

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

Zadanie č. 4,b – Klastrovanie

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 viac 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 zhľukovač pre 2D priestor, ktorý zanalyzuje 2D priestor so všetkými jeho bodmi a rozdelí tento priestor na k zhľukov (klastrov). Implementujte rôzne verzie zhľukovača, konkrétne týmito algoritmami:

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

Vyhodnocujte úspešnosť/chybovosť vášho zhľukovača. Za úspešný zhľukovač 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 označujete (napr. vyfarbíte, očísľujete, 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, centroidov. 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 najmenšia. Toto program opakuje 100 krát alebo dokým sa zhluky prestanú meniť.

Divízne zhlukovanie, kde stred je centroid

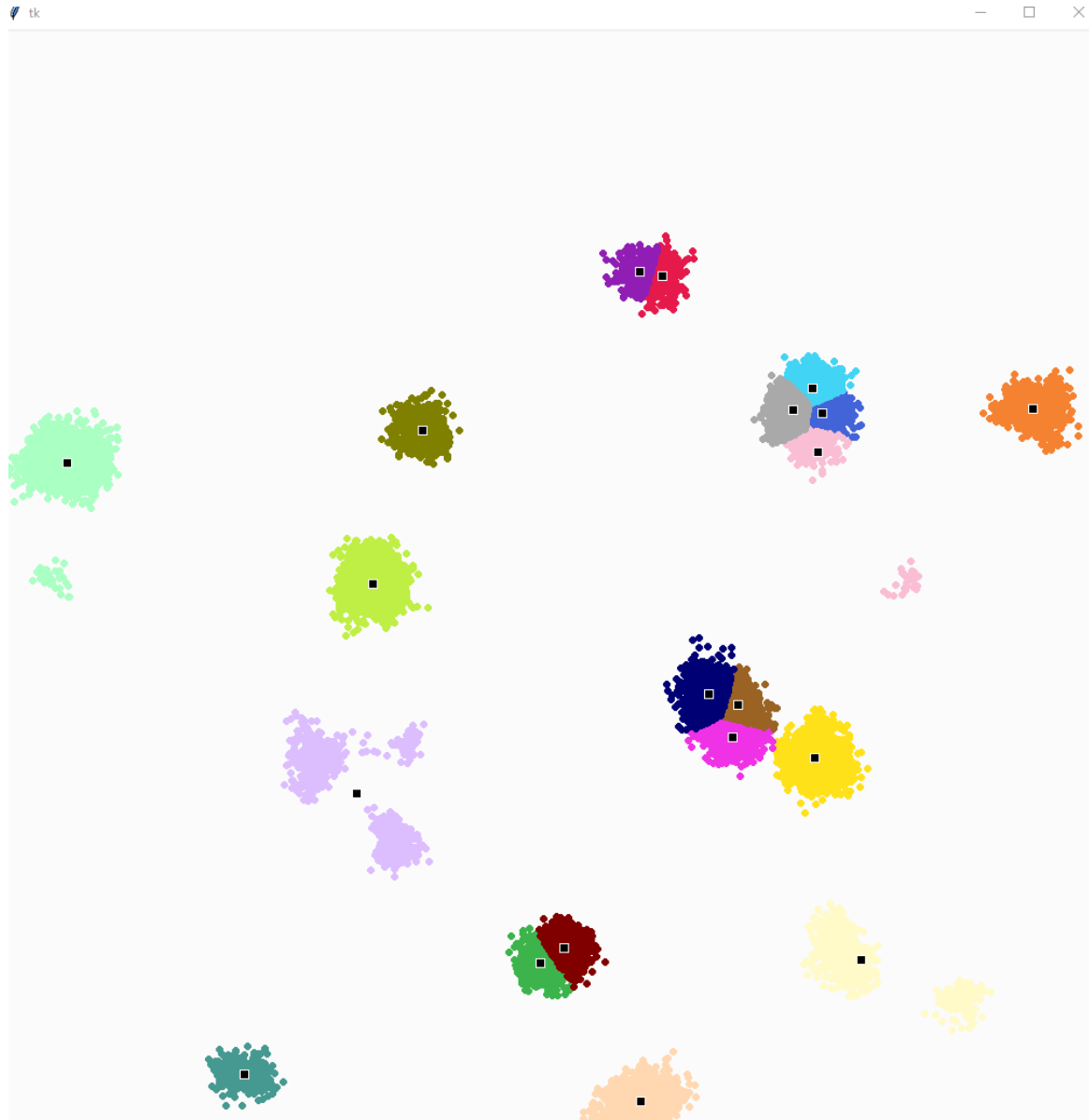
Všetky body sa na začiatku umiestia do jedného zhluku. Ten sa podľa k-means, kde stred je centroid, rozdelí na dva zhluky. Potom sa nájde zhluk, ktorý obsahuje najviac bodov a tiež sa podľa k-means rozdelí na dva. Toto sa opakuje n -krát, čo môže byť až dokým je každý bod vo vlastnom samostatnom zhluku.

