Vyhledávání K nejbližších sousedů na základě filtru

Search K nearest neighbors based on a filter

Bc. Jan Jedlička

Vedoucí: Doc. Ing. Radim Bača, Ph.D.

FEI, VŠB-TUO

2022

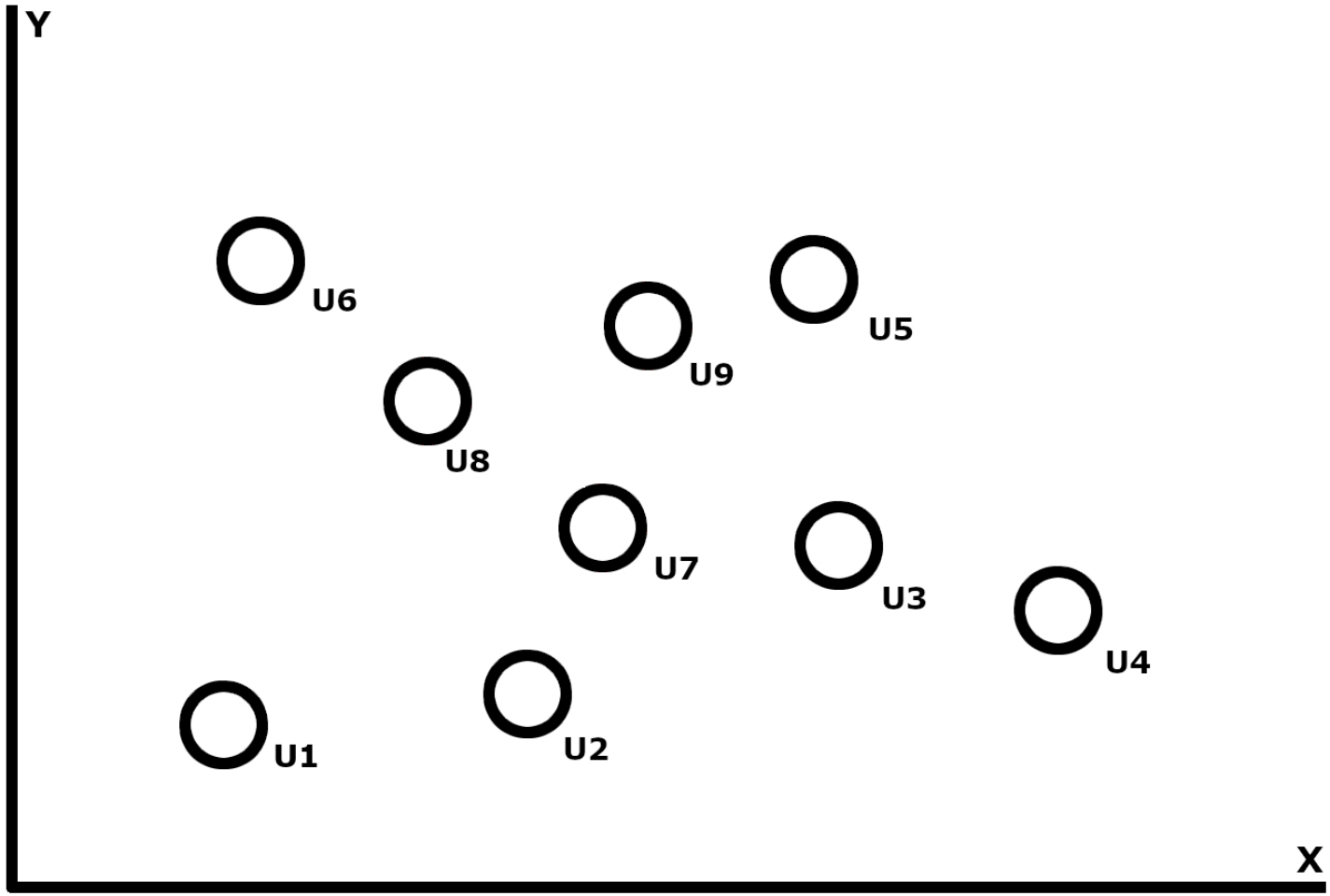
Úvod

- Porozumění HNSW
- Vlastní HNSW implementace nebo zprovoznění jiné HNSW implementace
- Návrh a implementace rozšíření HNSW o filtr (podmínka, která stanoví, které vektory se při prohledávání vynechají)

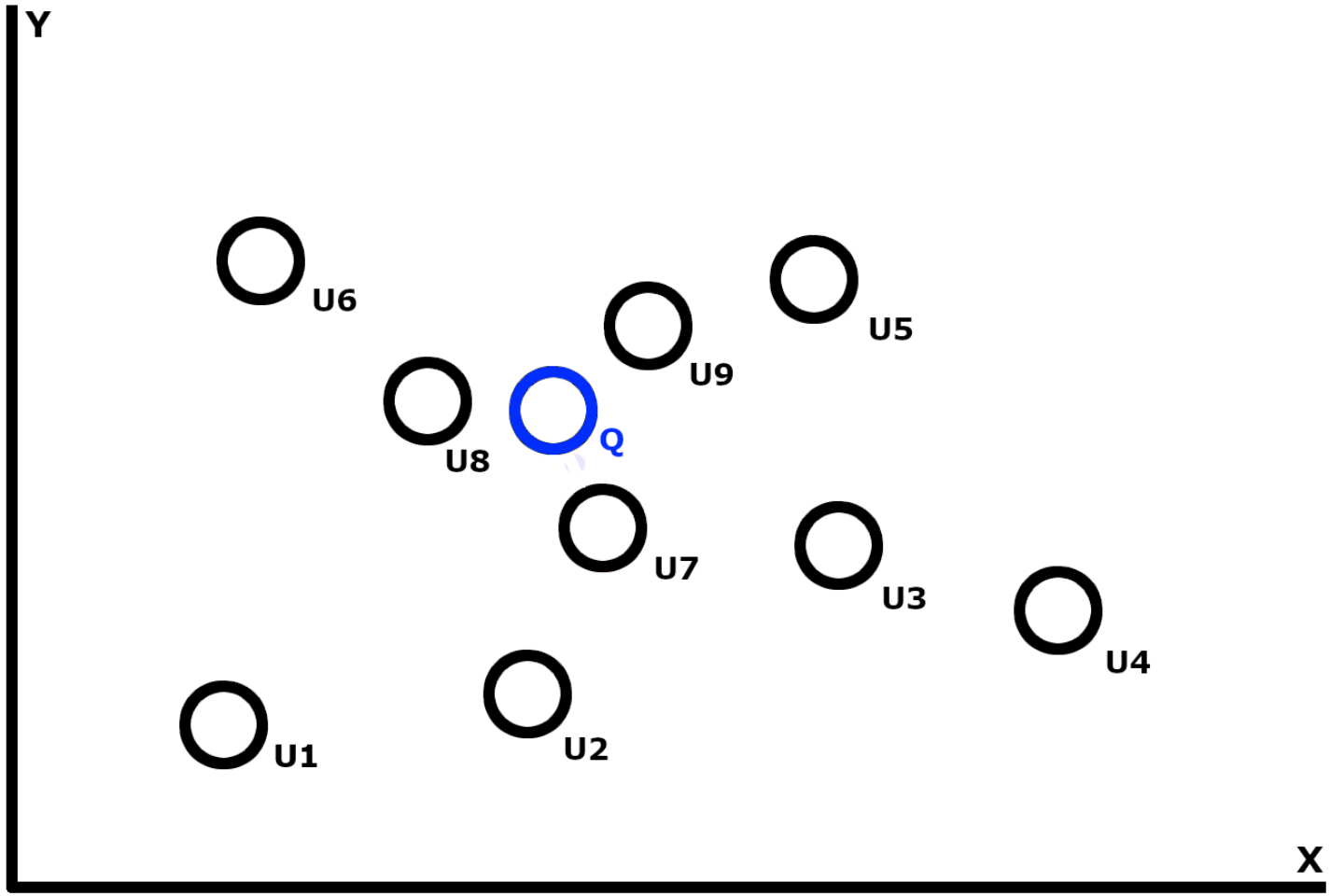
KNN

- Vyhledávání K nejbližších sousedů od dotazu Q v n dimenzionálním prostoru
- Vzdálenost mezi body v prostoru definována metrikou (Euklidova, Hammingova, Minkowského atd.)
- U velkých dimenzí je pro většinu technik rychlejší sekvenční průchod
- Přibližné vyhledávání (ANN)
- Porovnávání vektorizovaných dat, hledání shluků, podobných vlastností (například vyhledávání sémanticky podobných dokumentů)

