

Vizualizácia algoritmov pre Wheelerove grafy

Adam Struharňanský

Školiteľ: Mgr. Adrián Goga

Vizualizované algoritmy a štruktúry

- Sufixové pole
- Burrows-Wheelerova transformácia (BWT)
- Inverzná Burrows-Wheelerova transformácia
- Výpočet BWT pomocou sufixového pola
- Hľadanie vzoru pomocou BWT (zjednodušený FM-index)
- Wheelerove grafy
- Wheelerov graf reprezentujúci BWT
- Tunelovanie Wheelerovho grafu reprezentujúceho BWT
- Získanie pôvodného reťazca zo stunelovaného Wheelerovho grafu

Sufixové pole a Burrows-Wheelerova transformácia

i	LCP[i]	SA[i]	S[SA[i], n - 1]
0		9	\$
1	0	6	asy\$
2	3	1	asypeasy\$
3	0	5	easy\$
4	4	0	easypeasy\$
5	0	4	peasy\$
6	0	7	sy\$
7	2	2	sypeasy\$
8	0	8	y\$
9	1	3	ypeasy\$

Sufixové pole SA

F									L
\$	e	a	s	y	p	e	a	s	y
a	s	y	\$	e	a	s	y	p	e
a	s	y	p	e	a	s	y	\$	e
e	a	s	y	\$	e	a	s	y	p
e	a	s	y	p	e	a	s	y	\$
p	e	a	s	y	\$	e	a	s	y
s	y	\$	e	a	s	y	p	e	a
s	y	p	e	a	s	y	\$	e	a
y	\$	e	a	s	y	p	e	a	s
y	p	e	a	s	y	\$	e	a	s

BWT-matica, stĺpec L označuje BWT

Inverzná Burrows-Wheelerova transformácia

- i -ty znak v L cyklicky predchádza i -temu znaku F
- i -ty výskyt znaku c v reťazci L odpovedá i -temu výskytu znaku c v reťazci F

c	$C[c]$	i	$LF[i]$	$F[i]$	$L[i]$	$S[i]$
\$	1	0	8	$\$0$	$y0$	
a	3	1	3	$a0$	$e0$	
e	5	2	4	$a1$	$e1$	
p	6	3	5	$e0$	$p0$	
s	8	4	0	$e1$	$\$0$	
y	10	5	9	$p0$	$y1$	
		6	1	$s0$	$a0$	a
		7	2	$s1$	$a1$	s
		8	6	$y0$	$s0$	y
		9	7	$y1$	$s1$	\$

Zmena pozície pomocou LF-mapovania

Vyhľadávanie vzoru v reťazci pomocou BWT

- FM-index
- V našej implementácii sme použili celé sufixové pole, v reálnom použití je uložená iba časť (sufixové pole je pamäťovo náročné)

	F								L	i	S[i]	SA[i]	
0	\$ ₀	e	a	s	y	p	e	a	s	y ₀	0	e	9
1	a ₀	s	y	\$	e	a	s	y	p	e ₀	1	a	6
2	a ₁	s	y	p	e	a	s	y	\$	e ₁	2	s	1
3	e ₀	a	s	y	\$	e	a	s	y	p ₀	3	y	5
4	e ₁	a	s	y	p	e	a	s	y	\$ ₀	4	p	0
5	p ₀	e	a	s	y	\$	e	a	s	y ₁	5	e	4
6	s ₀	y	\$	e	a	s	y	p	e	a ₀	6	a	7
7	s ₁	y	p	e	a	s	y	\$	e	a ₁	7	s	2
8	y ₀	\$	e	a	s	y	p	e	a	s ₀	8	y	8
9	y ₁	p	e	a	s	y	\$	e	a	s ₁	9	\$	3

