SLOVENSKÁ TECHNICKÁ UNIVERZITA
Fakulta informatiky a informačných technologií
Ilkovičova 2, 842 16 Bratislava 4

Zadanie č. 3 - Riešenie problému obchodného cestujúceho algoritmov zakázaného prehľadávania a simulovaného žíhania

Dávid Penťa ID: 110871

Umelá inteligencia

FIIT STU 2021/2022

Zadanie

Máme 2D priestor, ktorý má rozmery X a Y, v intervaloch od -5000 do +5000. Tento 2D priestor vyplňte 20 bodmi, pričom každý bod má náhodne zvolenú polohu pomocou súradníc X a Y. Každý bod má unikátne súradnice (t.j. nemalo by byť viacej bodov na presne tom istom mieste).

Po vygenerovaní 20 náhodných bodov vygenerujte ďalších 20000 bodov, avšak tieto body nebudú generované úplne náhodne, ale nasledovným spôsobom:

1. Náhodne vyberte jeden zo **všetkých** doteraz vytvorených bodov v 2D priestore. Ak je bod príliš blízko okraju, tak zredukujete príslušný interval v nasledujúcich dvoch krokoch.
2. Vygenerujte náhodné číslo X_offset v intervale od -100 do +100
3. Vygenerujte náhodné číslo Y_offset v intervale od -100 do +100
4. Pridajte nový bod do 2D priestoru, ktorý bude mať súradnice ako náhodne vybraný bod v kroku 1, pričom tieto súradnice budú posunuté o X_offset a Y_offset

Vašou úlohou je naprogramovať zhlukovač pre 2D priestor, ktorý zanalyzuje 2D priestor so všetkými jeho bodmi a rozdelí tento priestor na k zhlukov (klastrov). Implementujte rôzne verzie zhlukovača, konkrétne týmito algoritmami:

- k-means, kde stred je centroid
- k-means, kde stred je medoid
- aglomeratívne zhlukovanie, kde stred je centroid
- divízne zhlukovanie, kde stred je centroid

Vyhodnocujte úspešnosť/chybovosť vášho zhlukovača. Za úspešný zhlukovač považujeme taký, v ktorom žiaden klaster nemá priemernú vzdialenosť bodov od stredu viac ako 500.

Vizualizácia: pre každý z týchto experimentov vykreslite výslednú 2D plochu tak, že označkujete (napr. vyfarbíte, očíslujete, zakrúžkujete) výsledné klastre.

Dokumentácia musí obsahovať opis konkrétne použitých algoritmov a reprezentácie údajov. V závere zhodnoťte dosiahnuté výsledky ich porovnaním.

Riešenie

Môj program dokáže zhlukovať podľa implementácií týchto štyroch algoritmov:

K-means, kde stred je centroid

Na začiatku program vyberie k rôznych bodov, centoidov. Potom ku každému bodu nájde najbližší centroid, ku ktorému mu ho následne priradí. Tak sa vytvoria zhluky. Pre každý zhluk sa vypočíta nový centroid a to tak, že sa spraví priemer súradníc každého bodu v zhluku. Toto program opakuje 100 krát alebo dokým sa zhluky prestanú meniť.

K-means, kde stred je medoid

Na začiatku program vyberie *k* rôznych bodov, medoidov. Potom ku každému bodu nájde najbližší medoid, ku ktorému mu ho následne priradí. Tak sa vytvoria zhluky. Pre každý zhluk sa vypočíta nový medoid a to tak, že sa nájde bod, ktorého vzdialenosť od všetkých ostatných bodov v zhluku je najmenší. Toto program opakuje 100 krát alebo dokým sa zhluky prestanú meniť.