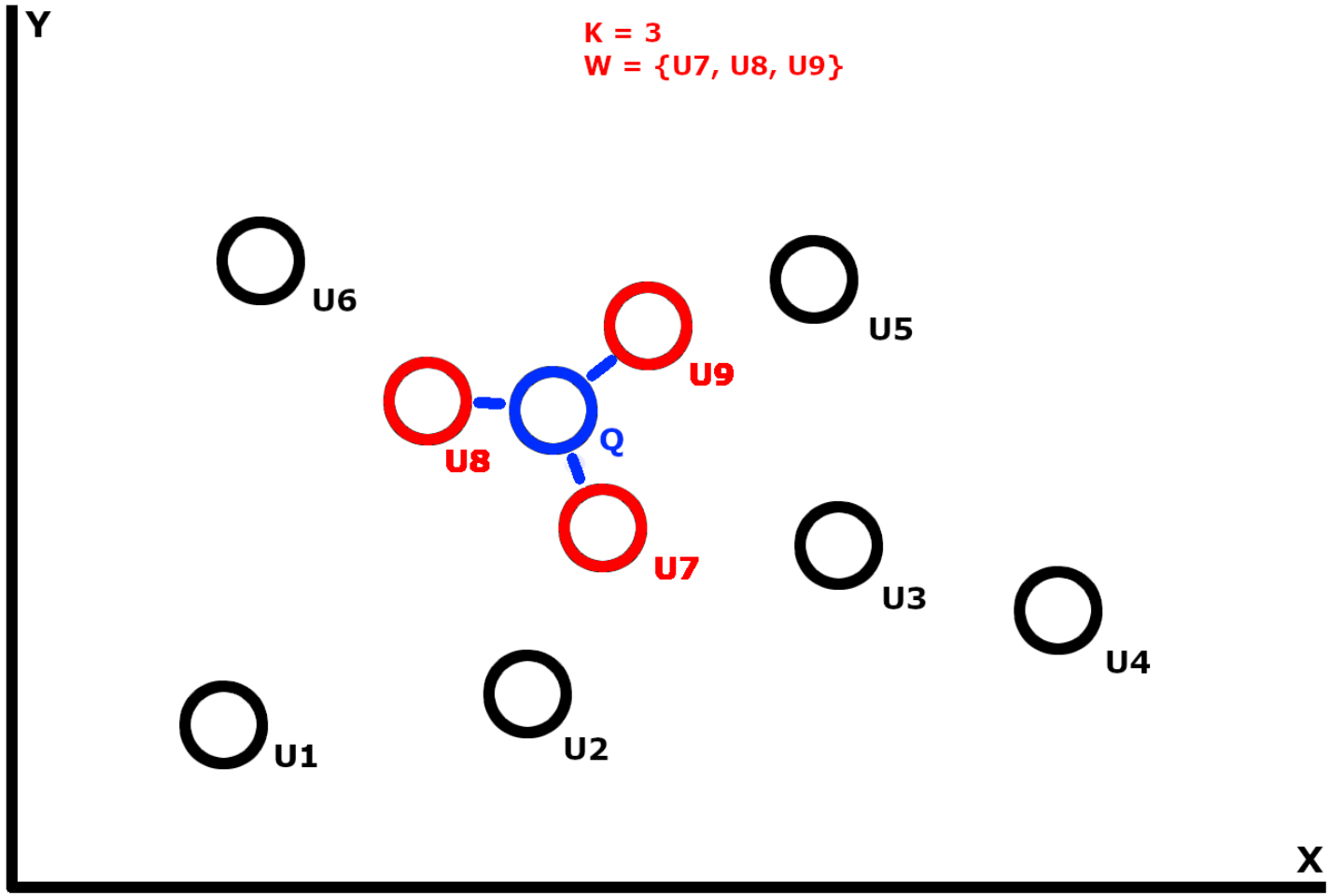
KNN



KNN



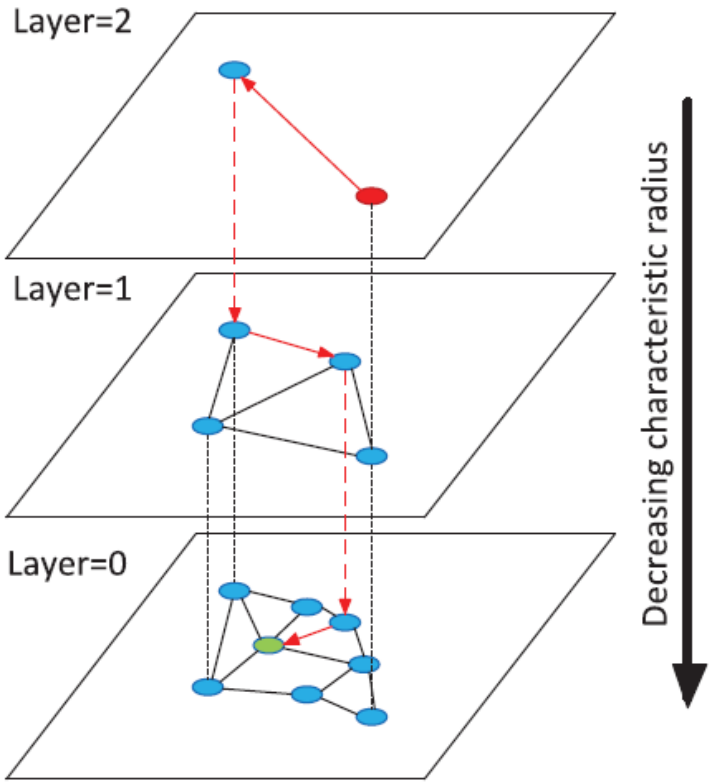
KNN



HNSW

- Hierarchical Navigable Small Worlds
- Řešení KNN problému, přibližné vyhledávání s využitím vícevrstvých grafů
- Výsledek poskytován s určitou přesností zvanou Recall
- Přesnost se dá zvýšit navýšením hodnoty parametru Ef, stejně tak poroste ale i čas vykonání dotazu

HNSW



Obrázek: Vrstvy grafů v HNSW

HNSW

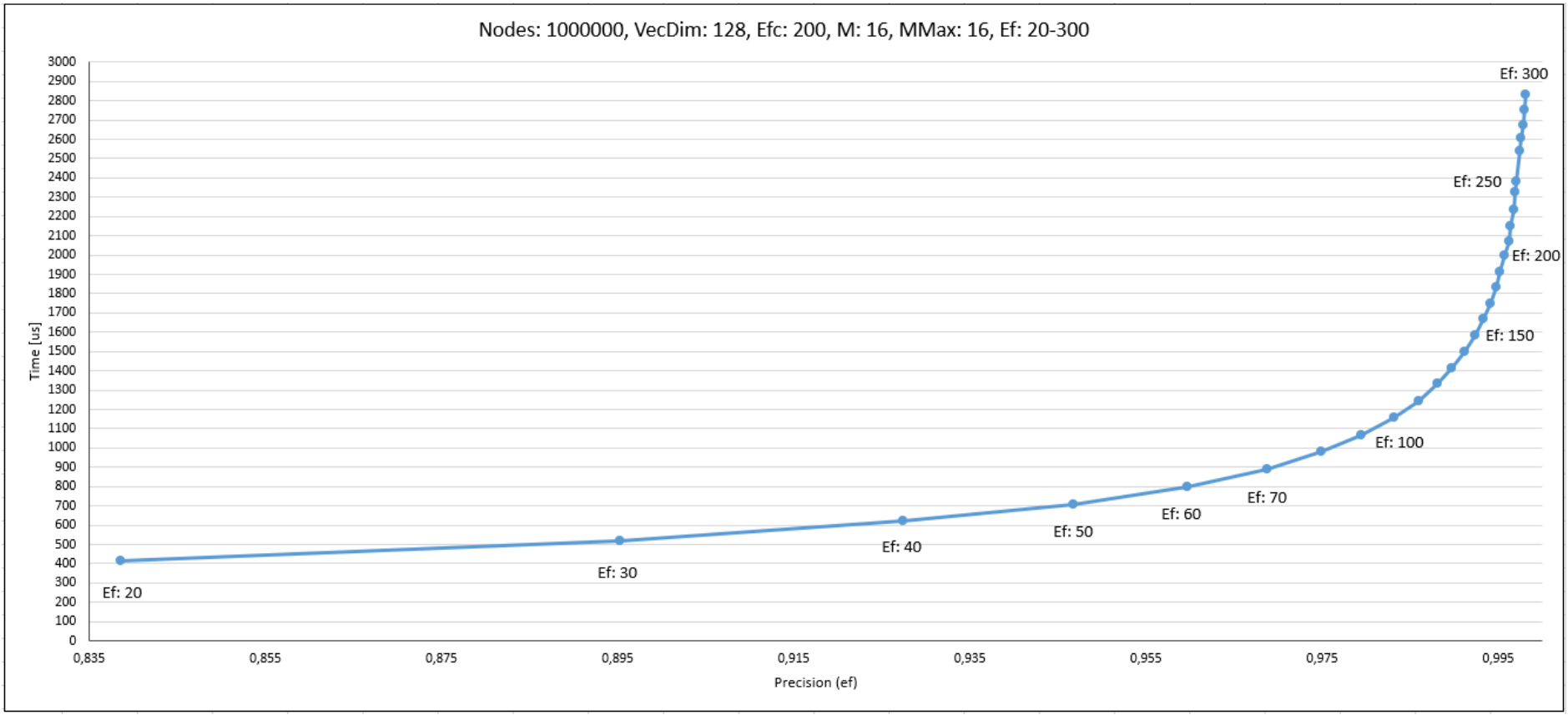
Algorithm 2. SEARCH-LAYER(q, ep, ef, l_c)

