

# String art

Michal, Juli, Jáchym

30. listopadu 2024

## 1 Co je to string-art?

String-art je umělecká forma kdy se snažíme přiblížit nějakému obrázku pomocí natahování nití v kruhu. Protože string-art je činnost veskrze praktická máme k dispozici pouze konečně mnoho špendlíků ( $n$ ), které se nacházejí po obvodu.

Naším cílem je mezi nimi natáhnout celkem  $k$  nitek<sup>1</sup> tak aby se výsledek co nejlépe podobal nějakému předem vybranému obrázku. Na vyřešení této úlohy jsme zpracovali dva algoritmy.

## 2 Algoritmy

Algoritmy v této sekci představíme a porovnáme jejich výhody a nevýhody. Zároveň představíme jejich asymptotickou časovou složitost s pomocí notace  $O$ . Jestliže jste s touto notací neseznámeni doporučujeme si o ní něco přečíst.

### 2.1 Všechny možnosti!

Asi jednodušší myšlenkou která nás napadne, chceme-li řešit takovýto problém je: zkusím všechny možné čáry a vezmu jen ty nejlepší. Touto myšlenkou se také řídil náš první algoritmus. Aby to nebylo tak jednoduché musíme si nějak chytře definovat jak dobrá je každá čára.

Tento problém jsme řešili tak, že na prázdné plátno nakreslíme jen tu čáru samotnou a poté celý tento obrázek odečteme od toho původního. Díky tomu, že zatím řešíme pouze černobílé obrázky, stačí odečíst každý pixel z jednoho obrázku s korespondujícím pixelem z toho druhého. Všechny tyto rozdíly poté sečteme, umocníme a vydělíme počtem pixelů.<sup>2</sup> Této hodnotě dále budeme říkat rozdíl čáry a originálu.

Následně všechny čáry podle této hodnoty seřadíme a vybereme  $k$  nejlepších.

Teď už by to mohlo vypadat, že jsme hotoví a můžeme si užívat string-artu jak jen budeme chtít. To by se nesměl v našem dosavadním řešení skrývat jeden menší problém. Protože každou čáru zkoušíme právě jednou může, nebo spíše měla by, být úplně černá. To znamená, že v každém bodě buď je nebo není čára, což nám ve výsledku vytvoří velmi nepřírozené obrázky.

Abychom tohle vyřešili musíme nějak zařadit aby mezi dvěma špendlíky mohlo vést více než jedna čára. Toho bychom mohli dosáhnout triviálním rozšířením toho algoritmu. Prostě před přidáním každé čáry seřadíme všech  $n^2$  čar. To ovšem z časové složitosti  $O(n^2)$  udělá  $O(kn^2)$ . Čímž jsme si docela dost pohoršili, protože  $k \gg n$ . Naštěstí existuje i lepší způsob, ten budiž nalezen v následující podsekci.

### 2.2 Postupně zlepšujeme

Abychom napravili obě chyby z předchozího řešení musíme celou předchozí ideu překopat. Místo toho bychom v každém kroku počítali všechny možné dvojice, začneme pouze s jedním vrcholem a spočítáme všechny čáry z něho vedoucí. Z těch vybereme tu nejlepší a její konec nastavíme jako začátek a spustíme celý proces znovu.

---

<sup>1</sup>V průběhu přeformulujeme naši úlohu tak aby namísto konstantního počtu nitek byla konstantní odchylka od originálu, které se snažíme dosáhnout.

<sup>2</sup>Této metodě se v praxi říká MSE (Mean Square Error) a pro její používanost jsme rozhodli pro ni.

```

1 def string_art(n,k)
2     lines = [(start,end) for start in range(n)] for end in range(n)]
3     lines.sort(key= lambda line: difference(line))
4     return lines[k:]

```

Obrázek 1: První algoritmus ve zjednodušené verzi

```

1 def string_art(n,k)
2     final = [0]
3     while len(final)<=k:
4         start = final[-1]
5         lines = [(start,end) for end in range(n)]
6         lines.sort(key= lambda line: difference(line))
7         final.append(lines[0][1])
8     return final

```

Obrázek 2: Druhý algoritmus ve zjednodušené verzi.

Tento proces opakujeme  $k$ -krát a efektivně jsme vyřešili oba problémy minulého algoritmu. Ovšem vyskytnul se nám jeden další. Jestliže bude na obrázku jeden velmi černý pruh, algoritmus bude dávat čáry jen do něj a nikam jinam.

Řešení tohoto je jednoduché: po přidání každé čáry do našeho string-artu stejnou čáru ale bílou, nakreslíme do původního obrázku.

Tento algoritmus má časovou složitost  $O(kn)$ . V praxi se nám vyplatilo nemít žádný pevný počet čar  $k$ , ale radši se koukat jak daleko je celý string-art od originálu. Jakmile se tento rozdíl dostane pod nějakou konstantu tak celý program zastavíme. Důvod proč jsme neuvažovali tuto variantu je kvůli její nejasné časové složitosti.

### 3 Výsledky

S programy představenými v předchozích kapitolách se nám povedlo převést spoustu obrázku do string-artové podoby. Mezi ty nejzajímavější patří spící Dláža. Také se nám povedlo udělat strig-art video. To jsme stvořili tak, že jsme do string-artu převedli každý snímek zvlášť a poté je zas spojili. Mezi poslední zajímavosti patří barevný string-art. Toho jsme docílili tak, že jsme každý z barevných kanálů převedli samostatně na stri-art variantu a poté všechny tři jednobarevné stri-arty spojily dohromady.