Slovenská Technická Univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 19 Bratislava 4

Richard Szarka

Klastrovanie

Zadanie č. 4

Prednášajúci: Ing. Lukáš Kohútka, PhD.

Cvičiaci: Ing. Boris Slíž

Cvičenie: Streda 15:00

1 Definícia problému

Máme body v 2D priestore -5000 po +5000. Tento priestor vyplníme 20 náhodnými bodmi. Ďalej generujeme ďalšie body nasledovne:

- Vybere sa jeden náhodný z posial vygenerovaných bodov
- Vygeneruje sa hodnota offset_X a offset_Y (-100, +100)
- Nový bod sa vypočíta ako staré súradnice sčítané s vygenerovanými offsetmi
- Ak je vybraný bod na kraji, tak sa interval offsetu zmenší

Správne klastrovanie je také, kde priemerná vzdialenosť od stredu je menšia ako 500.

2 Zadávanie vstupu

Do programu sa zadáva počet inicializačných bodov, počet klastrov na vytvorenie a nakoniec počet vygenerovaných bodov.

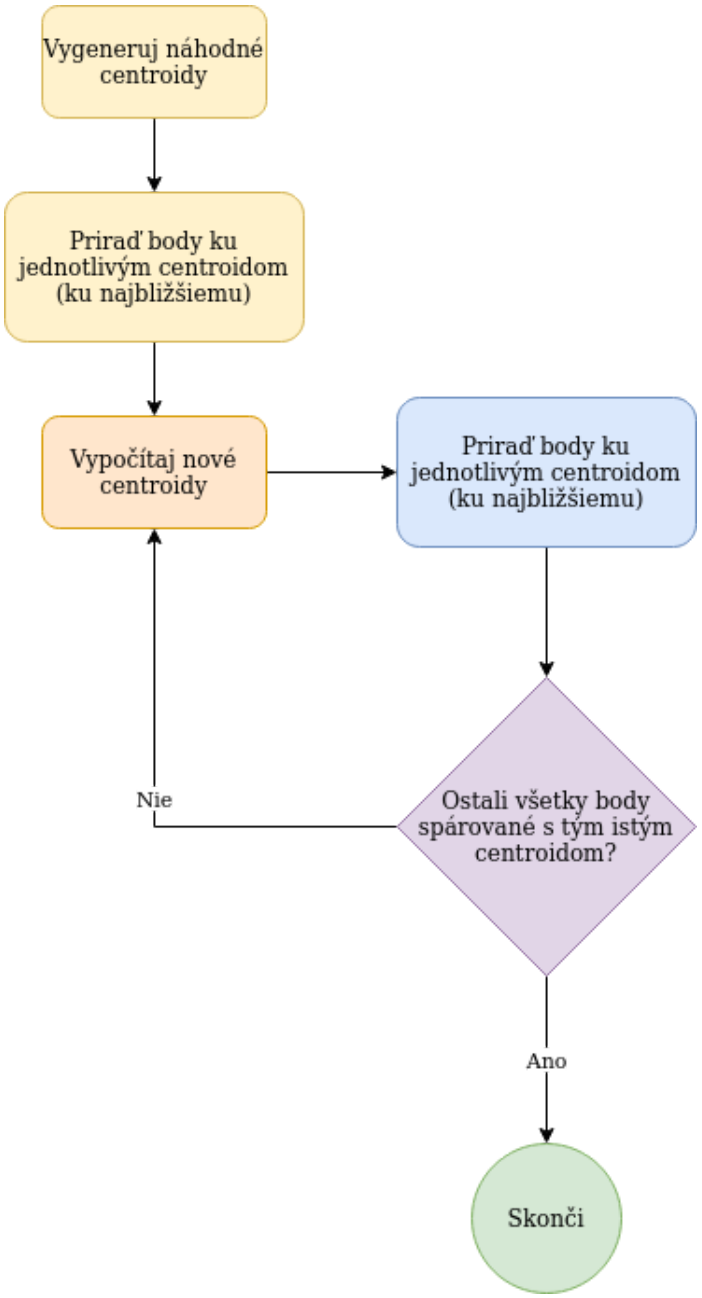
Príklad:

```
Generated init points: 5
How many clusters should be found? 5
Generated points: 1000
-------------------------
Kmeans centroid
12
Successful clusters: 4
Failed clusters: 1
-------------------------
Kmeans medoid
3
Successful clusters: 4
Failed clusters: 1
-------------------------
Divisions
5
Successful clusters: 4
Failed clusters: 1
-------------------------
Agglomerative
making matrix
5
Successful clusters: 5
Failed clusters: 0
6.893279075622559
```