Divízne zhlukovanie, kde stred je centroid

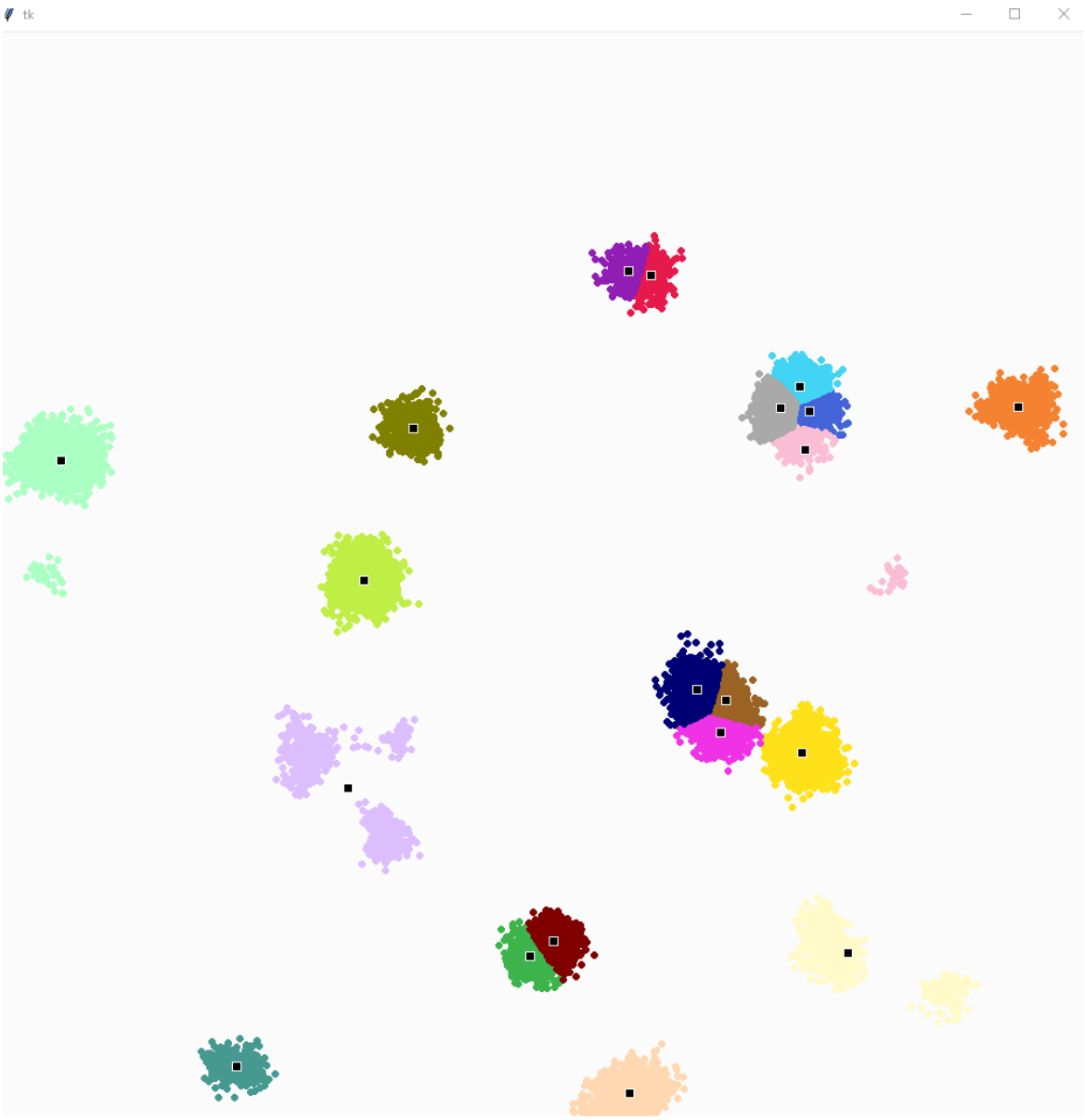
Všetky body sa na začiatku umiestia do jedného zhluku. Ten sa podľa k-means, kde stred je centroid, rozdelí na dva zhluky. Potom sa nájde zhluk, ktorý obsahuje najviac bodov a tiež sa podľa k-means rozdelí na dva. Toto sa opakuje n-krát, čo môže byť až dokým je každý bod vo vlastnom samostatnom zhluku.

Aglomeratívne zhlukovanie, kde stred je centroid

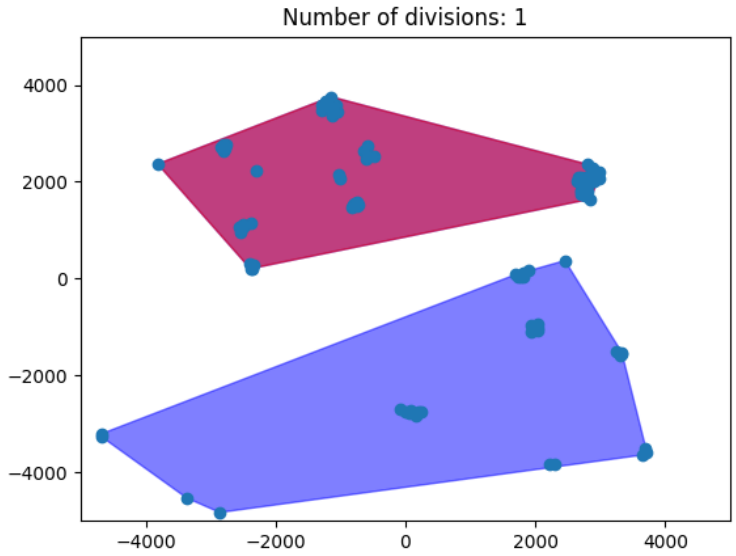
Tento algoritmus som implementoval pomocou matice vzdialeností. Na začiatku sa berie každý bod ako centroid svojho vlastného zhluku a vypočíta sa matica vzdialeností všetkých centroidov. Potom sa v nej nájde, ktoré centroidy sú k sebe najbližšie. Riadky a stĺpce týchto centroidov sa z matice odstránia a body ktoré do nich patria sa spoja v poli pre zhluky. Vypočíta sa centroid tohto zhluku a následne sa pridá do matice vzdialeností. Toto sa opakuje, dokým sa nevytvorí len jeden zhluk so všetkými bodmi.

