

Soutěž v programování 2023

Krajské kolo Praha

Zhodnocení úloh

Vážení účastníci,

nejpopulárnější úlohou tohoto krajského kola byla v Praze kalkulačka. Naopak v úloze 3 jsme hodnotili pouze 8 řešení. Z hlediska použitých jazyků byl letos nepoužívanější Python, ale mnoho účastníků pracovalo v C#, Javě, a nebo Typescriptu/Javascriptu. Jeden účastník použil jazyk Julia.

Zde naleznete náš rozbor řešení úloh a nejčastějších chyb.

Děkujeme za účast v soutěži, gratulujeme postupujícím, a doufáme, že se s některými z vás setkáme i napřesrok.

Adam Benda, Martin Horský, David Šertler - porotci
9.května 2023

Referenční řešení naleznete na webu programovacisoutez.cz

Úloha 1: Bludiště

Úkolem úlohy je v bludišti nalézt všechna dosažitelná slova. Poté je vypsat správně obrácená (určuje velké písmenko na začátku) a abecedně seřazená.

Nejčastější chyby

- Nerespektování stěn bludiště => vrácení i nedosažitelných slov
- Přesmyčka / rozdělení slova, na které průchod bludištěm narazí v jeho prostředku.
- Chybějící (abecední) seřazení slov na výstupu (pozorně čtete zadání!)

Úloha 2: Kalkulačka

Úkolem je vytvořit jednoduchou kalkulačku se zadáním čísel, operacemi sčítání, odčítání, násobení a dělení. Rozšíření by mělo fungovat pro zlomky, zadané se znakem pípy, například 5|3

Nejčastější chyby

- Kalkulačka by měla fungovat pro celá čísla, desetinná čísla (floaty) a zlomky

- Dle zadání desetinná čísla obsahují desetinnou čárku a ne tečku. Sice by to mělo být nastavitelné pomocí locale (místní a jazyková nastavení), v praxi však mnoho jazyků a knihoven toto nastavení nerespektuje, a používá desetinnou tečku. Nejjistější je tedy výměna čárky za tečku před napařováním čísla (float)
- Napařování čísla pomocí funkce `eval` může vypadat pohodlně, ale přináší s sebou neskutečné množství problémů a bezpečnostních hrozeb - nedělejte to. Použijte správnou funkci, parsující řetězec na float:

Python	TS/Javascript	C#	Java
<code>float (vstup)</code>	<code>parseFloat (vstup)</code>	<code>Single.Parse (vstup)</code>	<code>Float.parseFloat (vstup)</code>

Úloha 3: Mini Puzzle

Vstupem programu je obrázek rozdělený na čtvrtiny (přesné). Cílem je poskládat zpětně ze čtvrtin obrázek - nemáte však informaci, která čtvrtina je ve výsledném obrázku na kterém místě.

Pro řešení úlohy je třeba být kreativní. Pokud byste úlohu s rozstříhanými čtvrtinami řešili vy, jako člověk, jak byste postupovali? Jedna z možností je samozřejmě zaměření se na objekty, které přesahují z jedné části do druhé (například vodopád), a jejich správné sestavení. Rozpoznávání objektů je však komplikovaná úloha (kterou většinou řeší neurální sítě trénované na rozsáhlých datasetech příkladů), a zcela mimo časové možnosti soutěže.

Druhou, a realizovatelnější možností, je sledování švů, neboli okrajů. Využijeme toho, že u většiny obrázků na sebe dvě vedlejší pixelové sloupce (či řádky) navazují. Znamená to, že dva vedlejší pixely budou mít nejspíše stejnou, či podobnou barvu.

Jak zjistit podobnost barev: Pokud bychom měli obrázek ve stupních šedi od 0 do 255, byla by tato podúloha snadná. $\text{rozdilnost}(\text{pixelA}, \text{pixelB}) = \text{Math.abs}(\text{pixelA} - \text{pixelB})$ - rozdíl 0 znamená, že jsou barvy totožné, 10 že se liší o maličko, 255 je pak maximální rozdíl mezi bílou a černou.

Nejspíš ale víte, že barva pixelu je určena systémem RGB - intenzita červeného, zeleného a modrého světla, zpravidla zapsaná jako tři čísla 0-255 (bajt). To naši úlohu podobnosti barev znesnadňuje. Nejjednodušší vzoreček je

$\text{Math.abs}(\text{pixelA}.\text{RED}-\text{pixelB}.\text{RED})+\text{Math.abs}(\text{pixelA}.\text{GREEN}-\text{pixelB}.\text{GREEN})+\text{Math.abs}(\text{pixelA}.\text{BLUE}-\text{pixelB}.\text{BLUE})$ (prostě tři rozdílnosti v barevných kanálech sečtu). Nebo lze použít [převod barev do stupňů šedi](#).

Když umíte vypočítat rozdíl dvou pixelů, můžete tyto rozdíly vypočítat pro celý krajní sloupec/řádku a výsledky zprůměrovat. Tento postup uděláte pro všechny možné kombinace přilehlých sloupců/řádků, a zvolíte takovou přilehlost čtvrtin, aby průměr či součet rozdílností ve všech švech byl nejmenší možný.

Úloha 4: Easy Tetris

Potěšilo nás, kolik jsme dostali pěkných řešení úlohy. Znamená to, že se dobře orientujete v soustavě souřadné - a her ve čtvercové mřížce je spousta.

Největší výzvou bylo v této úloze ošetřit všemožné krajní případy - aby kostička nešla vystrčit ven z herní plochy, ale ani se nemohla při pohybu doleva-doprava zanořit do sloupce již umístěných čtverečků po straně. Stejně tak možnost rotace dlouhé modré | kostičky skrývala řadu nástrah. Viděli jsme i řešení, ve které se kostička pomocí rotování dala udržet navždy levitující.