Input: query element q, enter-points ep, number of nearest to q elements to return ef, layer number l_c
Output: ef closest neighbors to q

```
1  v ← ep // set of visited elements
2  C ← ep // set of candidates
3  W ← ep // dynamic list of found nearest neighbors
4  while |C| > 0
5    c ← extract nearest element from C to q
6    f ← get furthest element from W to q
7    if distance(c, q) > distance(f, q)
8      break // all elements in W are evaluated
9    for each e ∈ neighbourhood(c) at layer l_c // update C and W
10     if e ∉ v
11       v ← v ∪ e
12       f ← get furthest element from W to q
13       if distance(e, q) < distance(f, q) or W < ef
14         C ← C ∪ e
15         W ← W ∪ e
16         if |W| > ef
17           remove furthest element from W to q
18 return W
```

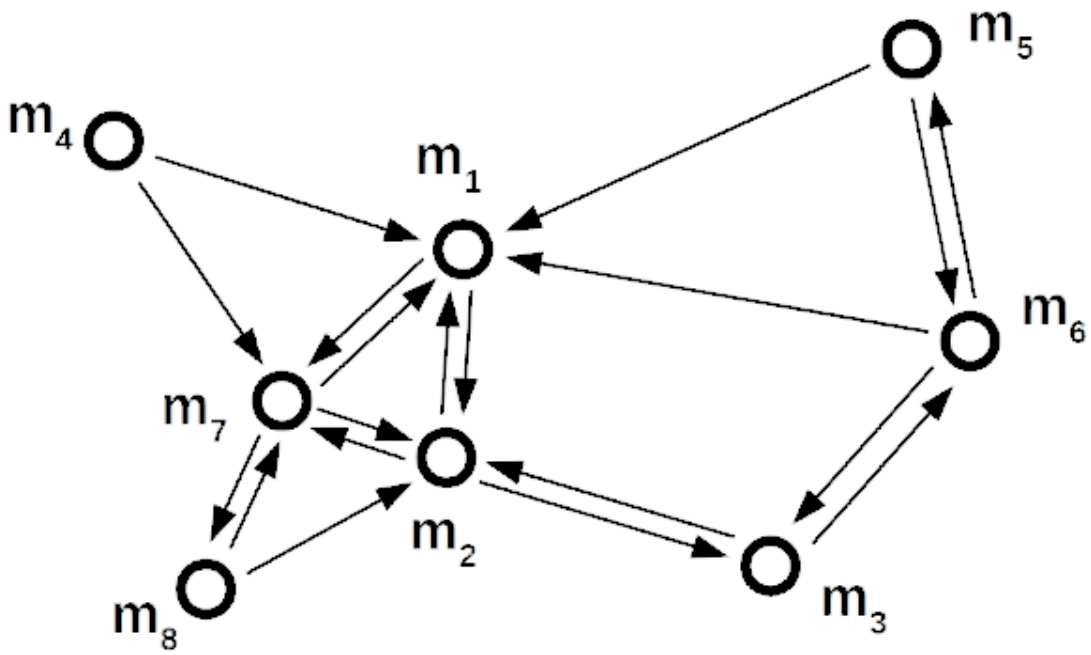
Obrázek: Pseudokód HNSW Search algoritmu

HNSW

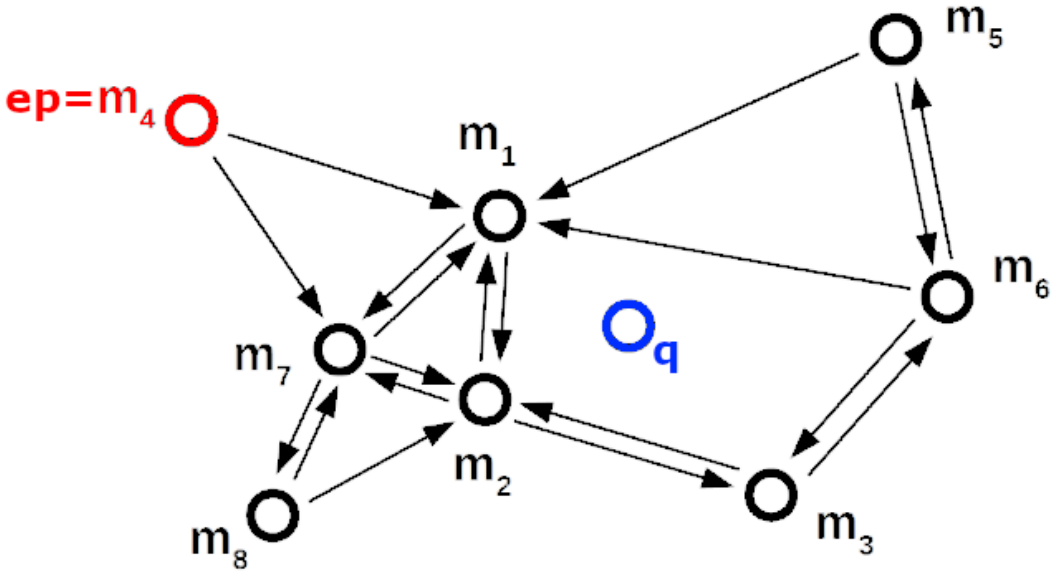


Obrázek: Graf závislosti průměrného času jednoho dotazu na přesnosti

HNSW

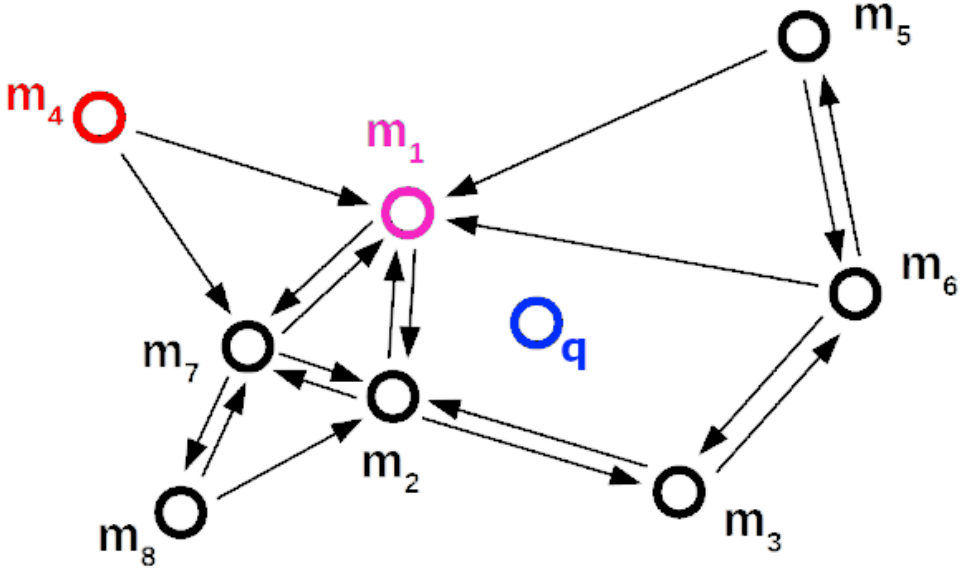


HNSW



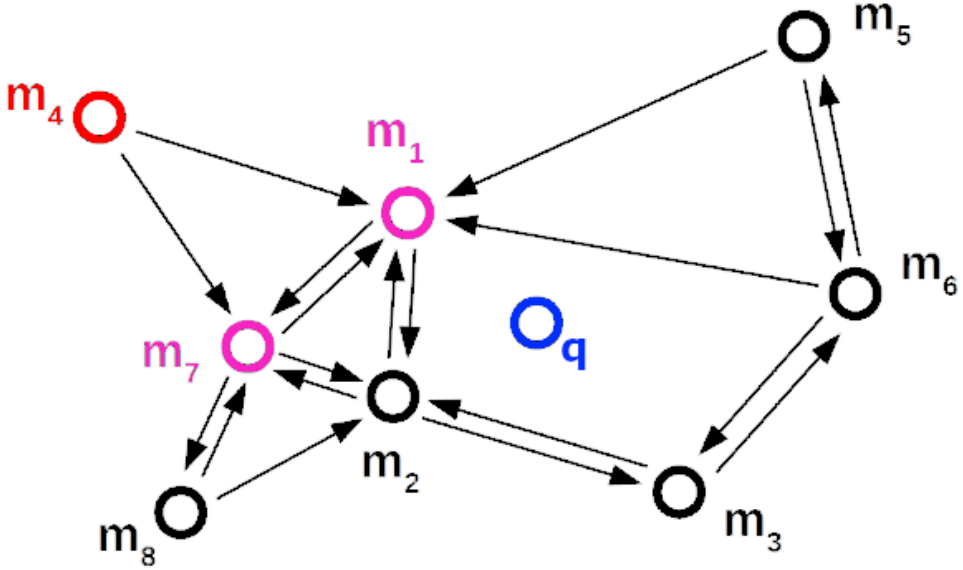
- ep = {m4}
- V = {m4}
- W = {m4}
- C = {m4}