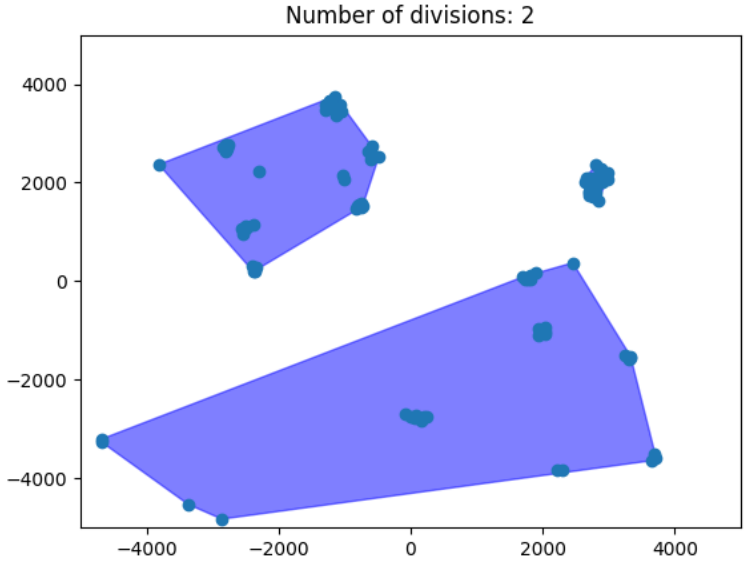
Vizualizácia

Algoritmy k-means vykresľujem pomocou modulu tkinter. Zhluky sú farebne odlíšené a centroidy alebo medoidy zhlukov sú vyznačené ako čierne štvorce.

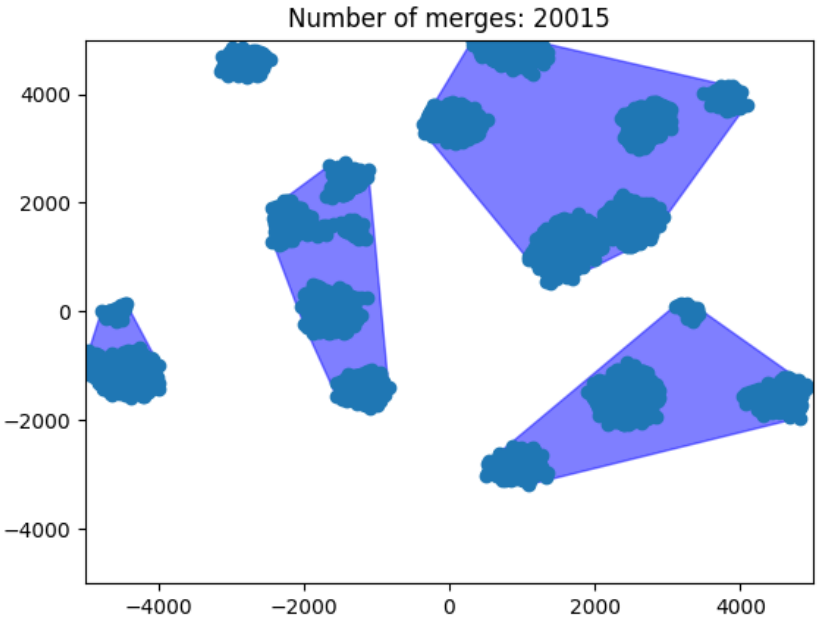


Divízne a aglomeratívne zhlukovanie vykresľujem pomocou modulu matplotlib. Jednotlivé zhluky sú orámované a vyfarbené. Pri animácii divízneho zhlukovania sa zhluk, ktorý sa bude rozdeľovať, farebne odlíši.