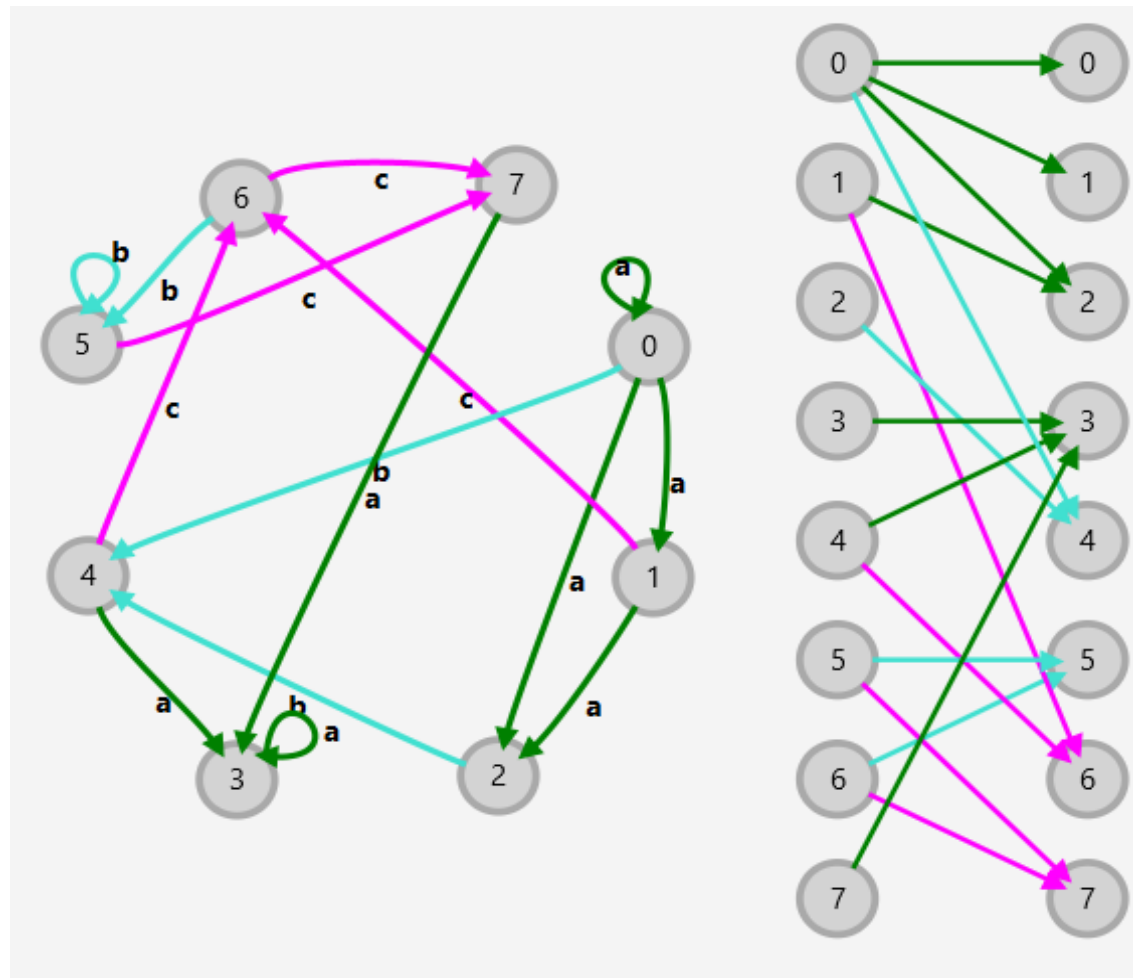
Zvýraznenie sufixov vzoru „easy“ v BWT-matici

Wheelerove grafy

- Každý vrchol má vstupnú aj výstupnú hranu
- Pre každú dvojicu hrán $e = (u, v)$ a $e' = (u', v')$, označené pomocou funkcie λ platí:

(i) $\lambda(e) < \lambda(e') \implies v \leq v'$

(ii) $\lambda(e) = \lambda(e') \wedge u < u' \implies v \leq v'$



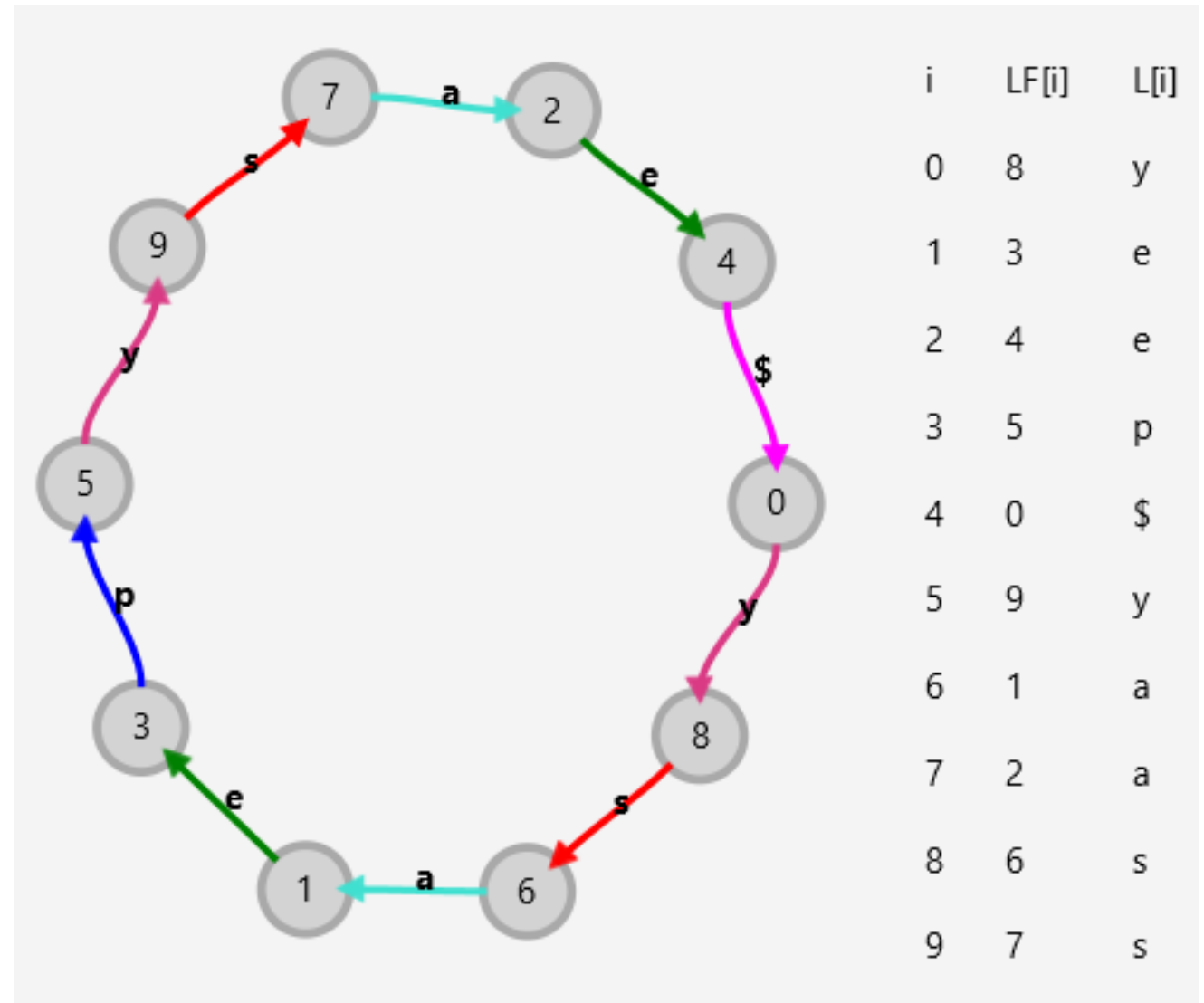
Príklad Wheelerovho grafu

i	L[i]	out[i]	in[i]	F[i]
0	b	1	1	a
1	a	0	1	a
2	a	0	1	a
3	a	0	0	a
4	c	1	1	a
5	a	0	0	a
6	b	1	0	a
7	a	1	1	b
8	c	1	0	b
9	a	0	1	b
10	c	1	0	b
11	b	0	1	c
12	c	1	0	c
13	b	0	1	c
14	a	1	0	c
		1	1	

