

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**  
**Fakulta informatiky a informačných technológií**

Projekt 2:  
**Vizualizácia counting-sort algoritmu**

Matej Skyčák a Tomáš Socha

Akademický rok 2021/2022

## 1. Princíp fungovania counting-sort algoritmu

Princíp fungovania counting-sort algoritmu je veľmi priamočiary a skladá sa so 4 krokov. Tento algoritmus sa používa na usporiadanie vstupného poľa celých kladných čísel a jeho princíp v skratke spočíva v tom, že spočíta výskyt (vyplýva z názvu counting-sort) jednotlivých prvkov v množine, vypočíta pre každé číslo index, na ktorom sa bude v novo vytvorenom poli začínať a následne ich už len postupne do novo vytvoreného nového poľa priradí, pričom po každom priradení konkrétneho čísla, dekrementuje jemu prislúchajúci index.

## 2. Detailný popis funkcionality programu

Vstupom algoritmu je teda nejaká **neusporiadaná množina celých kladných čísel** a číslo **k**, ktoré je **maximálny prvok** v danej množine a výstupom je usporiadaná množina čísel od najmenšieho po najväčšie. V prvom kroku sa vytvorí pole dĺžky **k** a jeho prvky sa inicializujú na nulu, toto pole slúži na zistenie **početnosti výskytov** jednotlivých čísel vo vstupnom poli. V druhom kroku sa postupne zľava doprava prechádza vstupné neusporiadané pole a pre každý prvok sa inkrementuje hodnota na príslušnom mieste v poli vytvorenom v predošlom kroku. Takýmto spôsobom sa prejde celé vstupné pole a nasleduje tretí krok. V tomto kroku sa prejde pole s počtami výskytov prvkov a postupne zľava doprava, počnúc druhým prvkom sa ku každému prvku pripočíta hodnota prvku naľavo od neho. Týmto krokom získame **indexy, na ktorých majú jednotlivé čísla začínať**. V poslednom štvrtom kroku si najprv vytvoríme ďalšie pole veľkosti **k**, toto už bude výsledné usporiadané pole, výstup algoritmu. Následne sa prechádza vstupné pole s číslami. Pre každé sa v poli indexov pozrie, na ktorom indexe sa má vo výslednom poli nachádzať. Na tento index ho vloží a index v poli indexov dekrementuje. Týmto spôsobom prejde celé vstupné neusporiadané pole a vytvorí výstupné usporiadané pole.

## 3. Používateľské rozhranie, systém ovládania

Ovládanie nášho programu je prosté a intuitívne, pred spustením programu má používateľ možnosť si zadať množinu náhodných celých čísel, na ktorej chce, aby sa triediaci algoritmus vykonával. Túto funkcionality poskytujú dva „slider-y“. Pomocou prvého si vie zvoliť veľkosť vstupnej množiny, ktorá je obmedzená na 4 až 15 prvkov a to z dôvodu aby sa algoritmus nevykonával príliš dlho a aby pomocné polia neboli príliš veľké, keďže slúži skôr ako učebná pomôcka. Druhý „slider“ slúži na zvolenie maximálneho čísla vo vstupnej množine, minimálna a maximálna hodnota je taktiež obmedzená. Pri posunutí „sliderov“ sa vždy vstupná množina aktualizuje, takže používateľ vidí, čo bude vstupom algoritmu.

Program sa spustí stlačením tlačidla „vybrať množinu“ a začne algoritmus, popísaný v predošlej kapitole. Program aj vizualizuje každý krok algoritmu, napríklad zväčšením konkrétneho čísla vo vstupnej množine alebo farebným zvýraznením.

## 4. Zdôvodnenie výberu a jeho prínos

Tento problém sme si vybrali, lebo sme chceli nejaký projekt, kde sa dá zvolený problém pekne vizualizovať, príkladom takýchto problémov sú napríklad grafové algoritmy alebo triediace algoritmy. Pri triediacich algoritmoch sme mali predstavu ako by sme ich chceli, krok po kroku, vizualizovať, aby sa z nášho projektu dal aj pochopiť. Preto sme si vybrali vizuálnu demonštráciu triediaceho algoritmu, ktorý sme už preberali na predmete DSA

minulý semester, a preto tá najväčšia výzva bola ho naprogramovať aj s vizualizáciou v programovacom jazyku Mathematica, s ktorým sme predošlé skúsenosti nemali.

Cieľ nášho projektu bol vizualizovať triediaci algoritmus counting-sort krok po kroku, tak aby sa z neho dalo algoritmus jednoducho pochopiť. Keďže sa jednotlivé políčka, s ktorými aktuálne program pracuje aj farebne zvýrazňujú, dal by sa napríklad použiť ako učebná pomôcka pre študentov, ktorí sa o triediacich algoritmoch práve učia. Reálne využitie by mohlo byť v už spomínanom predmete DSA (dátové štruktúry a algoritmy).

## 5. Záver

Keďže ako sme spomínali, že sme nemali žiadne predošlé skúsenosti s písaním komplexnejších programov v Mathematice, okrem jednoriadkových príkazov na cvičeniach, ktoré slúžili skôr na pochopenie princípov toho ako Mathematica funguje, bola to spočiatku celkom veľká výzva. Po dokončení projektu sme sa určite veľa naučili, a aj zopakovali vedomosti z iného predmetu. Naša pôvodná predstava sa nám celkom aj podarila a s výsledkom sme spokojný. Taktiež by sme chceli ešte spomenúť, že program nemusí fungovať úplne ideálne na online verzií Mathematici, kde sme mali problém s vizualizáciou. Ale fungovanie algoritmu to nijako neovplyvnilo.