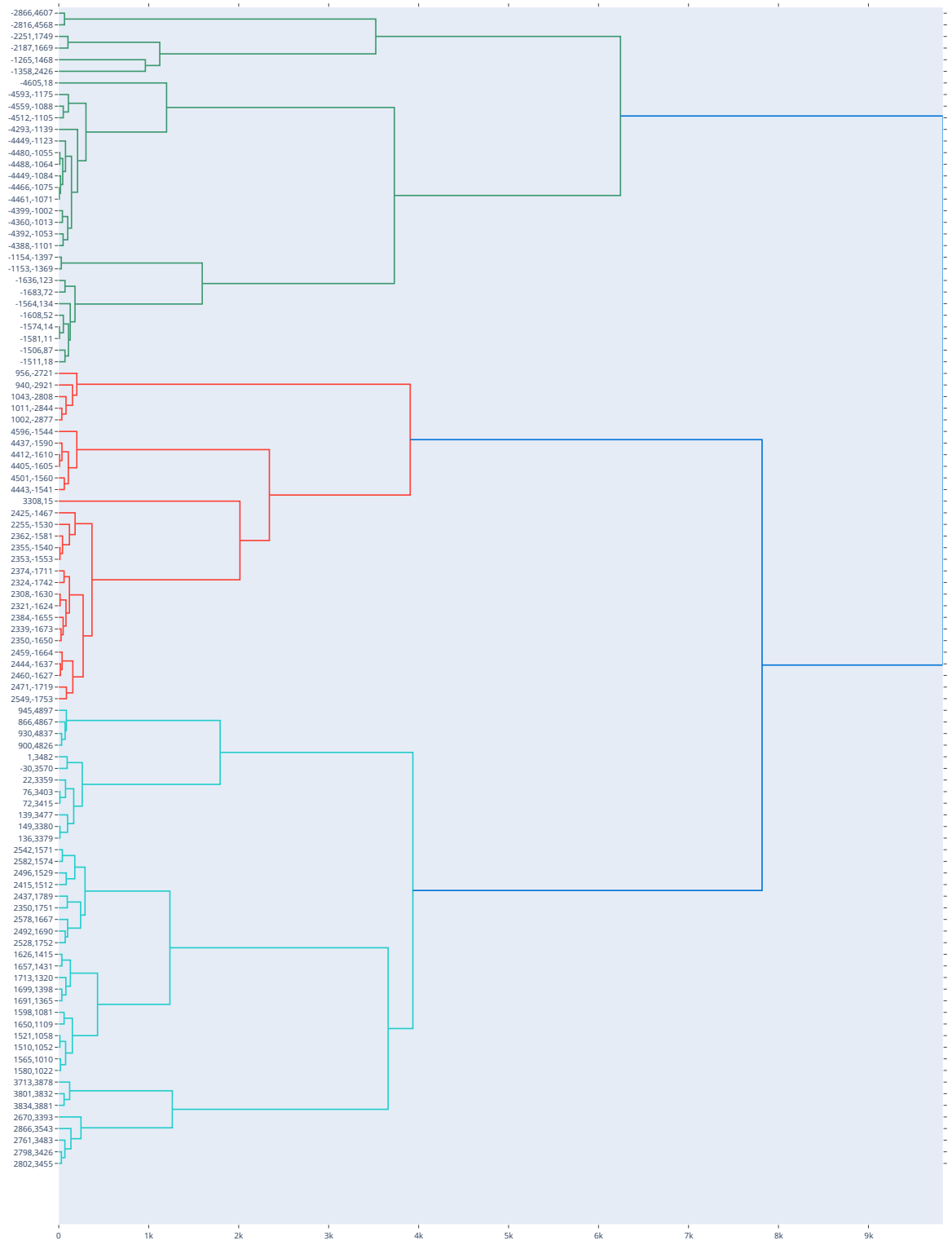




Pri aglomeratívnom zhlukovaní je možné vykresliť dendrogram alebo graf zo súboru, do ktorého sa postupne ukladali zhluky. Môžeme takto vykresliť zhluk pri ľubovoľnom počte spojení zhlukov.