HNSW



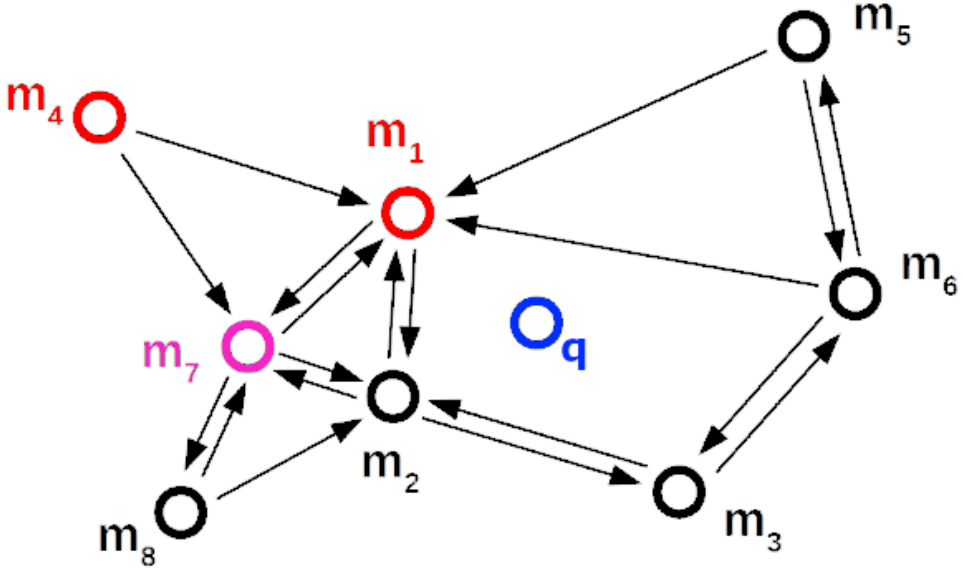
- V = {m4,m1}
- W = {m1,m4}
- C = {m1}
- f = m4
- c = m4

HNSW



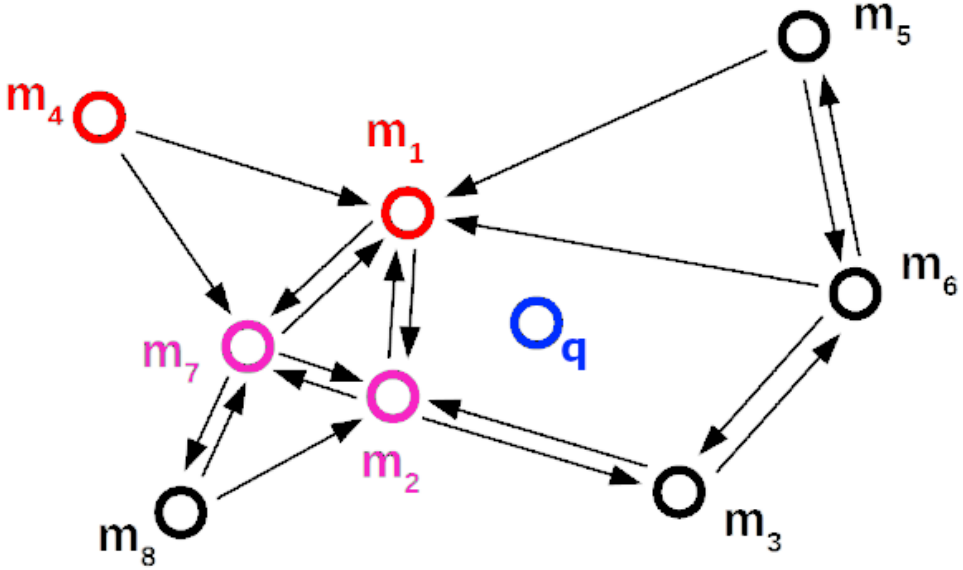
- V = {m4,m1,m7}
- W = {m1,m7,m4}
- C = {m1,m7}
- f = m4
- c = m4

HNSW



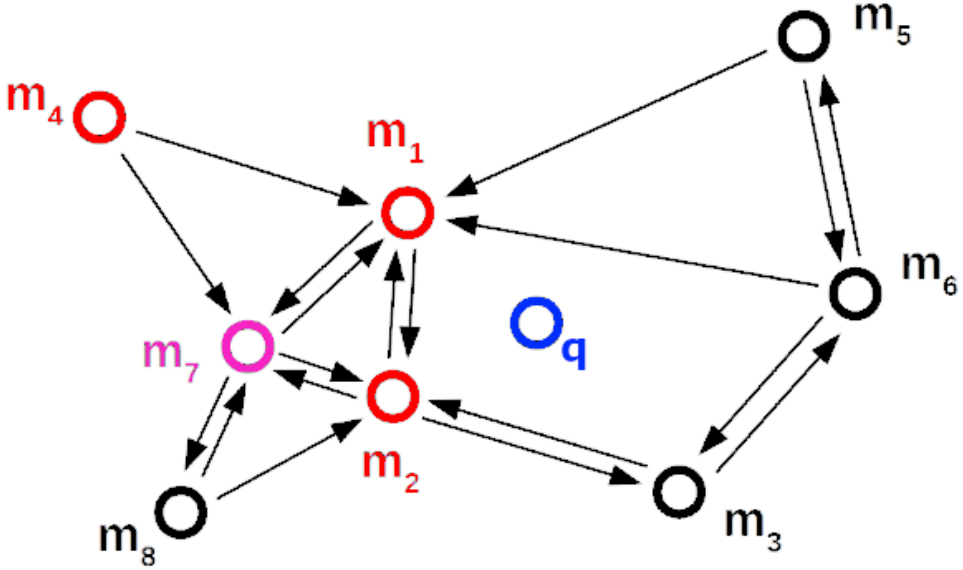
- V = {m4,m1,m7}
- W = {m1,m7,m4}
- C = {m7}
- f = m4
- c = m1

HNSW



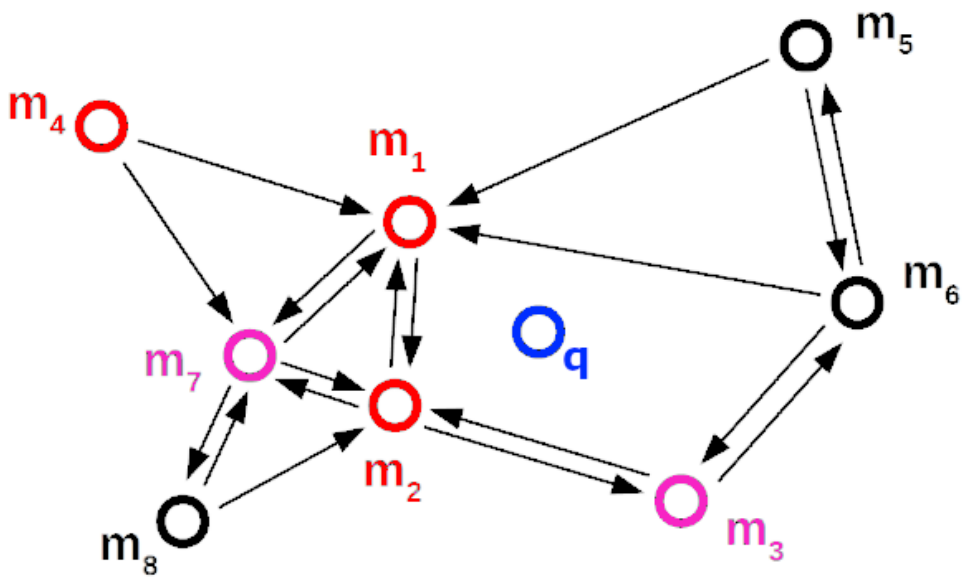
- V = {m4,m1,m7,m2}
- W = {m1,m2,m7}
- C = {m2,m7}
- f = m7
- c = m1