Aglomeratívne zhlukovanie, kde stred je centroid

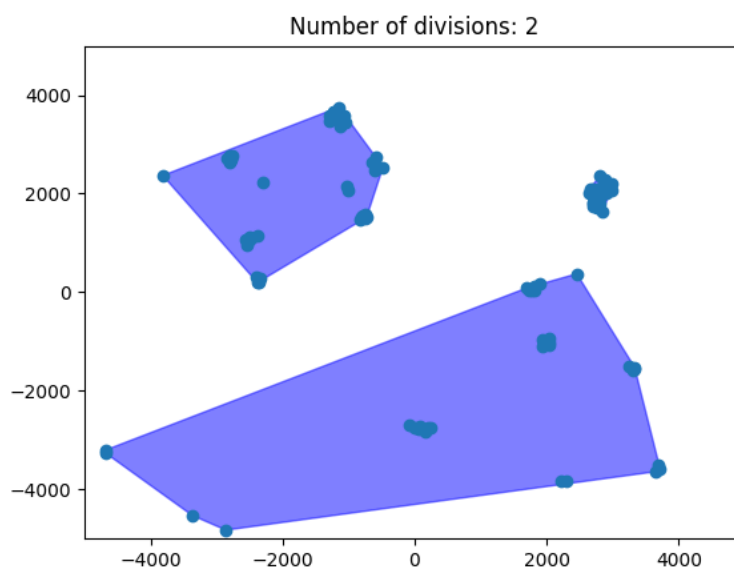
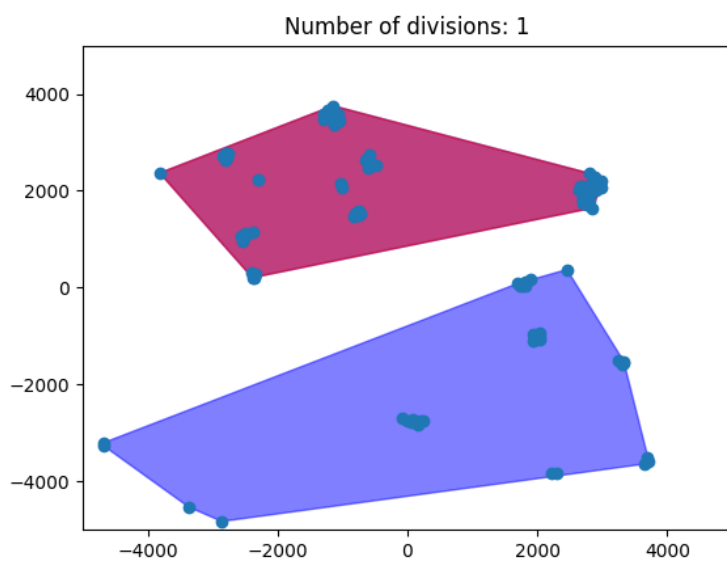
Tento algoritmus som implementoval pomocou matice vzdialeností. Na začiatku sa berie každý bod ako centroid svojho vlastného zhluku a vypočíta sa matica vzdialeností všetkých centroidov. Potom sa v nej nájde, ktoré centroidy sú k sebe najbližšie. Riadky a stĺpce týchto centroidov sa z matice odstránia 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