Program následne vypíše výsledky jednotlivých algoritmov. Číslo pod Kmeans algoritmami je počet iterácií ktoré bolo potrebné vykonať kým sa stredy klastrov ustálili. Čísla pod aglomeratívnym a divisívnym vyznačujú progres vytvárania klastrov.

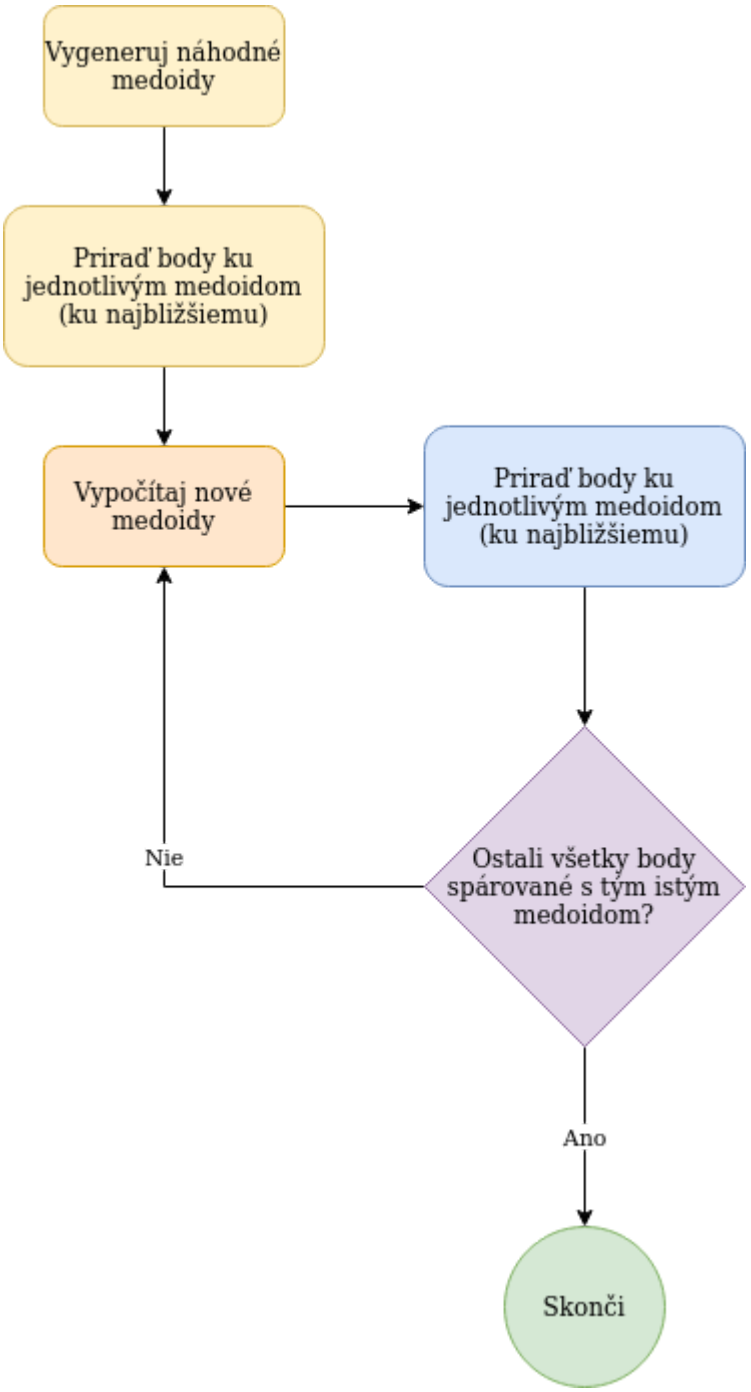
3 K-means so stredom v centroidoch

Princíp fungovania algoritmu K-means s centroidmi je znázornený na diagrame. Centroid definujeme ako priestorový stred daného klastra. Vypočíta sa ako súčet X a Y súradníc a následne ich vydelenie počtu bodov v klastri.