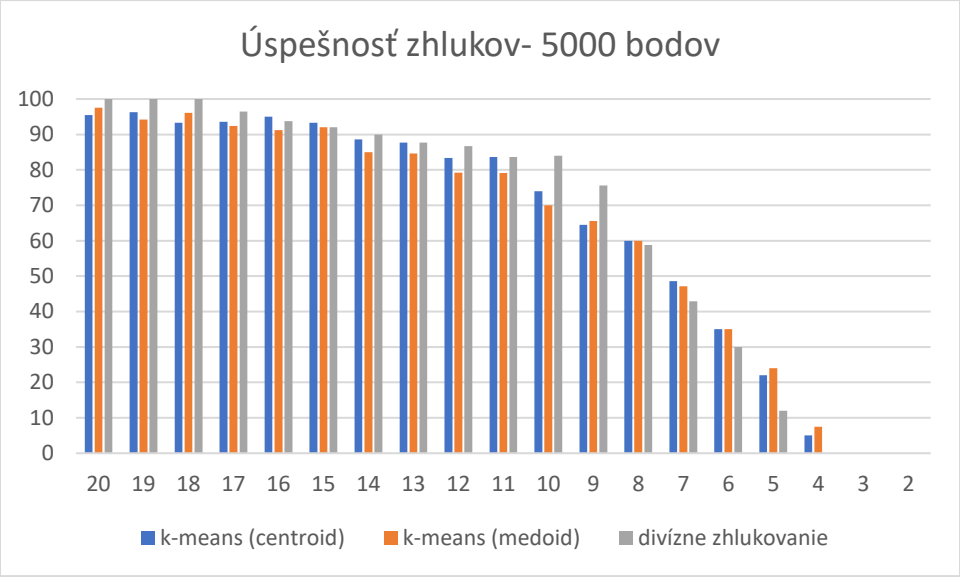
Dendrogram

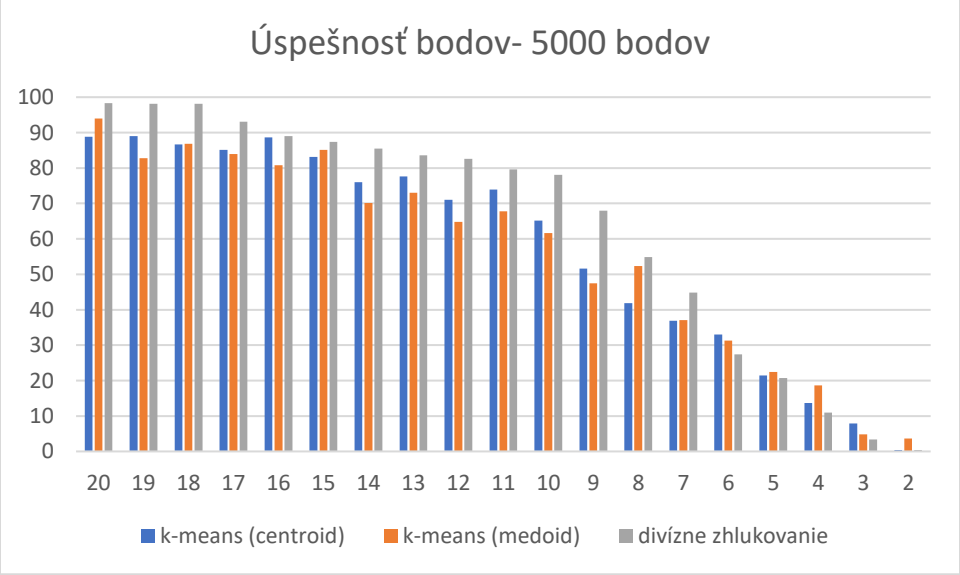

-2866,4607
-2816,4568
-2251,1749
-2187,1669
-1265,1468
-1358,2426
-4605,18
-4593,-1175
-4559,-1088
-4512,-1105
-4293,-1139
-4449,-1123
-4480,-1055
-4488,-1064
-4449,-1084
-4466,-1075
-4461,-1071
-4399,-1002
-4360,-1013
-4392,-1053
-4388,-1101
-1154,-1397
-1153,-1369
-1636,123
-1683,72
-1564,134
-1608,52
-1574,14
-1581,11
-1506,87
-1511,18
956,-2721
940,-2921
1043,-2808
1011,-2844
1002,-2877
4596,-1544
4437,-1590
4412,-1610
4405,-1605
4501,-1560
4443,-1541
3308,15
2425,-1467
2255,-1530
2362,-1581
2355,-1540
2353,-1553
2374,-1711
2324,-1742
2308,-1630
2321,-1624
2384,-1655
2339,-1673
2350,-1650
2459,-1664
2444,-1637
2460,-1627
2471,-1719
2549,-1753
945,4897
866,4867
930,4837
900,4826
1,3482
-30,3570
22,3359
76,3403
72,3415
139,3477
149,3380
136,3379
2542,1571
2582,1574
2496,1529
2415,1512
2437,1789
2350,1751
2578,1667
2492,1690
2528,1752
1626,1415
1657,1431
1713,1320
1699,1398
1691,1365
1598,1081
1650,1109
1521,1058
1510,1052
1565,1010
1580,1022
3713,3878
3801,3832
3834,3881
2670,3393
2866,3543
2761,3483
2798,3426
2802,3455
0
1k
2k
3k
4k
5k
6k
7k
8k
9k