HNSW



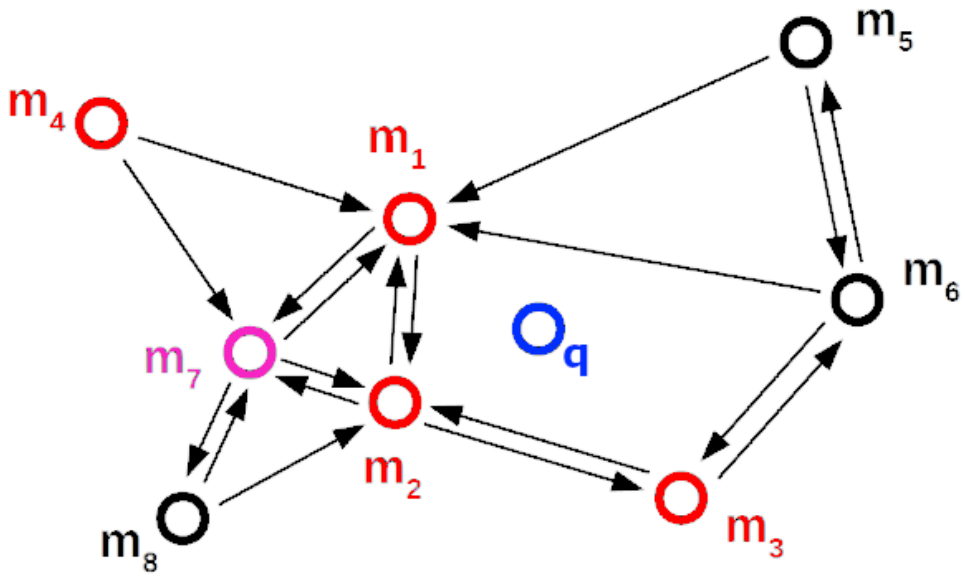
- V = {m4,m1,m7,m2}
- W = {m1,m2,m7}
- C = {m7}
- f = m7
- c = m2

HNSW



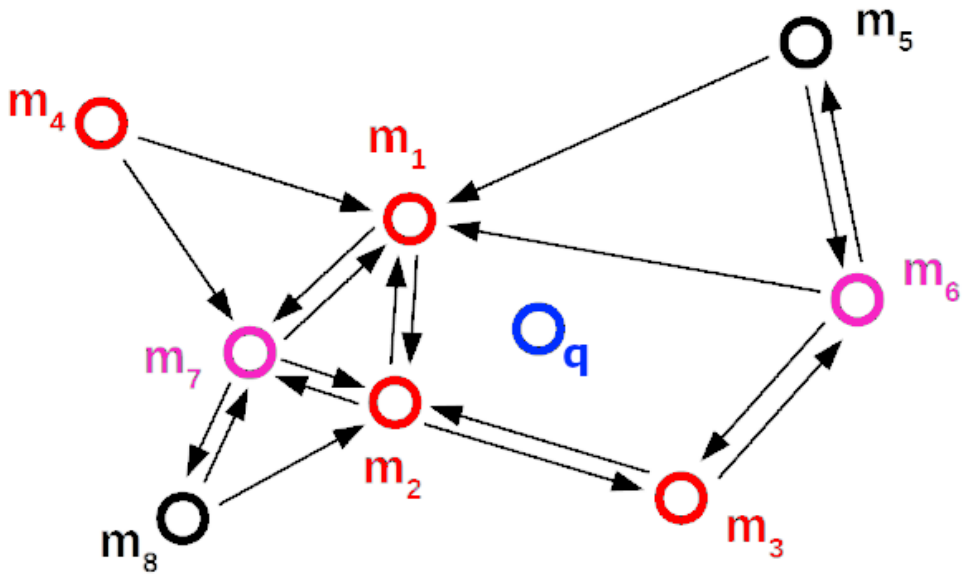
- V = {m4,m1,m7,m2,m3}
- W = {m1,m2,m3}
- C = {m3,m7}
- f = m3
- c = m2

HNSW



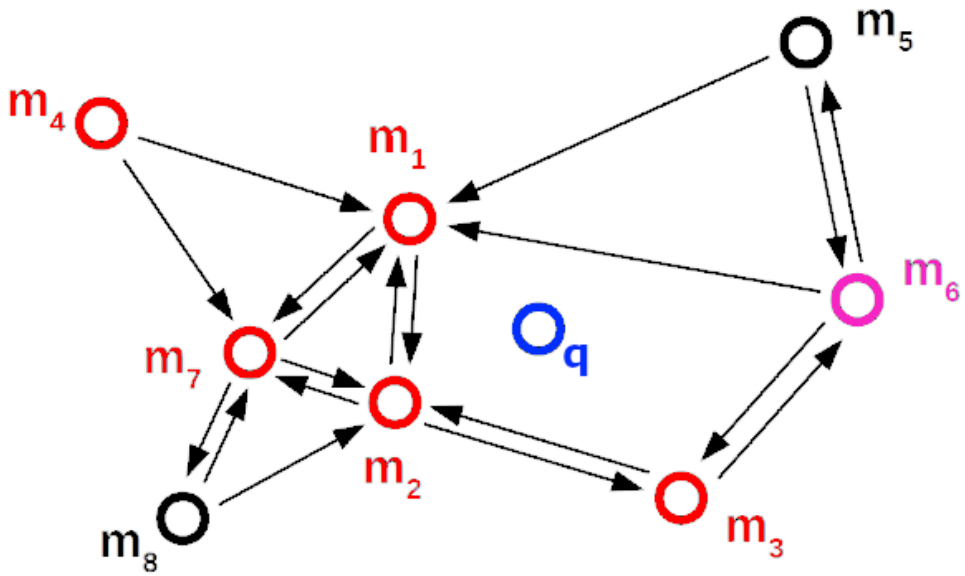
- V = {m4,m1,m7,m2,m3}
- W = {m1,m2,m3}
- C = {m7}
- f = m3
- c = m3

HNSW



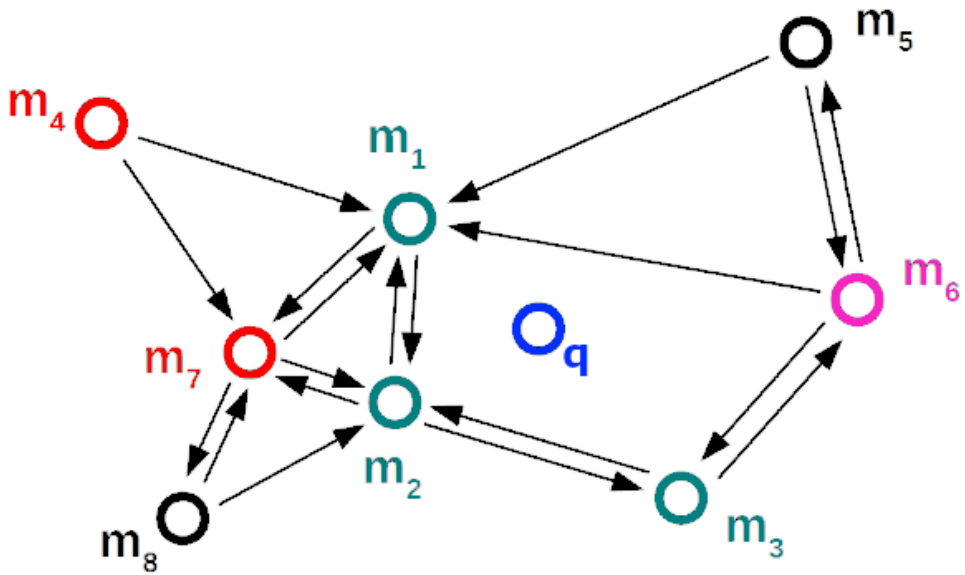
- V = {m4,m1,m7,m2,m3,m6}
- W = {m1,m2,m3}
- C = {m7,m6}
- f = m3
- c = m3

HNSW



- V = {m4,m1,m7,m2,m3,m6}
- W = {m1,m2,m3}
- C = {m6}
- f = m3
- c = m7

HNSW



- V = {m4,m1,m7,m2,m3,m6}
- W = {m1,m2,m3}
- C = {m6}
- f = m3
- c = m7
- $dist(c, q) > dist(f, q)$

Filtr

- Podmínka určující které vektory neprocházet a nevracet do výsledku
- Udává jakých hodnot mají nabývat jednotlivé atributy, nebo v jakých intervalech hodnot se mají nacházet
- Nemusíme omezovat všechny atribury
- Selektivita filtru $< 0, 1 >$ udává procentuální počet uzlů z celé množiny všech uzlů, které filtr přijme