4 K-means so stredom v medoidoch

Princíp fungovania algoritmu K-means s medoidmi je znázornený na diagrame. Medoid definujeme ako bod v klastri, ktorý má najmenšiu priemernú vzdialenosť ku všetkým ostatným bodom v klastri. Týmto spôsobom vieme zaručiť, že stred klastra nebude v “strede ničoho” ako je to pri centroidoch. Princíp fungovania algoritmu je znázornený na diagrame.



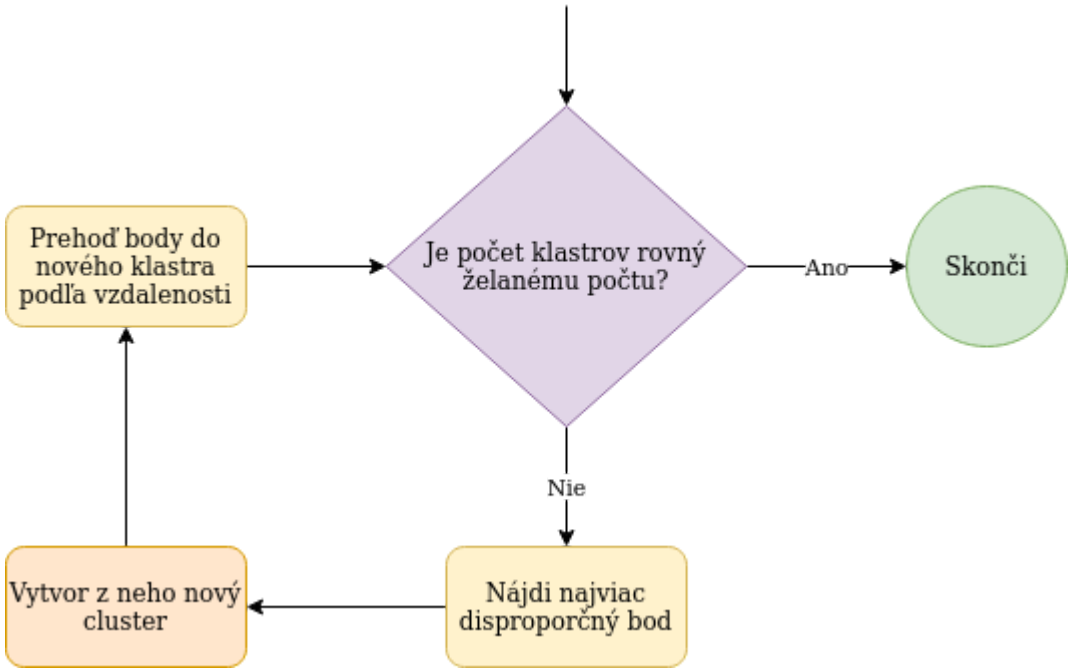
5 Divizívne zhlukovanie

Divizívne zhlukovanie je založené na princípe, že na začiatku máme jeden obrovský klaster so všetkými bodmi v ňom. Vypočítame, ktorý bod má najväčšiu priemernú vzdialenosť od všetkých bodov v danom klastri. Ten bod vyberieme z daného klastra a vytvoríme z neho nový klaster. Všetky body bližšie ku danému bodu pridáme do nového klastra a všetky body bližšie ku centroidu starého klasra necháme v starom.

Takto delíme klastre kým nemáme želaný počet klastrov. Hlavné jadro algoritmu je nasledujúci cyklus:

```
while len(clusters) != k:   # kým nie je počet klastrov K

   cluster_todivide, e1 = find_largestD(clusters)   # najdi najviac
disproporčný bod
   clusters = divide(clusters, cluster_todivide, e1)   # vytvor nový cluster
```