Testovanie

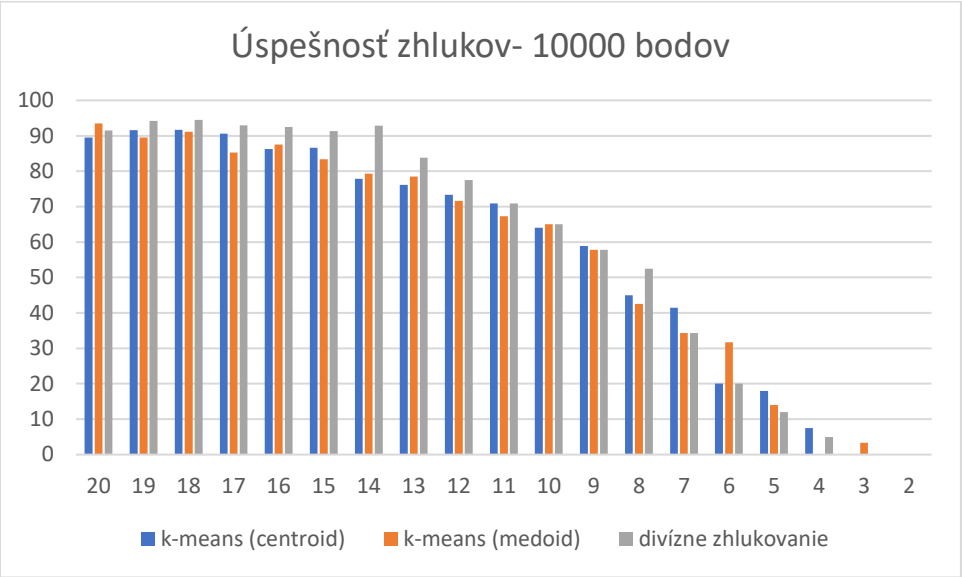
V testovaní som porovnával úspešnosť zlučovačov k-means a aglomeratívneho zhlukovania na jednej množine bodov, ktorú som postupne zväčšoval. Úspešnosť zhlukov som vyhodnocoval podľa toho, koľko percent zhlukov malo priemernú vzdialenosť bodov od stredu menej ako 500. Úspešnosť bodov som vyhodnocoval podľa toho, koľko percent bodov malo vzdialenosť od stredu menej ako 500. V grafoch je zapísaný priemer z 10 behov algoritmu.