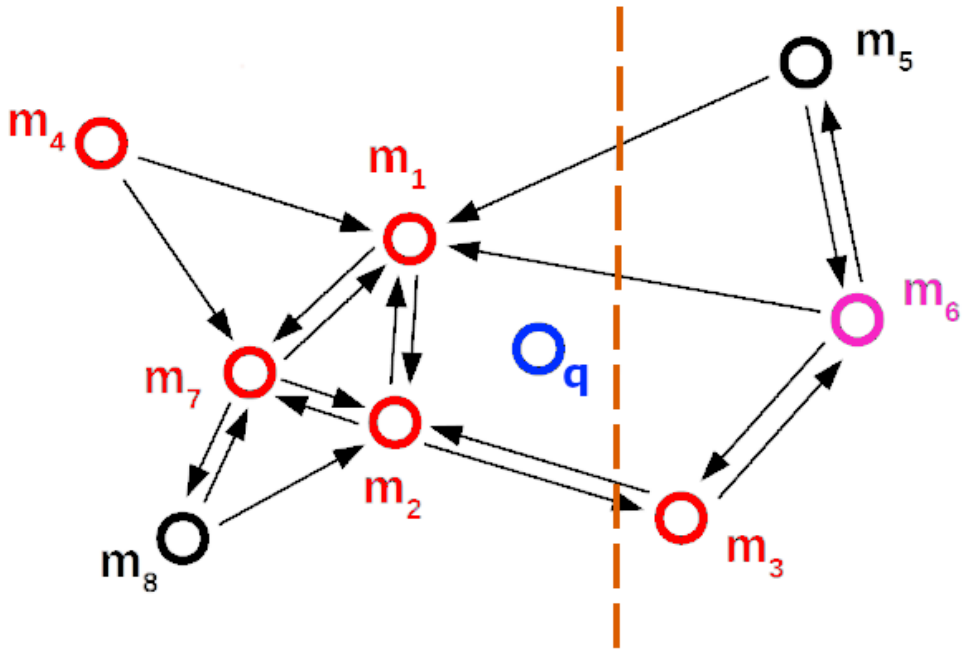
Filtr

- 72: $< 51.32, 143.87 >$; 88: $< 3 >$; 110: $< 72.40, 106.84 >$;

- $\text{vec}[72] \in < 51.32, 143.87 >$
- $\text{vec}[88] = 3$
- $\text{vec}[110] \in < 72.40, 106.84 >$

Filtr

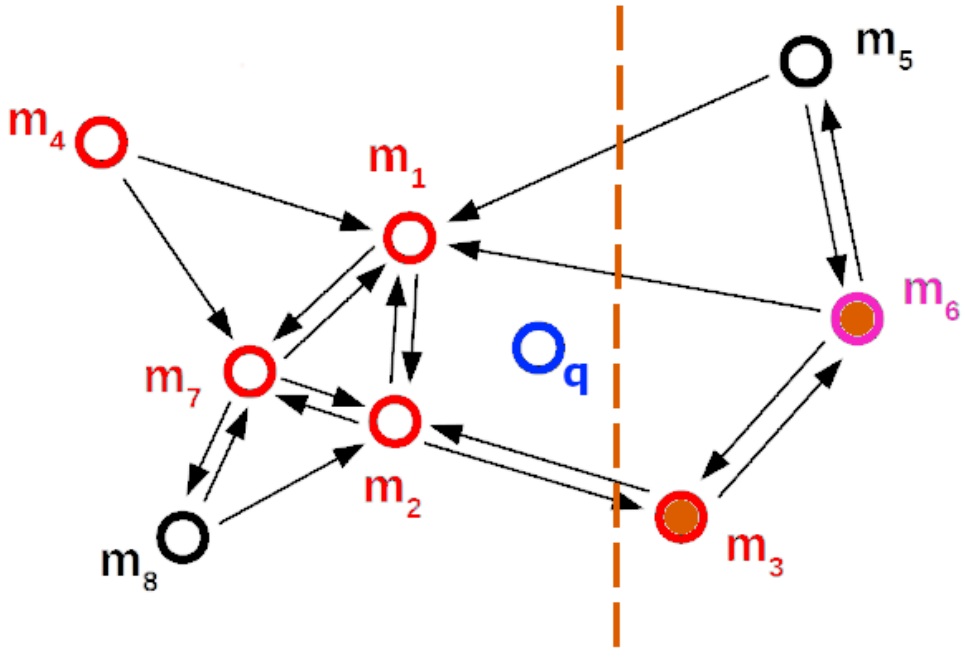
Filtr: vec[0] $\in < 50, 100 >$



- V = {m4,m1,m7,m2,m3,m6}
- W = {m1,m2,m3}
- C = {m6}
- f = m3
- c = m7
- $dist(c, q) > dist(f, q)$

Filtr

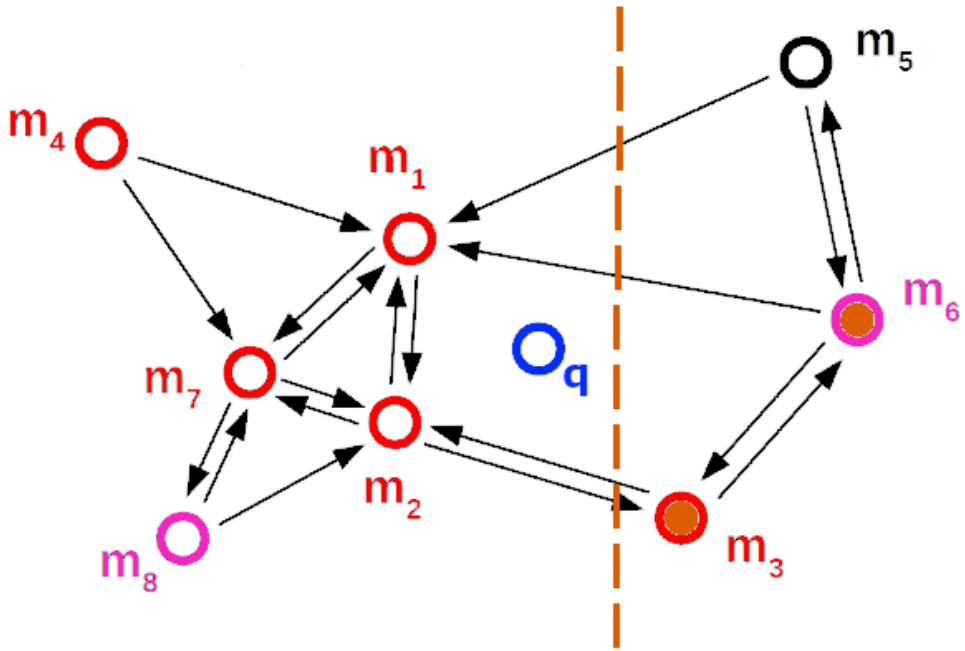
Filtr: vec[0] $\in < 50, 100 >$



- V = {m4,m1,m7,m2,m3,m6}
- F = {m3,m6}
- W = {m1,m2,m3}
- C = {m6}
- f = m3
- c = m7
- $dist(c, q) > dist(f, q)$
- $|F| == K$

Filtr

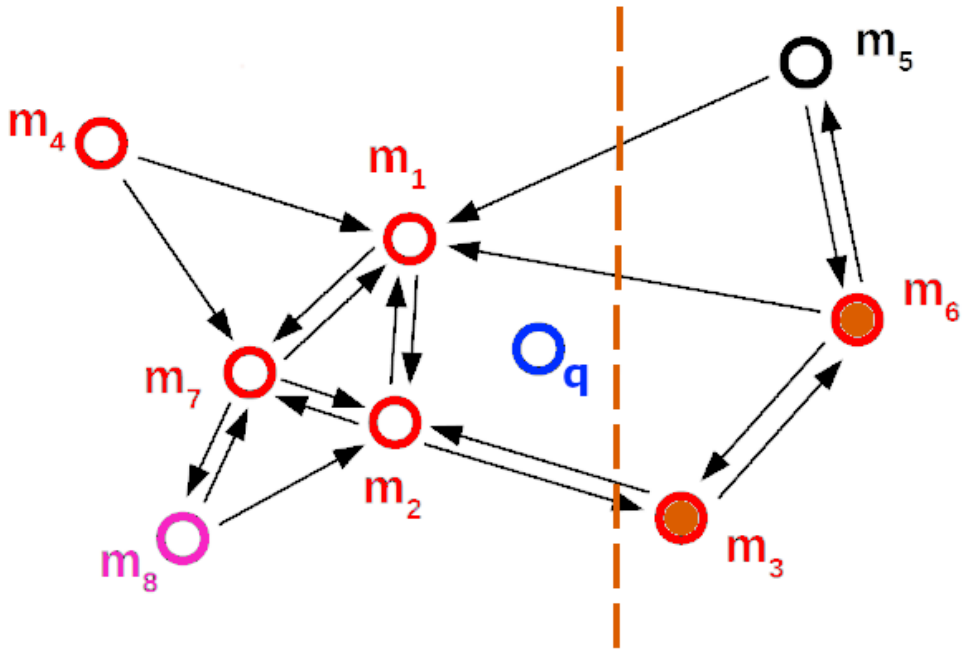
Filtr: vec[0] $\in < 50, 100 >$



- V = {m4,m1,m7,m2,m3,m6, m8}
- F = {m3,m6}
- W = {m1,m2,m3}
- C = {m6,m8}
- f = m3
- c = m7