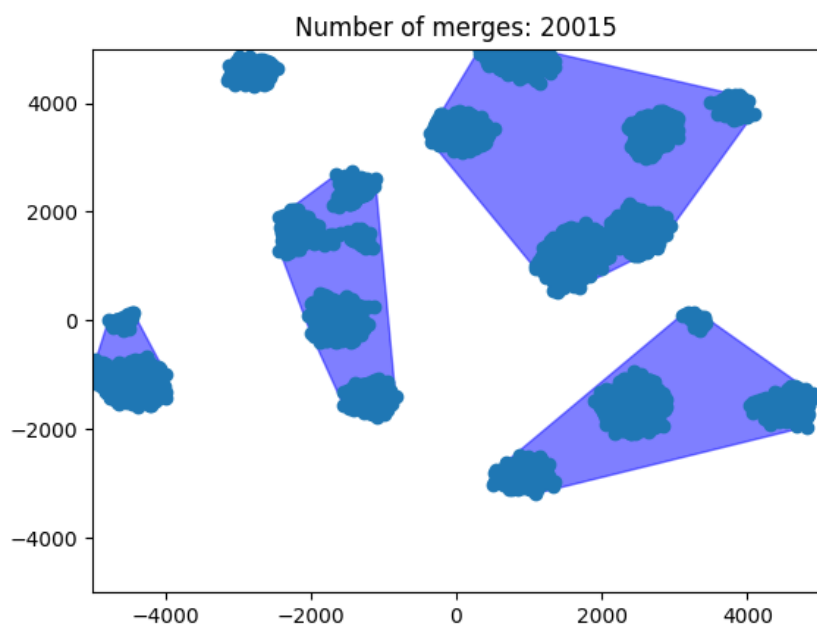
Algoritmy k-means vykresľujem pomocou modulu tkinter. Zhluky sú farebne odlíšené a centroidy alebo medoidy zhlukov sú vyznačené ako čierne štvorce.



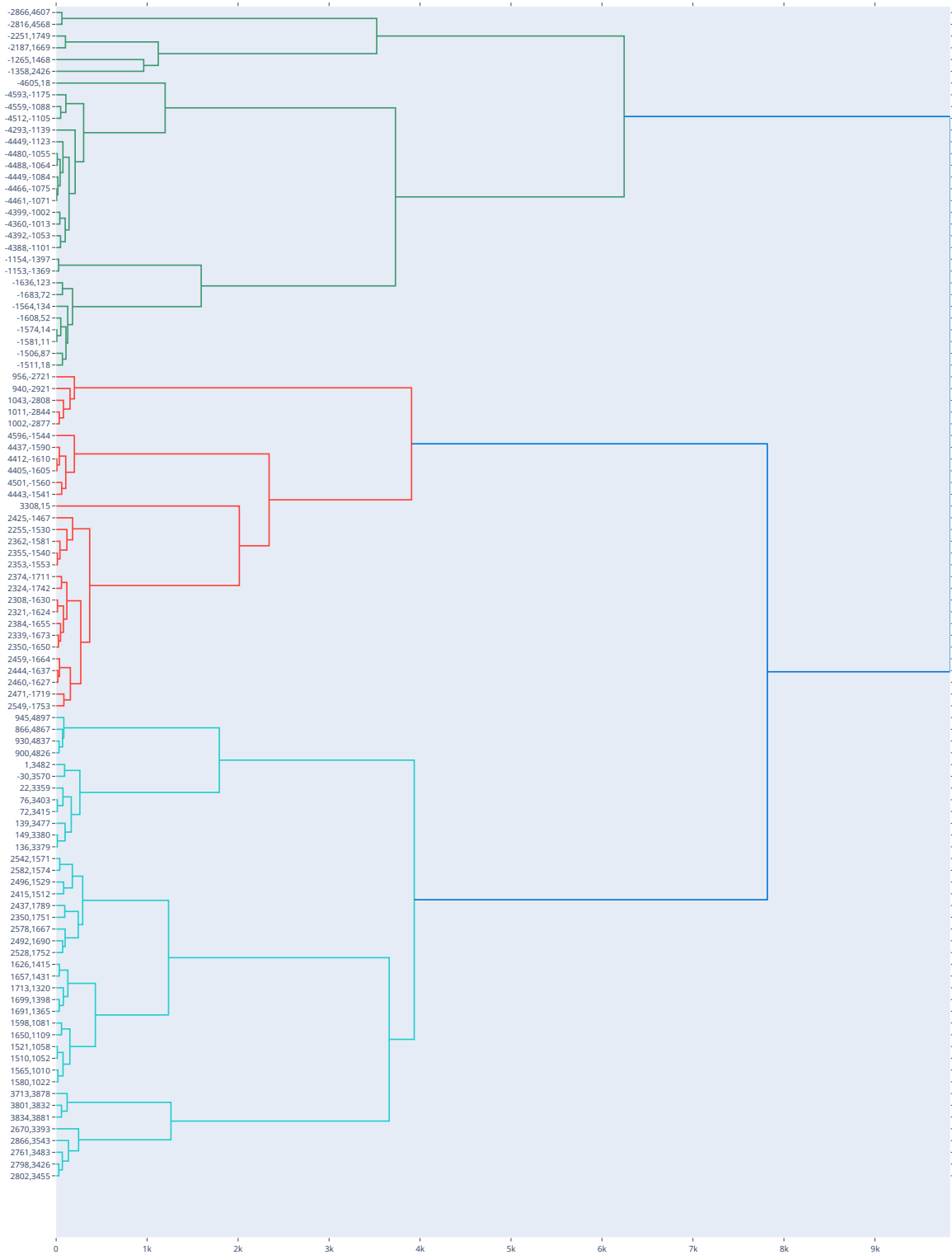
Divízne a aglomeratívne zhukovanie vykresľujem pomocou modulu matplotlib. Jednotlivé zhukky sú orámované a vyfarbené. Pri animácii divízneho zhukovania sa zhuk, ktorý sa bude rozdeľovať, farebne odlíši.