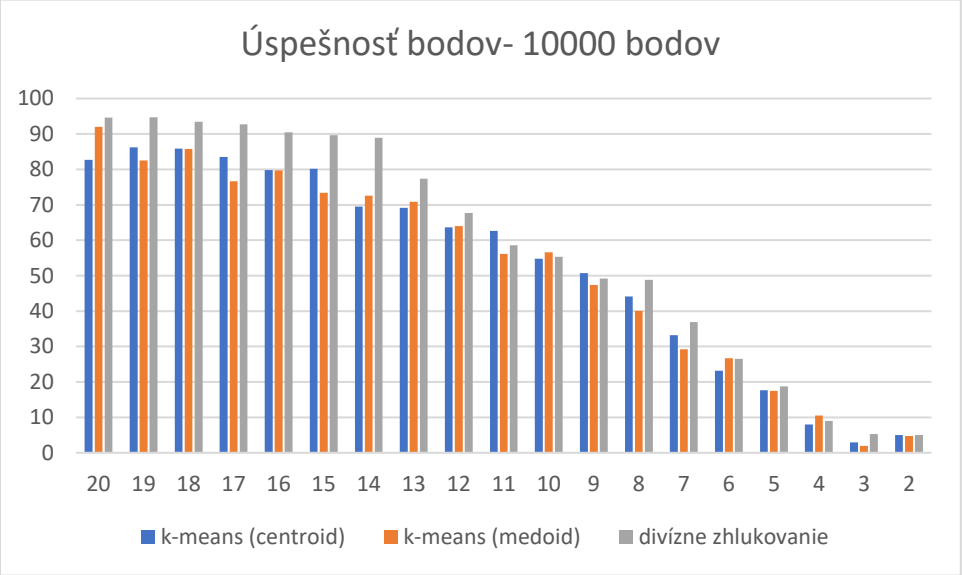
5000 bodov



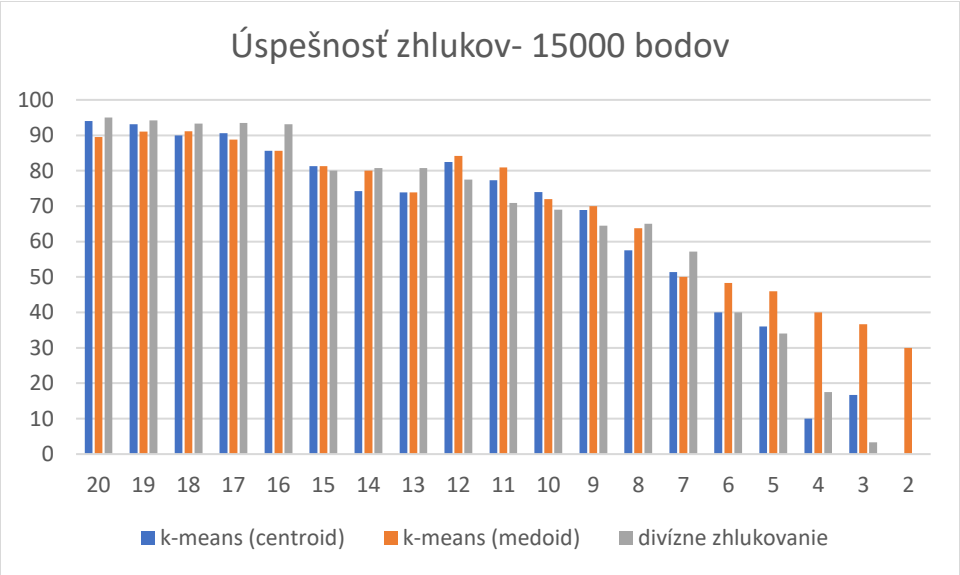


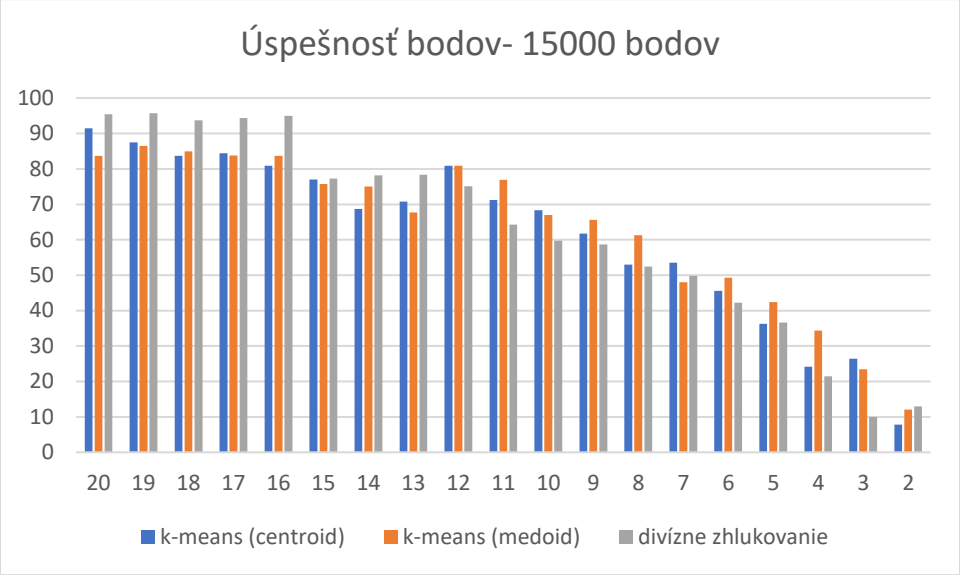
10000 bodov


Úspešnosť zhlukov- 10000 bodov
100
90
80
70
60
50
40
30
20
10
0
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2
k-means (centroid)
k-means (medoid)
divízne zhlukovanie