Filtr

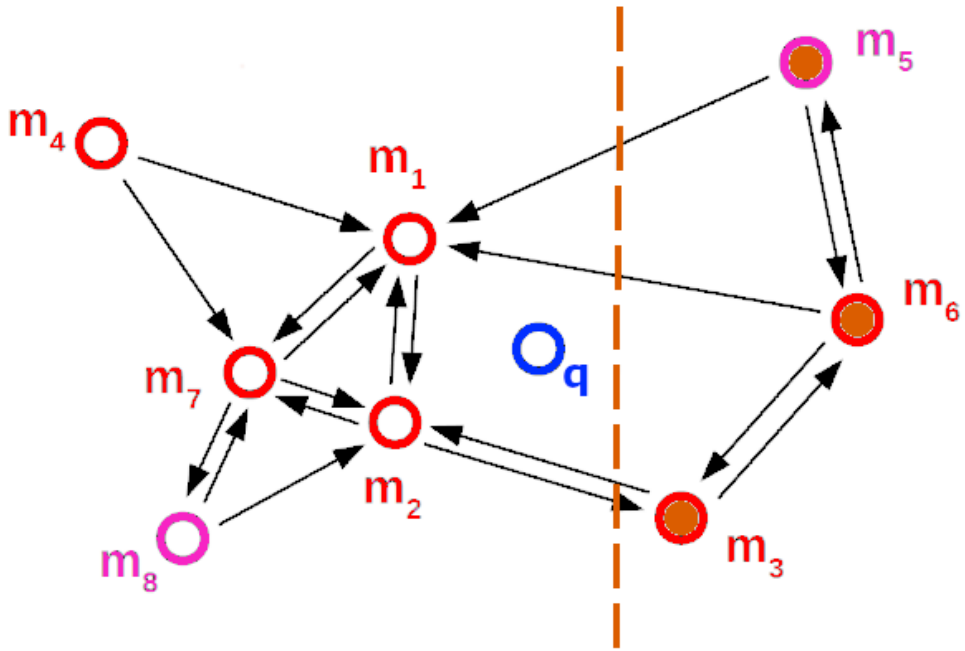
Filtr: vec[0] $\in < 50, 100 >$



- V = {m4,m1,m7,m2,m3,m6, m8}
- F = {m3,m6}
- W = {m1,m2,m3}
- C = {m8}
- f = m3
- c = m6

Filtr

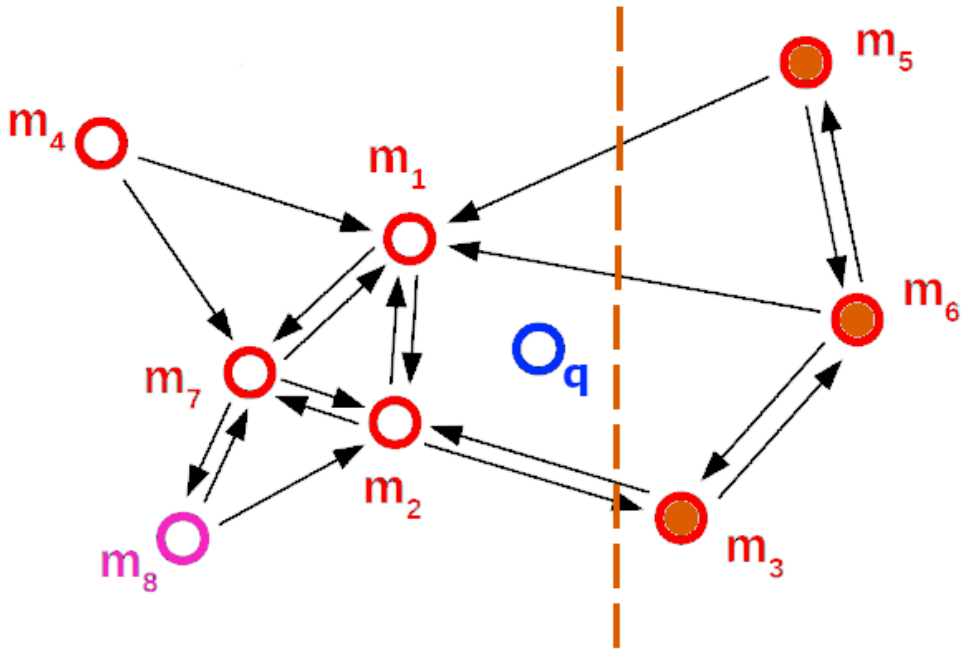
Filtr: vec[0] $\in < 50, 100 >$



- V = {m4,m1,m7,m2,m3,m6, m8,m5}
- F = {m3,m6,m5}
- W = {m1,m2,m3}
- C = {m5,m8}
- f = m3
- c = m6

Filtr

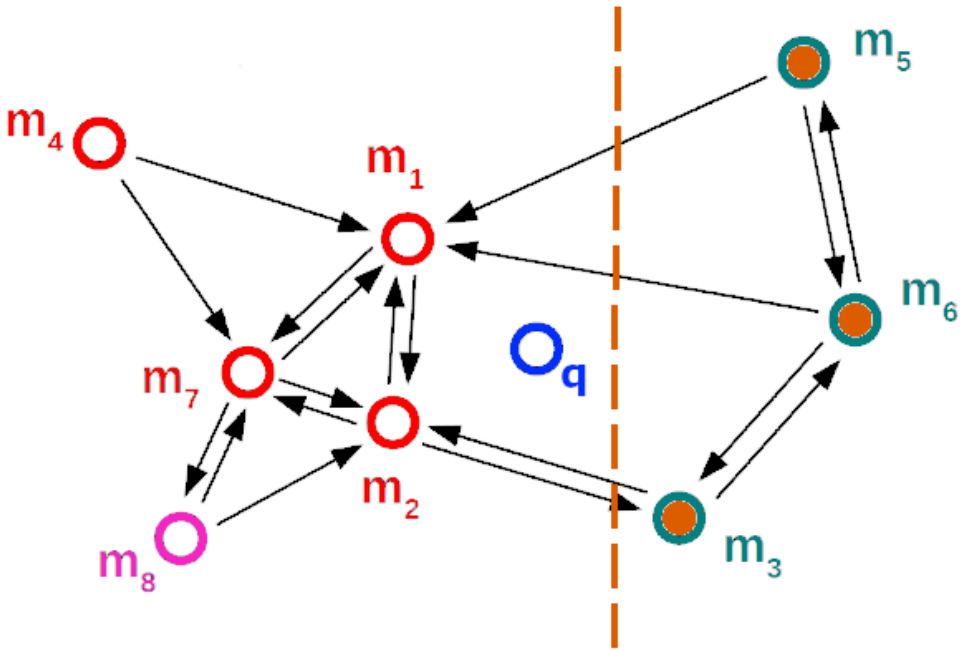
Filtr: vec[0] $\in < 50, 100 >$



- V = {m4,m1,m7,m2,m3,m6, m8,m5}
- F = {m3,m6,m5}
- W = {m1,m2,m3}
- C = {m8}
- f = m3
- c = m5