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



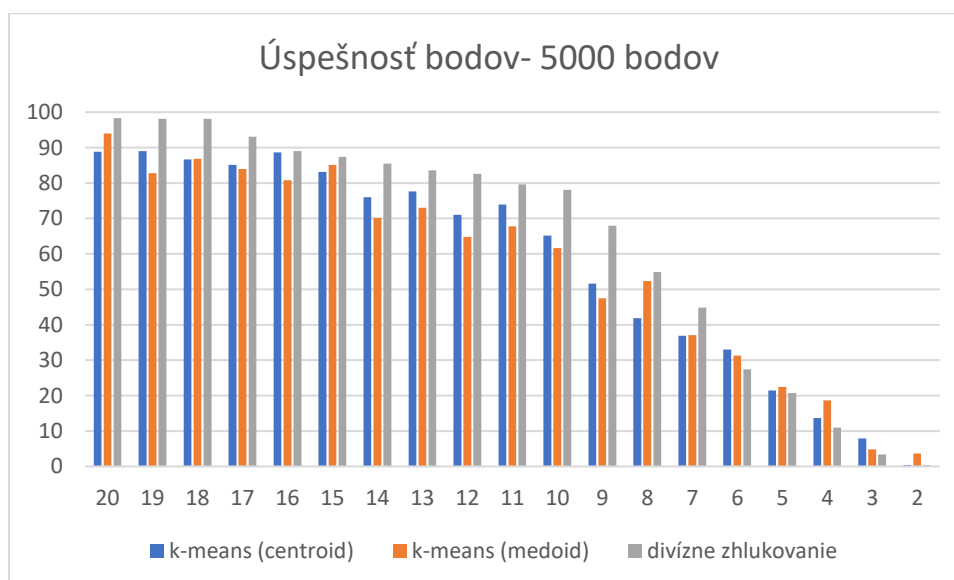
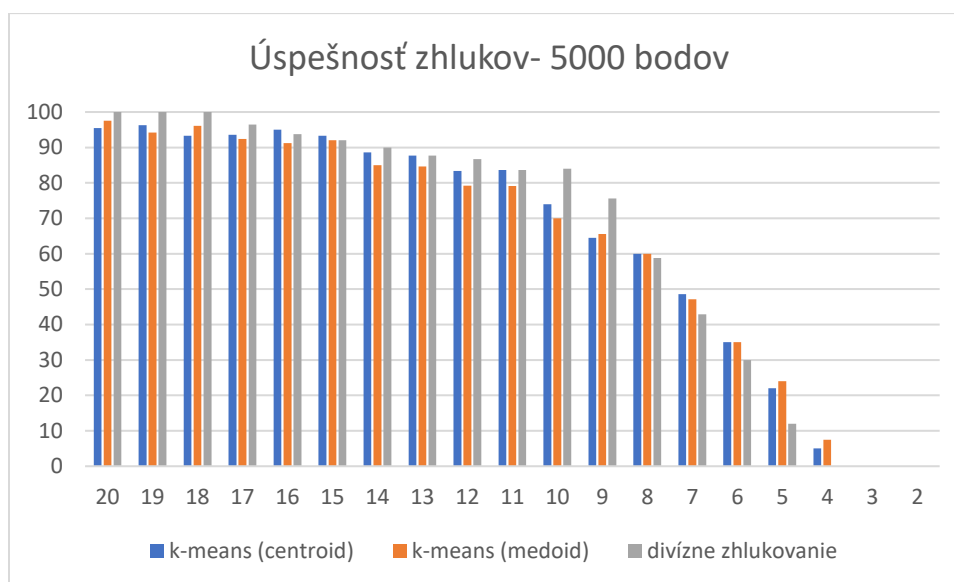
Dendrogram



