár. Samotné finále je velice prestižní událost a se všemi formálními ceremoniemi, doprovodným programem a testovacími koly trvá bezmála celý týden. Ani probojovat se v silné konkurenci do finále není často vůbec snadné. Přesto se to v roce 2015 týmu organizátorů PraSátka ve složení Filip Hlásek, Mirek Olšák a Štěpán Šimsa podařilo. Navíc byly soutěžní úlohy finále tentokrát hodně *matematické* a našemu týmu se velmi hodily zkušenosti z matematických soutěžích. Cílem tohoto příspěvku je ukázat, jak je možné matematické dovednosti uplatnit v příbuzném oboru.

Jak vypadá úloha na programátorské soutěži

Běžná programátorská úloha se skládá z popisu, jak bude vypadat vstup, a přesné definice očekávaného výstupu. Obvykle se vstup i výstup skládá z několika čísel či posloupnosti písmen (tzv. řetězců). V každém správném zadání úlohy nalezneme také omezení na velikosti všech čísel a na délky všech řetězců. Není potřeba ošetřovat nevalidní a chybné vstupy. Často je poměrně snadné implementovat nějaké řešení daného problému a hlavní výzvou zůstává časový limit na běh programu, který je obvykle nastaven na několik sekund.

Jak odhadnout dobu běhu programu na vstup dané velikosti

Dnešní počítače zvládnou vykonat několik miliard elementárních operací za sekundu. Bohužel není snadné odhadnout, kolik operací vykoná daný program na konkrétních vstupních datech. Při odhadování není třeba být přesný, protože spustíme-li program na jiném počítači, poběží jinak dlouho a málokdy dopředu víme přesnou konfiguraci stroje. Nejprve uvedeme přesnou definici, pomocí které lze řádově aproximovat rychlost růstu některých funkcí, a poté se ji naučíme intuitivně využívat na odhady časové složitosti algoritmů, jak se často nazývá počet operací, které daný program vykoná na vstupu konkrétní velikosti.

Definice. (Asymptotický odhad, „velké O notace“) Řekneme, že funkce g je „velké O od f“, značíme $g(n) \in \mathcal{O}(f(n))$, pokud existuje kladná konstanta c taková, že pro všechna přirozená čísla n platí $g(n) \leq c \cdot f(n)$. Tento vztah můžeme také popsat slovně tak, že funkce g je asymptoticky menší nebo rovna funkci f.

Uvedená definice je poměrně robustní a těžko uchopitelná, ale my se naučíme uvedené vztahy chápat intuitivně.

Cvičení. U uvedených dvojic funkcí určete tu, která je asymptoticky větší, případně ukažte, že jsou asymptoticky *stejné* (tj. obě jsou větší nebo rovny té druhé).

(1) $f(n) = n$, $g(n) = n^2$,
(2) $f(n) = 3n$, $g(n) = n$,
(3) $f(n) = 2n^4 - 3n^3 + 5n^2 - n + 17$, $g(n) = 29n^3 - 5n + 3$,
(4) $f(n) = n^{99}$, $g(n) = 1{,}01^n$,
(5) $f(n) = \sqrt{n}$, $g(n) = \log n$.

Při práci s asymptotickou složitostí je vždy potřeba si představovat, že nás zajímají opravu velké hodnoty n. Důležité je, jak rychle jednotlivé funkce „rostou“. Pokládáme si vlastně následující otázku: „Zvětšíme-li dvakrát vstup, kolikrát více času zabere výpočet?“ Vzhledem k tomu, že operace našeho programu nejsou elementární, můžeme počítat, že lineární algoritmus ($\mathcal{O}(n)$) stihne dokončit výpočet za několik vteřin pro vstupy do velikosti desítek milionů. Od toho se potom odvíjí, že o kvadratický algoritmus ($\mathcal{O}(n^2)$) má smysl se pokoušet pouze tehdy, pokud omezení na vstupní data jsou do několika málo tisíc, a podobně o kubický ($\mathcal{O}(n^3)$) tam, kde velikost vstupu je jen několik stovek. Dále algoritmus se složitostí $\mathcal{O}(2^n)$ uspěje na soutěži nejvýše pro $n \leq 25$, zatímco $\mathcal{O}(n!)$ pouze pro n do 11.

Konec teorie, jdeme na úlohy!

Následují stručné verze zadání některých úloh, které se vyskytly na zmiňovaném světovém finále. Často chybí méně podstatné detaily a toto zadání se snaží popsat pouze klíčové části úlohy. Podrobnosti o úlohách naleznete v originálním zadání na oficiálních stránkách soutěže.

Úloha 1. Artyčoky je možné na trhu nakoupit a prodat kterýkoliv z n dnů číslovaných od 1 do n. Cena artyčoků na trhu v den k je rovna

$$\text{cena}(k) = p \cdot (\sin(a \cdot k + b) + \cos(c \cdot k + d) + 2).$$

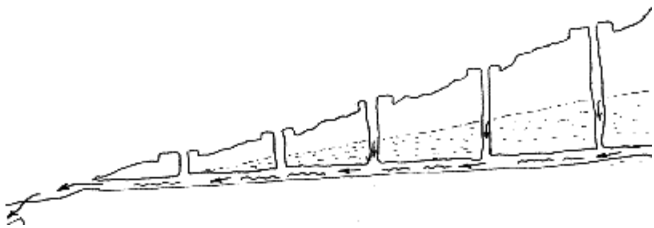
Určete největší pokles ceny mezi dvěma dny, kdy je možné artyčoky nakoupit (tj. maximální hodnotu $\text{cena}(i) - \text{cena}(j)$ pro $1 \leq i \leq j \leq n$). Vstup se skládá z šesti celých čísel p $(1 \leq p \leq 1\,000)$, a, b, c, d $(0 \leq a, b, c, d \leq 1\,000)$ a n $(1 \leq n \leq 10^6)$. Výstup programu bude považován za správný, bude-li mít relativní nebo absolutní chybu nejvýše 10^{-6}.