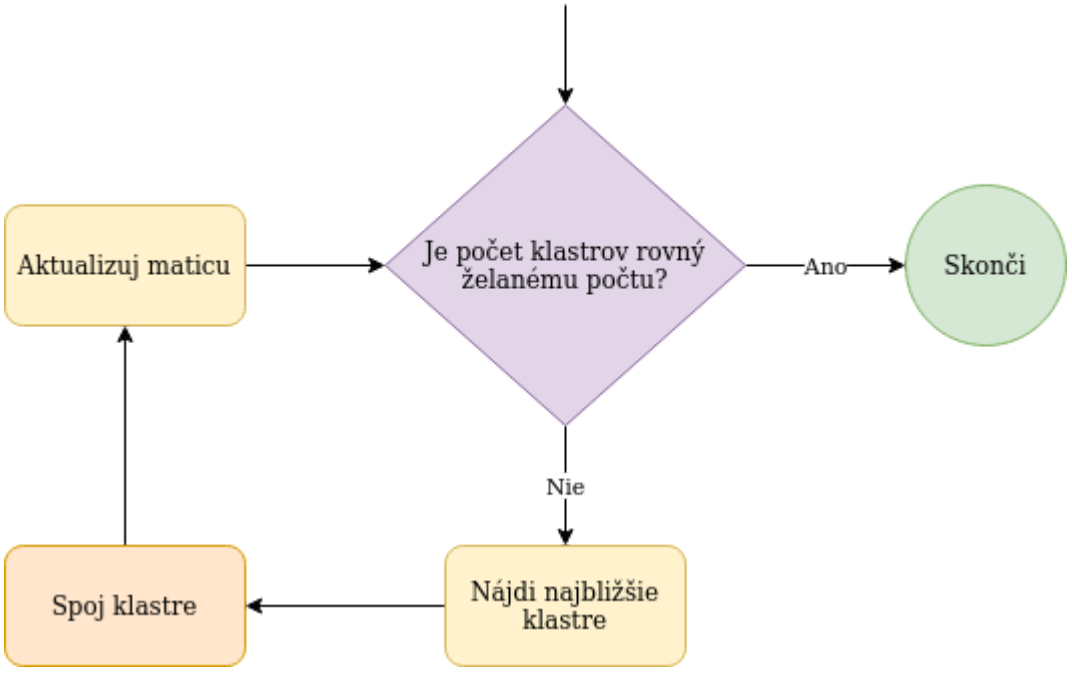
6 Aglomeratívne zhlukovanie

Aglomeratívne zhlukovanie je založené na “opačnom” princípe ako je divizívne. Na začiatku sú všetky body samostatné klastre. Vytvorí sa matica vzdialeností každého jedného bodu. V matici sa následne nájde najmenšia vzdialenosť bodov. Dané dva body sa zlúčia do jedného klastra a z matice sa vymaže záznam druhého z bodov a prvého záznam sa aktualizuje (viacbodové klastre reprezentuje ich centroid). Matica sa redukuje a klastre sa spájajú kým nemáme želaný počet klastrov.

Hlavné jadro funkcie vyzerá nasledovne:

```
while len(matrix) != k:      # opakuj kým nemáš potrebný počet klastrov

   cluster1, cluster2 = minimum_distance(matrix)    # nájdi najmenšiu
vzdialenosť
   data, matrix = merge(data, matrix, centroids, cluster1, cluster2)    #
spoj
```