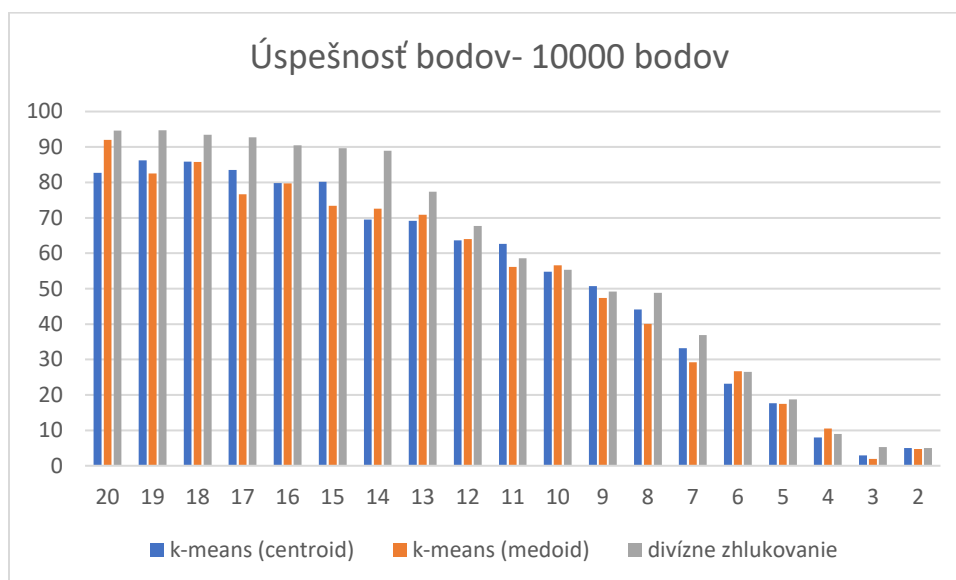
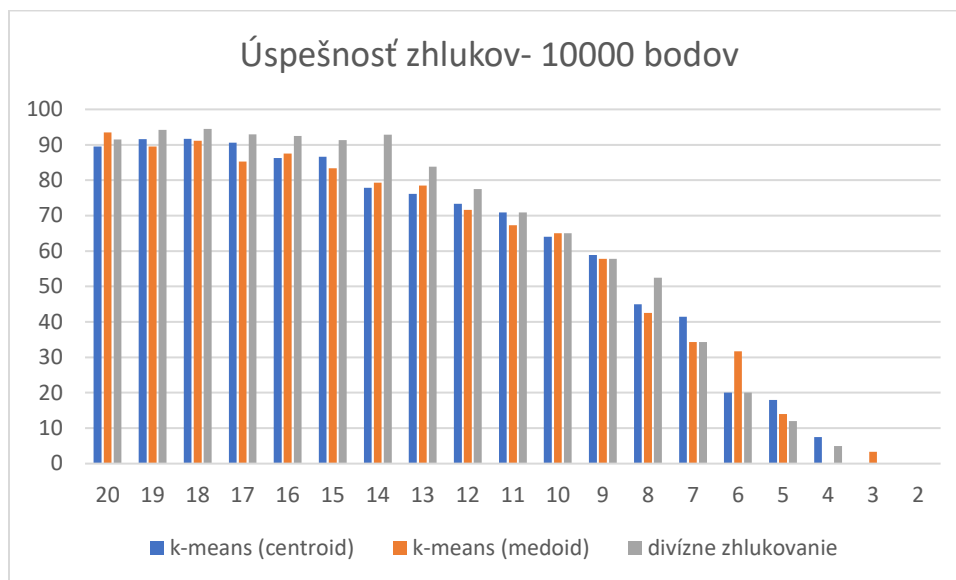
Testovanie