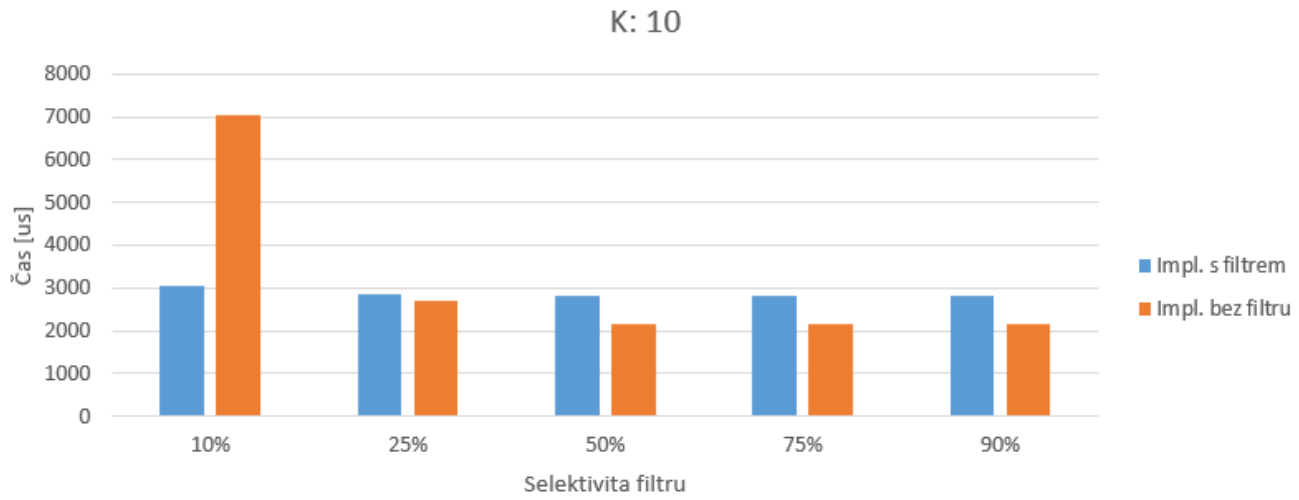
Filtr

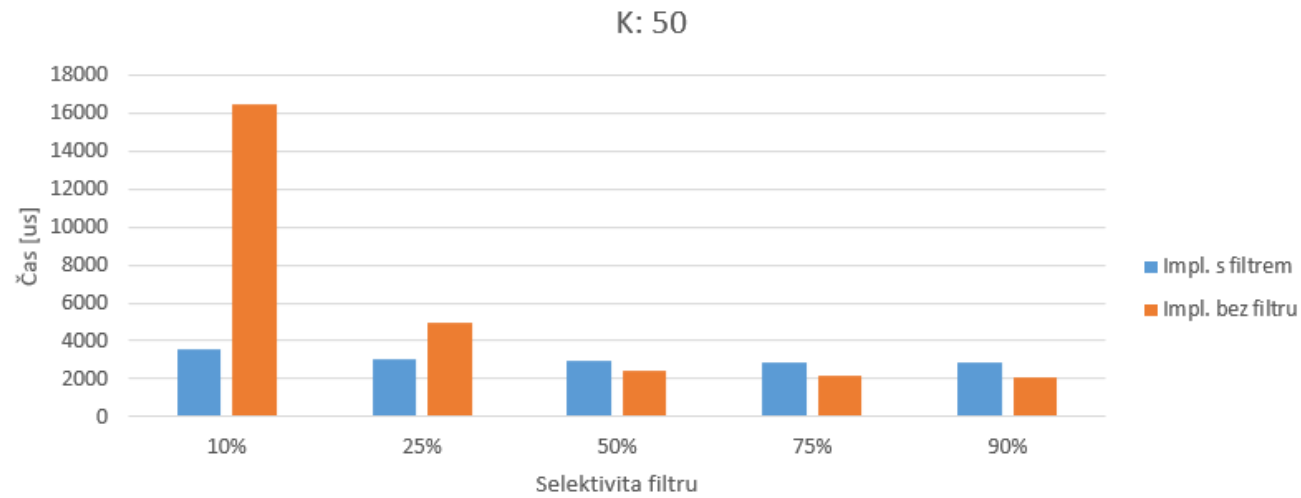
Filtr: vec[0] $\in < 50, 100 >$



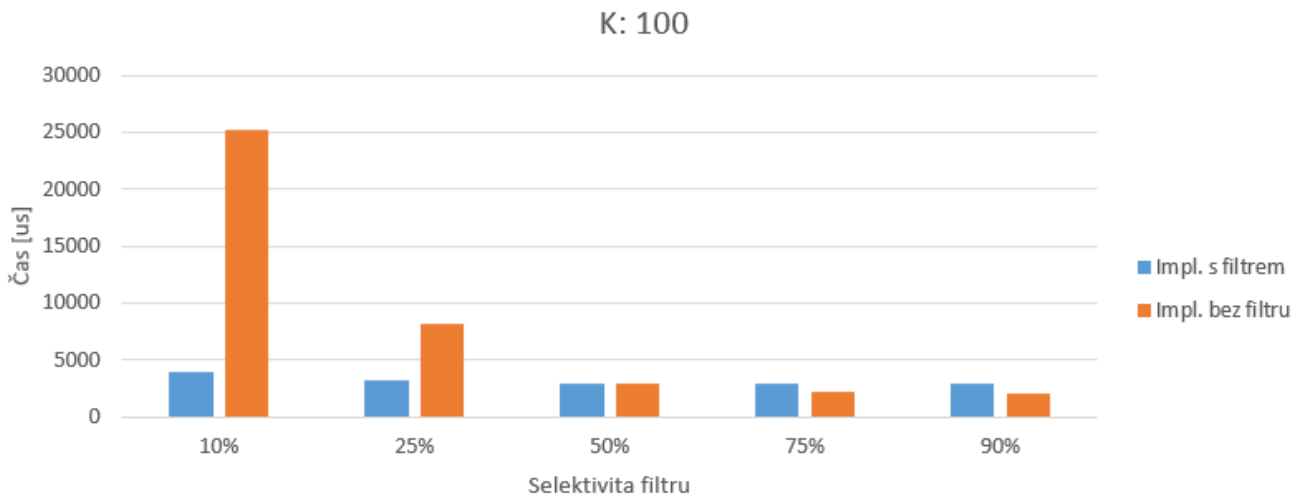
- V = {m4,m1,m7,m2,m3,m6, m8,m5}
- F = {m3,m6,m5}
- W = {m1,m2,m3}
- C = {m8}
- f = m3
- c = m5
- $dist(c, q) > dist(f, q)$
- $|F| == K$

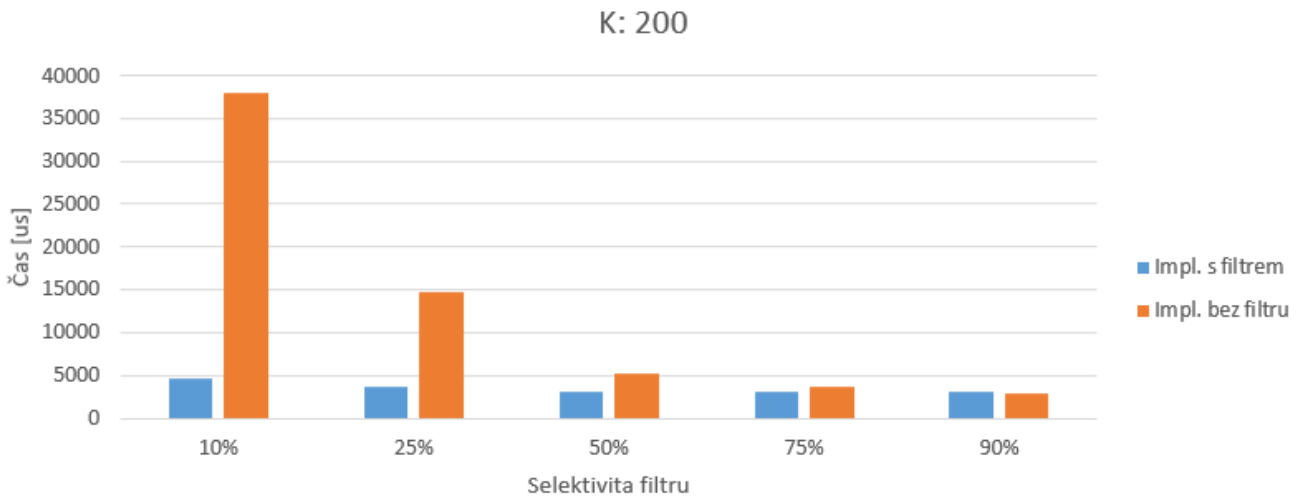
Filtr


K: 10
Čas [us]
8000
7000
6000
5000
4000
3000
2000
1000
0
10%
25%
50%
75%
90%
Selektivita filtru
Impl. s filtrem
Impl. bez filtru


K: 50
Čas [us]
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
10%
25%
50%
75%
90%
Selektivita filtru
Impl. s filtrem
Impl. bez filtru

Filtr


K: 100
30000
25000
20000
15000
10000
5000
0
Čas [us]
10%
25%
50%
75%
90%
Selektivita filtru
Impl. s filtrem
Impl. bez filtru


K: 200
40000
35000
30000
25000
20000
15000
10000
5000
0
Čas [us]
10%
25%
50%
75%
90%
Selektivita filtru
Impl. s filtrem
Impl. bez filtru

Závěr

- Splnění všech požadavků
- Funkční implementace původního HNSW
- HNSW implementace o polovinu pomalejší než reference
- Funkční implementace rozšířeného HNSW o filtry

Poděkování

Děkuji za pozornost

Citace I

- *ann-benchmarks* [online]. 2022. [cit. 2022-03-06]. Dostupné z: http://ann-benchmarks.com/index.html.
- *git-hnswlib: hnswlib* [online]. 2022. [cit. 2022-03-06]. Dostupné z: https://github.com/nmslib/hnswlib.
- *git-hnsw: hnswlib* [online]. 2022. [cit. 2022-04-10]. Dostupné z: https://github.com/RadimBaca/HNSW.
- *Nearest neighbor search* [online]. 2022. [cit. 2022-03-06]. Dostupné z: https://en.wikipedia.org/wiki/Nearest_neighbor_search.
- *Optimalizace v INFORMIXU* [online]. 2022. [cit. 2022-04-04]. Dostupné z: http://www.ms.mff.cuni.cz/~jkoc5219/Optimalizace_v_INFORMIXU.html.

Citace II

MALKOV, Yu A; YASHUNIN, Dmitry A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*. 2018, roč. 42, č. 4, s. 824–836.

Metrický prostor. 2022. Dostupné také z: `https://cs.wikipedia.org/wiki/Metrick%C3%BD_prostor#Definice`.

K-Nearest Neighbors (KNN) algorithm. 2022. Dostupné také z: `https://towardsdatascience.com/k-nearest-neighbors-knn-algorithm-23832490e3f4`.