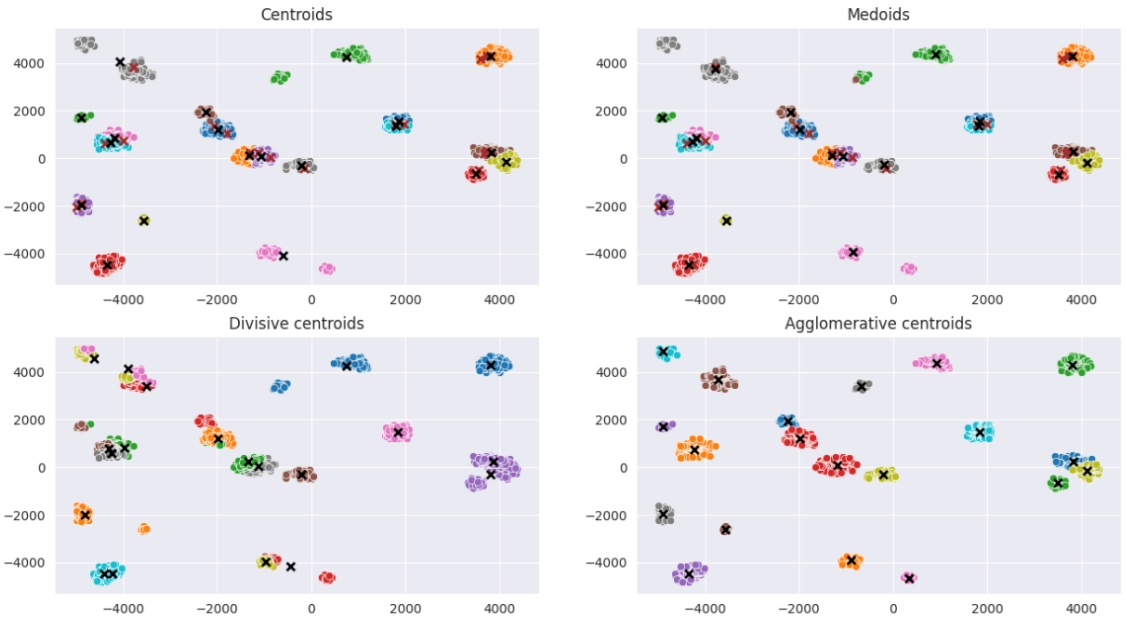
7 Testovanie

V testovaní čierne X značí koncový centroid/medoid a hnedý značí počiatočný.

Prípad s 5000 bodmi:

```
Generated init points: 20
How many clusters should be found? 20
Generated points: 5000
-------------------------
Kmeans centroid
28
Successful clusters: 18
Failed clusters: 2
-------------------------
Kmeans medoid
3
Successful clusters: 19
Failed clusters: 1
-------------------------
Divisions
20
Successful clusters: 16
Failed clusters: 4
-------------------------
Agglomerative
making matrix
20
Successful clusters: 20
Failed clusters: 0
1180.7883338928223

Process finished with exit code 0
```

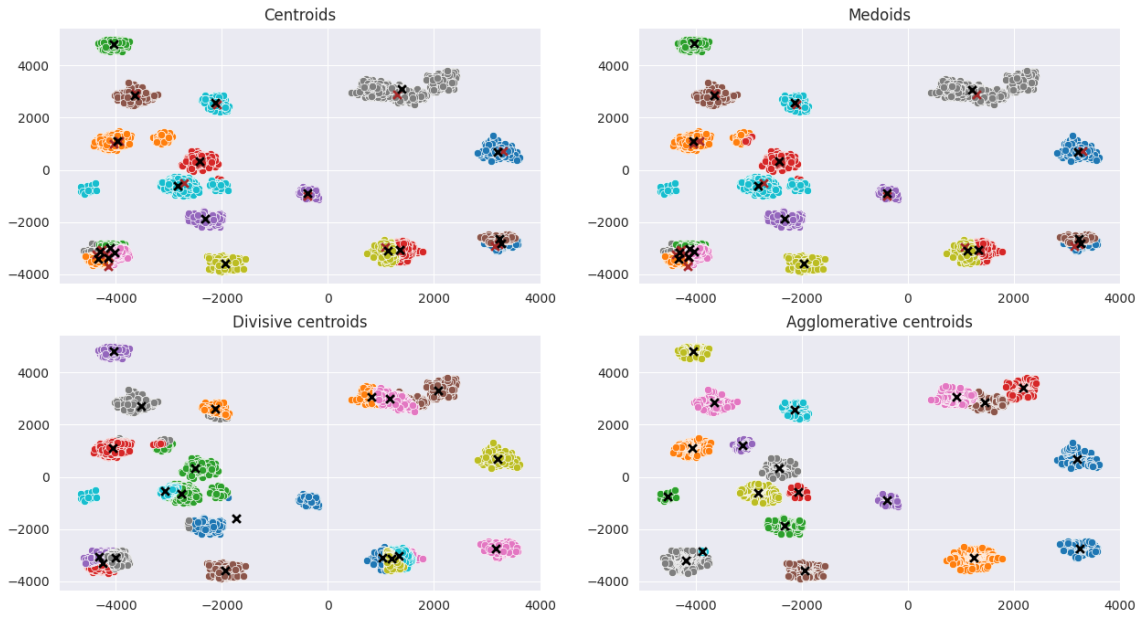


V tomto prípade vidíme, že jediné dokonale úspešné bolo aglomeratívne a najmenej úspešné bolo divizívne.