V testovaní som porovnával úspešnosť zlučovačov k-means a aglomeratívneho zhľukovania na jednej množine bodov, ktorú som postupne zväčšoval. Úspešnosť zhľukov som vyhodnocoval podľa toho, koľko percent zhľukov 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.

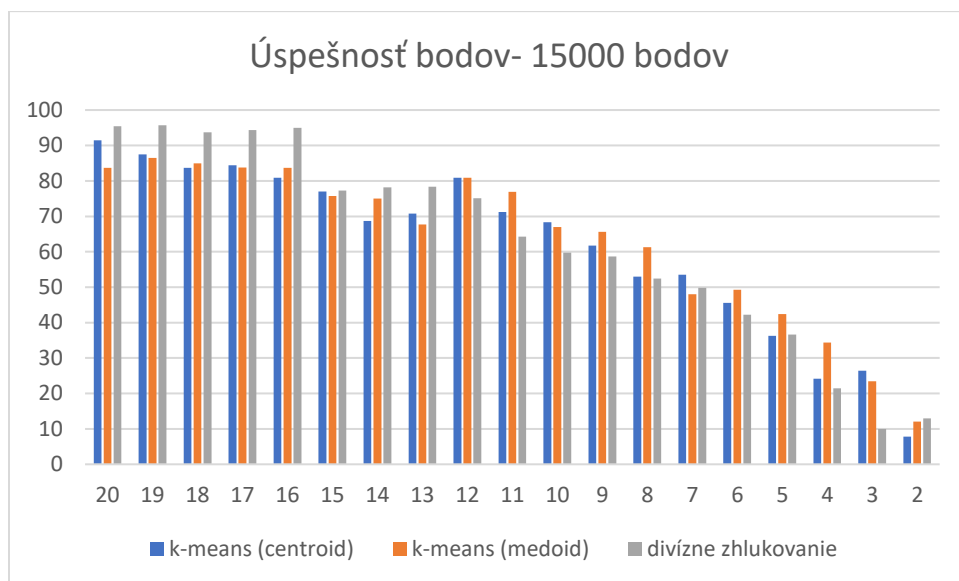
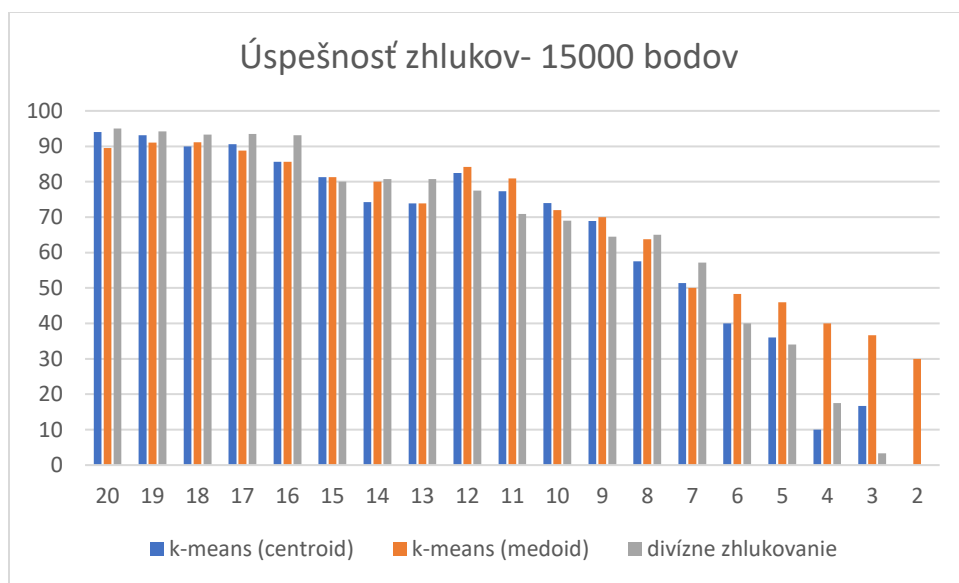
5000 bodov