Úsporná reprezentácia

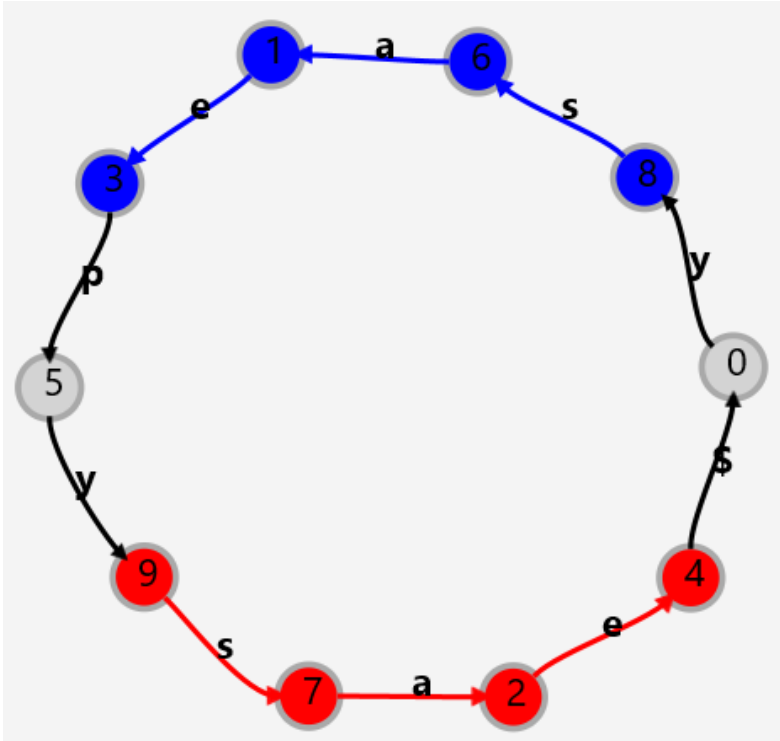
Wheelerov graf reprezentujúci BWT

- Hrany sú $(i, LF[i])$ s označením $L[i]$

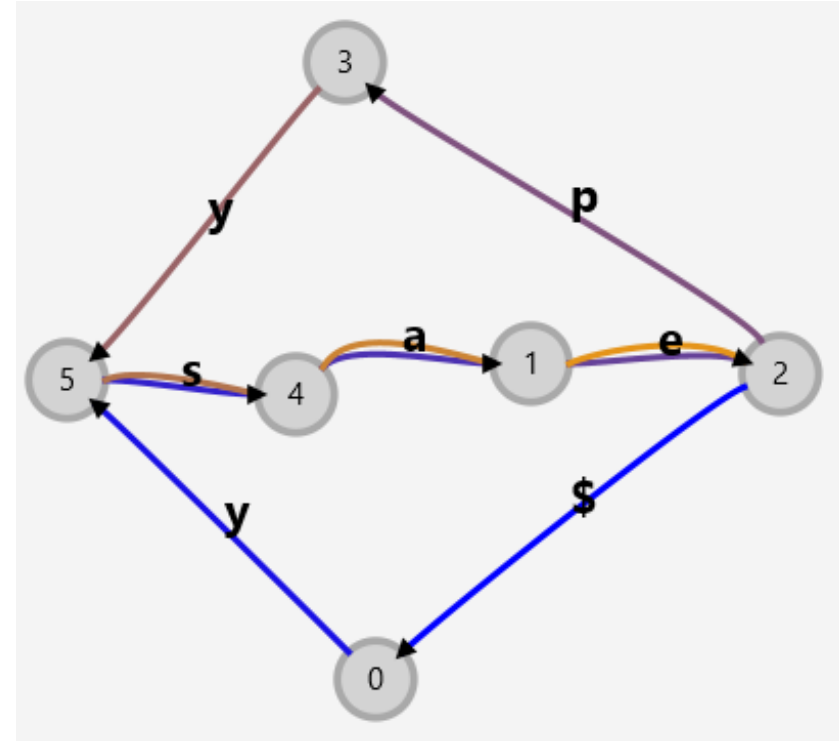


Príklad Wheelerovho grafu reprezentujúceho BWT

Tunelovanie Wheelerových grafov

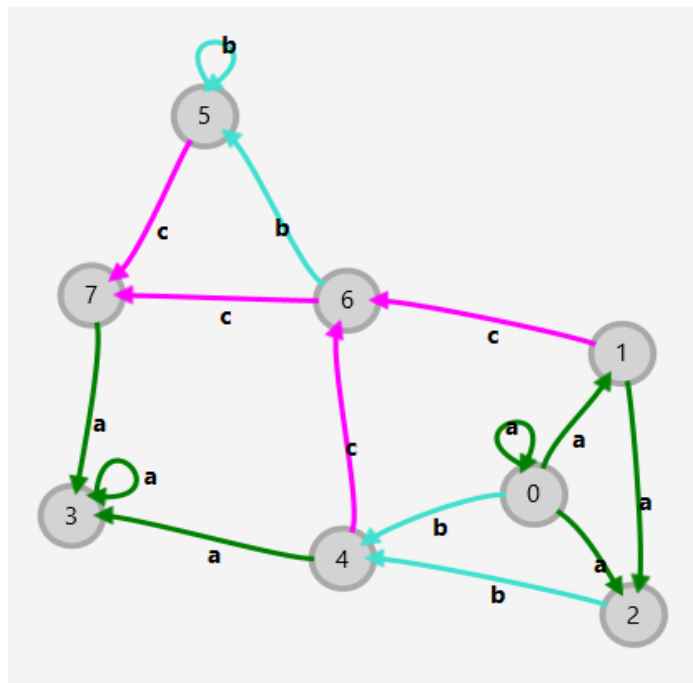


Wheelerov graf s označením prefixových intervalov

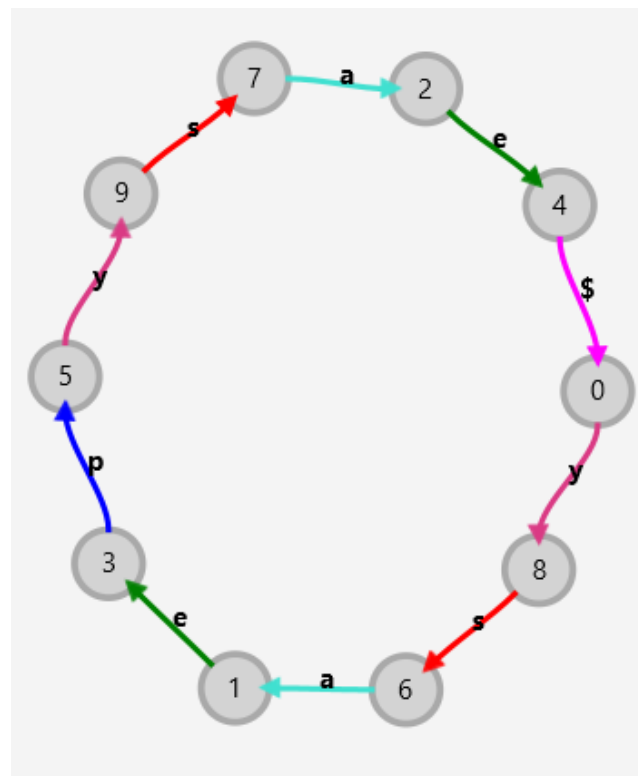


Pochôdzka v stunelovanom grafe

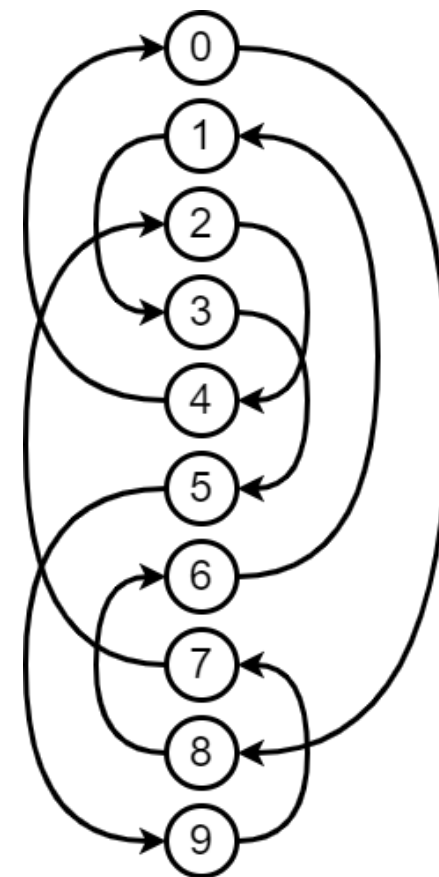
Vykreslenie grafu



Graf po použití funkcie *beautify()*



Sufixové pole SA



Algorithmy Jazyk

Získanie reťazcu

Graf

Pamät'

Pseudokód

Popis

i	L[i]	out[i]	in[i]	F[i]	SR[i]
0	y	1	1	\$	\$
1	e	1	1	a	y
2	p	1	1	e	
3	\$	0	1	p	
4	y	1	1	s	
5	a	1	1	y	
6	s	1	0	y	
7		1	1		

```

1 i = L.select($, 0)
2 offset = 0
3 for(int k = 0; k < n; k++)
4   SR[k] = L[i]
5   i = L.C(L[i]) + L.rank(L[i], i)
6   nodeRank = in.rank(1, i + 1) - 1
7   if(in[i] = 0 || in[i + 1] = 0)
8     offset = i - in.select(1, nodeRank)
9   i = out.select(1, nodeRank)
10  if(out[i + 1] = 0)
11    i = i + offset
12  offset = 0