Prípad s 10000 bodmi:

```
Generated init points: 20
How many clusters should be found? 20
Generated points: 10000
-------------------------
Kmeans centroid
20
Successful clusters: 19
Failed clusters: 1
-------------------------
Kmeans medoid
3
Successful clusters: 19
Failed clusters: 1
-------------------------
Divisions
20
Successful clusters: 19
Failed clusters: 1
-------------------------
Agglomerative
making matrix
20
Successful clusters: 20
Failed clusters: 0
8971.31680226326

Process finished with exit code 0
```

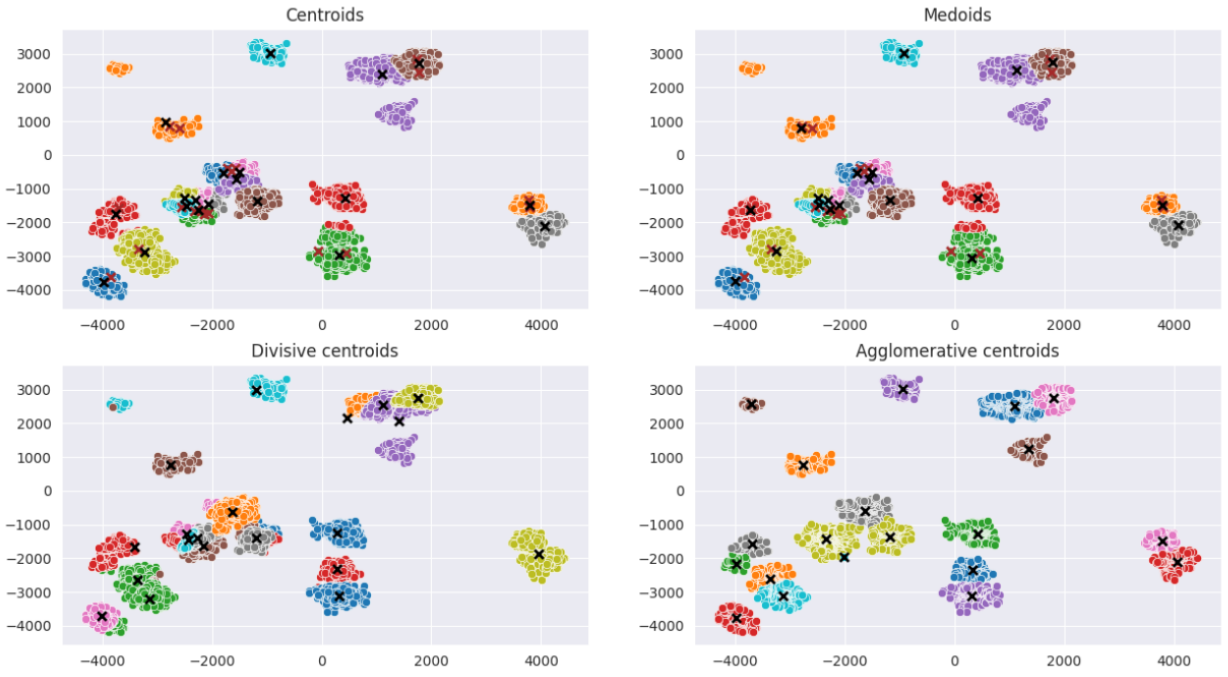


V tomto prípade boli rovnako úspešné všetky algoritmy okrem aglomeratívneho. Ten mal zase 100% úspešnosť clusterov. Ostatné algoritmi mali 95%.