Úloha 2. Na vstupu jsou dány dva konvexní mnohoúhelníky, které se rovnoměrně pohybují v rovině (nerotují a pohybují se po přímce). Známé jsou jak jejich počáteční pozice, tak rychlosti a směry. Určete čas, kdy plocha jejich překryvu bude největší. Nepotkají-li se nikdy, vypište *nikdy*. Počet vrcholů každého mnohoúhelníku je nejvýše 10.

Úloha 3. Sýr má tvar krychle o rozměrech $100 \times 100 \times 100$. Vstup začíná dvěma celými čísly n, s $(0 \leq n \leq 10\,000,\ 1 \leq s \leq 100)$. Dalších n řádků popisuje jednotlivé díry v sýru tvaru koule souřadnicemi jejich středů a jejich poloměry. Všechny díry jsou celé uvnitř sýra a navzájem se neprotínají. Rozkrájejte sýr na s plátků rovnoběžných s osou z tak, aby všechny plátky měly stejný objem.

Úloha 4. Každý organizmus na nově objevené planetě lze popsat jeho genetickou informací tvořenou konečnou posloupností elementárních nukleotidů ‘A’, ‘C’ a ‘M’. Na počátku věků žily na planetě dva jednoduché organizmy, každý složený z jediného nukleotidu. Občas se do genetické informace jednoho organizmu přimíchá (na libovolnou pozici) další nukleotid a nově vzniklý organizmus svého předchůdce okamžitě zahubí. Nyní žije na planetě pouze jediný druh organizmů (tj. obě vývojové větve se sloučily), jehož genetická informace je dána na vstupu. Dále jsou dané genetické informace organizmů odhalené z vykopávek. Rozhodněte, zda je možné, aby se na zkoumané planetě organizmy vyvíjely podle zmíněného modelu. Vykopávek na vstupu je nejvíce 4 000 a každá z nich obsahuje nejvíce 4 000 nukleotidů.

Úloha 5. *Kanát* je umělý podzemní kanál, jímž se do pouštních oblastí přivádí samospádem voda z hor. Tento dávný lidský vynález umožnil osídlení pouštních oblastí v Africe a v Asii. Kanát se skládá z jednoho hlavního vodorovného kanálu a několika svislých šachet (viz obrázek) vyhloubených do nakloněné roviny. Kanál i všechny šachty mají stejný konstantní průřez. Při stavbě kanátu je nesnadným problémem odvoz vykopané horniny, která musí být odstraněna právě budovanými šachtami. Je dán požadovaný tvar kanátu, navrhněte rozmístění n $(1 \leq n \leq 1\,000)$ pomocných svislých šachet (to jsou všechny kromě té nejvyšší a slouží především k usnadnění stavby) tak, aby celková práce potřebná ke stavbě kanátu byla co nejmenší. Chceme-li souvislý úsek horniny délky L posunout k jednomu jeho konci, musíme vynaložit $\frac{L^2}{2}$ práce.



Úloha 6. Řeka se skládá z n $(1 \leq n \leq 10^5)$ rovnoběžných a stejně širokých dopravních pruhů. V každém pruhu se lodě pohybují buď po proudu, nebo proti proudu. Všechny lodě plují stejnou konstantní rychlostí. Na vstupu jsou dány startovní pozice všech plavidel, jejich směry a délky. Celkový počet lodí je nejvýše 10^5. Z dané pozice jednoho břehu řeky bychom chtěli vypravit přívoz známé rychlosti, který by kolmo přeplul na druhou stranu tak, aby po cestě nenaboural do žádné lodě. Hledáme nejdelší souvislý úsek mezi časy t_1 a t_2, kdy je možné přívoz bezpečně vypravit.

Úloha 7. Je známá pravděpodobnost, že nějaký den bude slunečno, že bude oblačno, že bude pršet a že budou padat trakaře. Každý den nastane právě jedna z těchto možností. Předpokládáme, že počasí v jednotlivé dny je nezávislé na ostatních dnech. Z naší meteorologické stanice bychom chtěli výsledky za n $(1 \leq n \leq 20)$ dní posílat na centrálu pomocí posloupnosti nul a jedniček. Navíc bychom chtěli, aby žádná z možných reportáží nebyla prefixem jiné (neboli počátečním úsekem), aby bylo možné určit, že již reportáž skončila. Navrhněte protokol posílání reportů o počasí, který minimalizuje střední délky zprávy.

Odkazy

[1] *https://icpc.baylor.edu/* – oficiální stránky soutěže
[2] *https://icpc.baylor.edu/worldfinals/problems/icpc2015.pdf* – zadání úloh
[3] *https://en.wikipedia.org/wiki/ACM_ICPC* – podrobnosti o soutěži
[4] *http://codeforces.com/blog/entry/18016* – videa s řešeními úloh
[5] *http://www.csc.kth.se/~austrin/icpc/finals2015solutions.pdf* – řešení úloh