```

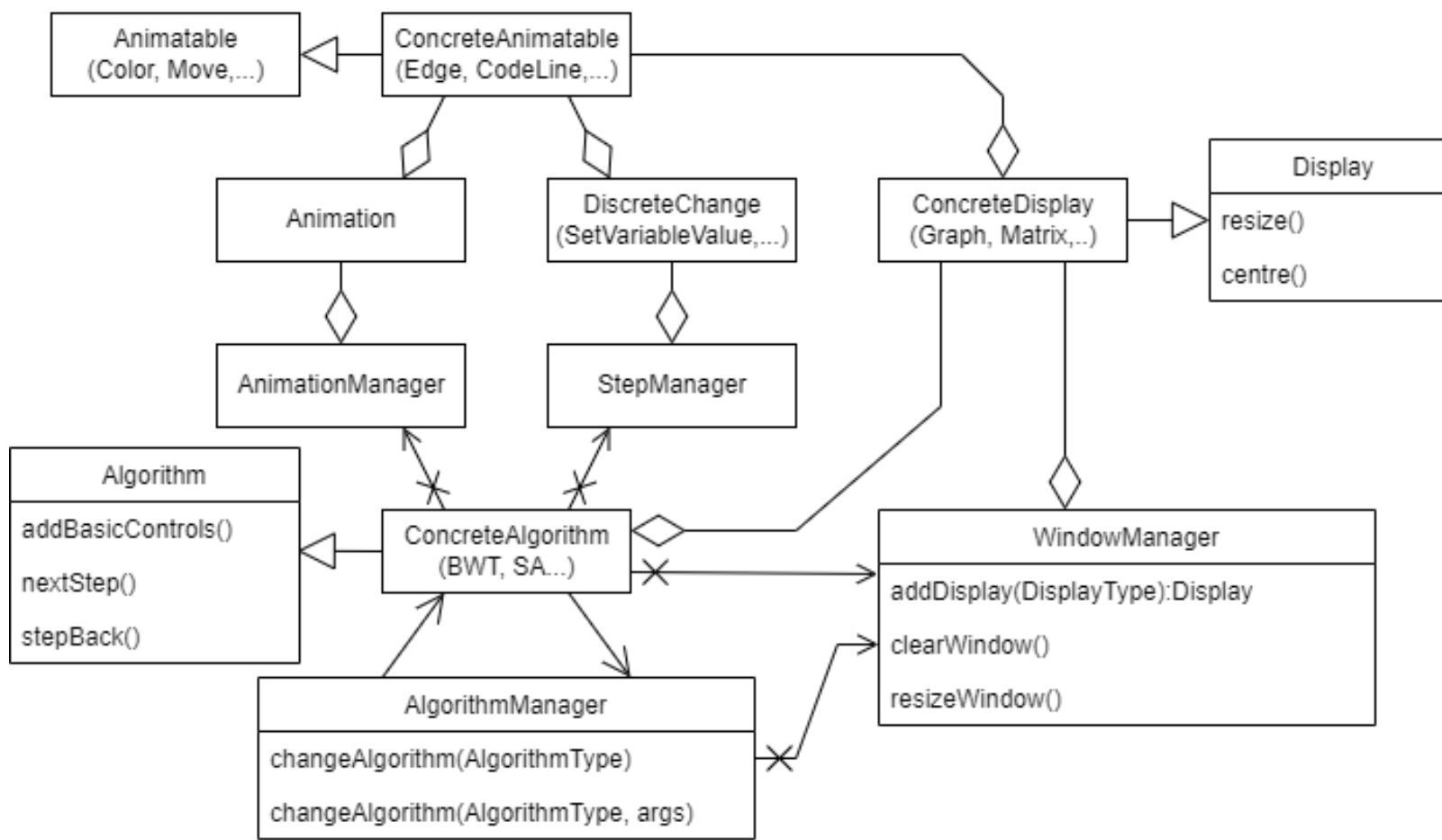
i = 5 offset = 0 k = 1 nodeRank = 5

Keďže vrchol, do ktorého vstupujeme má viac vstupných hrán, znamená to, že sme narazili na vstup do tunela

Krok späť Animovať Rýchlosť animácie

Okno programu pri vizualizácii algoritmu

Približný diagram tried programu



Budúca práca na projekte

- Pridať viacnásobné tunelovanie a získanie reťazca z viacnásobne tunelovaného grafu
- Pridať vizualizáciu platnosti LF-mapovania
- Pridať nové vykreslenie grafov
- Spojiť „animovateľné“ a „diskrétne“ zmeny
- Opraviť „bugy“, refaktorovať kód
- V dlhšej budúcnosti pridať ďalšie algoritmy na prácu s reťazcami

Ďakujem za pozornosť