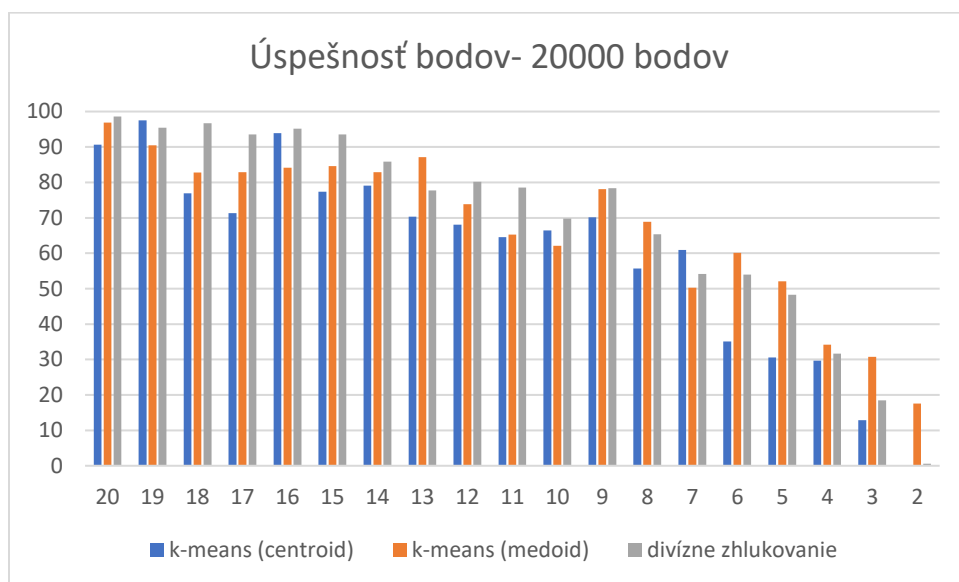
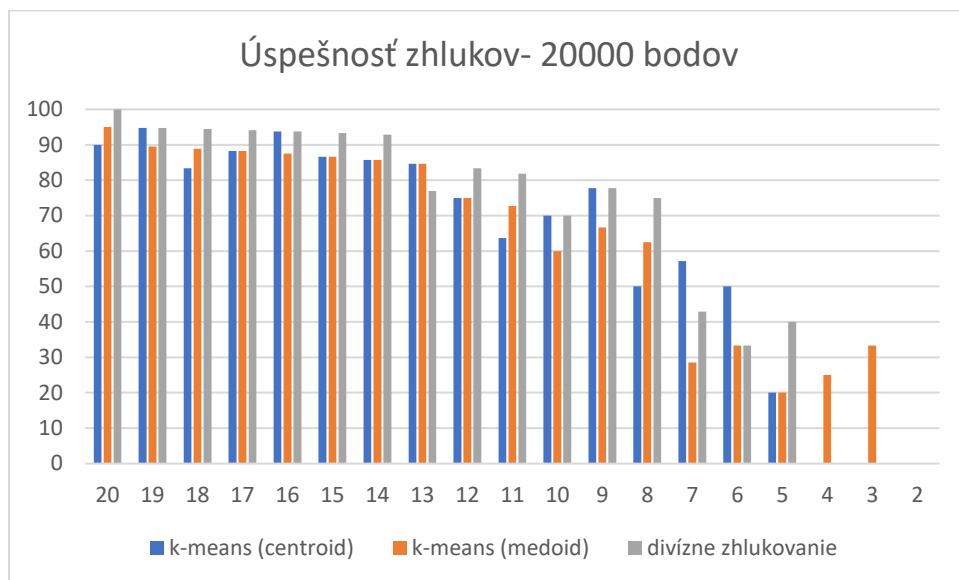
10000 bodov



15000 bodov



20000 bodov



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.