Prípad s 15000 bodmi:

```
Generated init points: 20
How many clusters should be found? 20
Generated points: 15000
-------------------------
Kmeans centroid
31
Successful clusters: 20
Failed clusters: 0
-------------------------
Kmeans medoid
5
Successful clusters: 20
Failed clusters: 0
-------------------------
Divisions
20
Successful clusters: 16
Failed clusters: 4
-------------------------
Agglomerative
making matrix
20
Successful clusters: 20
Failed clusters: 0
25763.501668691635

Process finished with exit code 0
```

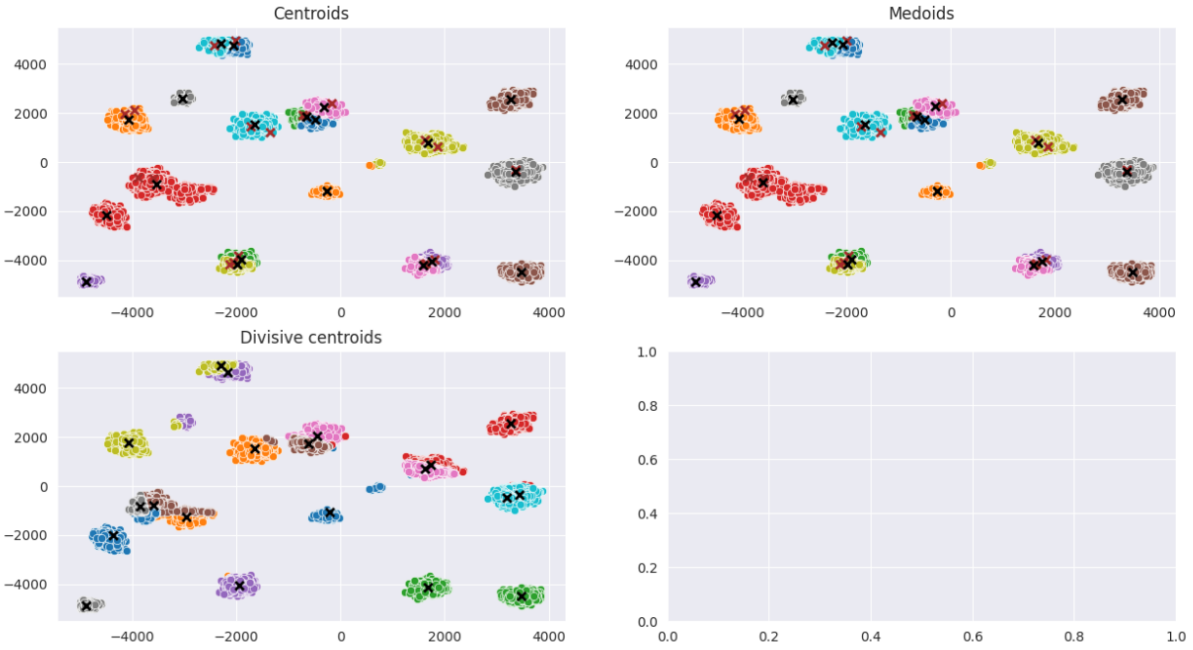


V prípade s 15000 bodmi boli 100% úspešné všetky okrem divizívneho. Ten mal 80% úspšenosť.

Prípad s 20000 bodmi:

```
Generated init points: 20
How many clusters should be found? 2
Generated points: 20000
-------------------------
Kmeans centroid
16
Successful clusters: 20
Failed clusters: 0
-------------------------
Kmeans medoid
3
Successful clusters: 20
Failed clusters: 0
-------------------------
Divisions
20
Successful clusters: 20
Failed clusters: 0

Process finished with exit code 0
```



V tomto prípade nebolo možné spustiť aglomerativny, kvôli nedostatku pamäte a tým pádom aj veľkej časovej náročnosti. Všetky ostatné algoritmy mali 100% úspešnosť.

8 Záver

V tomto zadaní sme mali naprogramovať známe klastrovacie algoritmy. Z testovania jednoznačne vyplýva, že centroid a medoid sú algoritmy pomocou ktorých je klastrovanie pomerne dobre, ale ak si zvolia zlé počiatočné inicializačné body, klastrovanie vôbec nemusí dopadnúť dobre. Divizívny algoritmus bol v našich testoch najmenej úspešnejší a to najmä kvôli jeho vlastnosti deliť veľké ucelené klastre. Jednoznačný víťaz bol aglomerativny algoritmus. Počas testovania aj mimo tejto dokumentácie, nenastal prípad kedy by klastre neodhadol na dokonalú úroveň. Jediná nevýhoda použitia daného algoritmu je obrovská časová zložitosť

Zadanie som napísal v programovacom jazyku python a na zobrazovanie grafov som použil knižnicu seaborn, matplotlib a numpy.