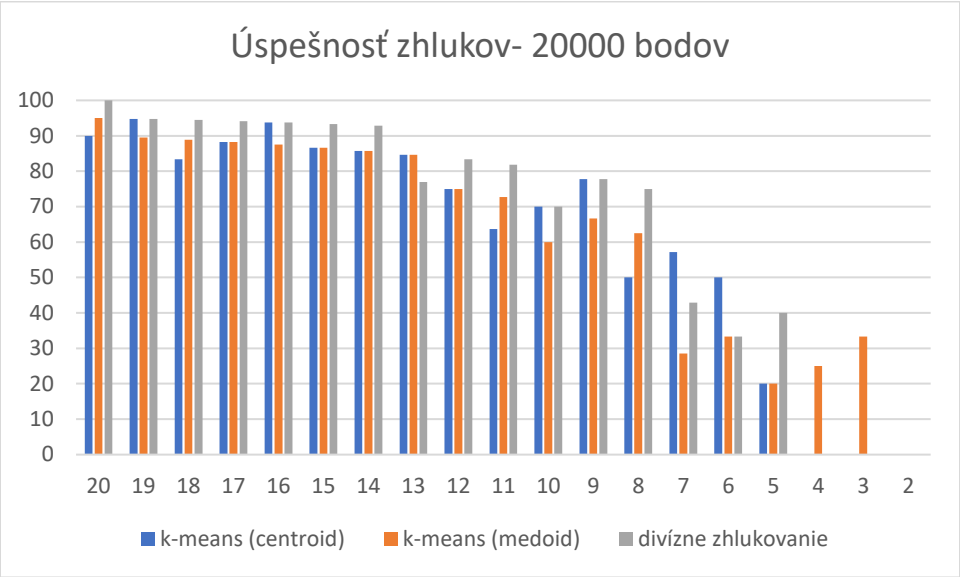

Úspešnosť bodov- 10000 bodov
100
90
80
70
60
50
40
30
20
10
0
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2
k-means (centroid)
k-means (medoid)
divízne zhlukovanie

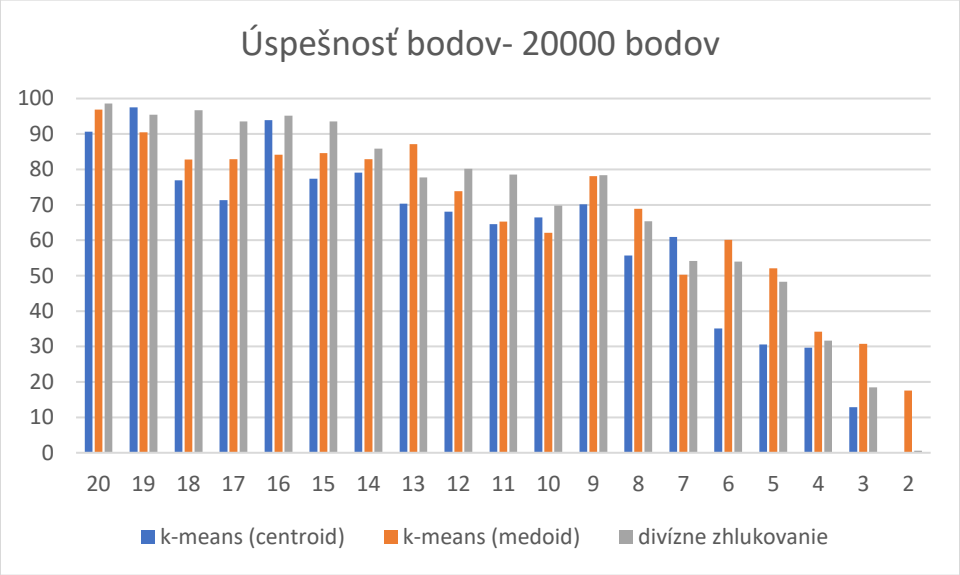
15000 bodov


Úspešnosť zhlukov- 15000 bodov
100
90
80
70
60
50
40
30
20
10
0
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2
k-means (centroid)
k-means (medoid)
divízne zhlukovanie


Úspešnosť bodov- 15000 bodov
100
90
80
70
60
50
40
30
20
10
0
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2
k-means (centroid)
k-means (medoid)
divízne zhlukovanie

20000 bodov


Úspešnosť zhlukov- 20000 bodov
100
90
80
70
60
50
40
30
20
10
0
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2
k-means (centroid)
k-means (medoid)
divízne zhlukovanie


Úspešnosť bodov- 20000 bodov
100
90
80
70
60
50
40
30
20
10
0
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2
k-means (centroid)
k-means (medoid)
divízne zhlukovanie

Zhodnotenie

Pri testovaní som zistil, že s klesajúcim *k,* klesá aj úspešnosť zlučovačov. Pri *k* = 11 začína úspešnosť klesať pri všetkých algoritmoch rýchlejšie. Medzi tromi porovnávanými algoritmami som nenašiel veľké rozdiely v úspešnosti. Najväčší nájdený rozdiel je, že k-means, kde stred je medoid, bol z týchto troch ďaleko najpomalší. K-means algoritmy by bolo možné optimalizovať lepším zvolením začiatočných centroidov a medoidov.