

**České vysoké učení v Praze
Fakulta stavební**

155ADKG - Algoritmy digitální kartografie

Úloha 2: Konvexní obálky

Radek Novotný

Zadání:

Úloha č. 2: Konvexní obálky a jejich konstrukce

Vstup: množina $P = \{p_1, \dots, p_n\}$, $p_i = [x, y_i]$.

Výstup: $\mathcal{H}(P)$.

Nad množinou P implementujte následující algoritmy pro konstrukci $\mathcal{H}(P)$:

- Jarvis Scan,
- Quick Hull,
- Sweep Line.

Vstupní množiny bodů včetně vygenerovaných konvexních obálek vhodně vizualizujte. Pro množiny $n \in \langle 1000, 1000000 \rangle$ vytvořte grafy ilustrující doby běhu algoritmů pro zvolená n . Měření proveďte pro různé typy vstupních množin (náhodná množina, rastr, body na kružnici) opakovaně (10x) a různá n (nejméně 10 množin) s uvedením rozptylu. Naměřené údaje uspořádejte do přehledných tabulek.

Zamyslete se nad problematikou možných singularit pro různé typy vstupních množin a možnými optimalizacemi. Zhodnoťte dosažené výsledky. Rozhodněte, která z těchto metod je s ohledem na časovou složitost a typ vstupní množiny P nejvhodnější.

Hodnocení:

Krok	Hodnocení
Konstrukce konvexních obálek metodami Jarvis Scan, Quick Hull, Sweep Line.	15b
Konstrukce konvexní obálky metodou Graham Scan	+5b
Konstrukce striktně konvexních obálek pro všechny uvedené algoritmy.	+5b
Ošetření singulárního případu u Jarvis Scan: existence kolineárních bodů v datasetu.	+2b
Konstrukce Minimum Area Enclosing box některou z metod (hlavní směry budov).	+5b
Algoritmus pro automatické generování konvexních/nekonvexních množin bodů různých tvarů (kruh, elipsa, čtverec, star-shaped, popř. další).	+4b
Max celkem:	36b

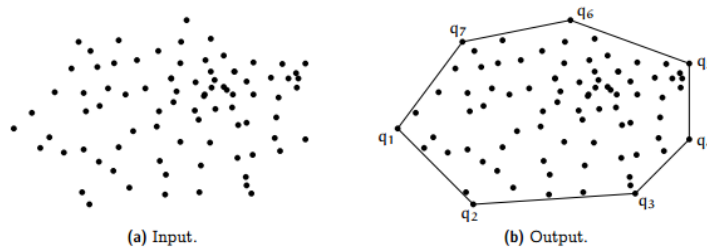
Čas zpracování: 3 týdny.

Bonusové úlohy:

V rámci bonusových úloh bylo implementováno ošetření singulárního případu u Jarvis Scan. Také byly vytvořeny algoritmy pro automatické generování množin bodů – kruhu, elipsy, čtverce a obdélníku.

Popis problému:

Konvexní obálka H dané množiny M , jest taková množina bodů, že po spojení všech bodů ohraničuje vzniklý mnohoúhelník všechny body dané množiny M .



Obr. 1

K sestrojení konvexní obálky se nejčastěji používají následující metody:

- Jarvis Scan
- Quick Hull
- Zametací přímka
- Graham Scan
- Inkrementální konstrukce
- Metoda „Rozděl a panuj“

První tři metody byly v této práci implementovány a jsou dále podrobněji popsány.

Popis algoritmů:

Jarvis Scan (Gift Wrapping)

Po setřídění množiny je vybrán pivot q , jehož souřadnice Y (případně X) je minimální. Jde o to, aby byl pivot nejkrajnějším bodem množiny, dále budeme v popisu uvažovat bod, jehož Y souřadnice je minimální.

Poté je zvolena spojnice rovnoběžná s osou X . Následuje cyklus, kdy pro každý přidáváme do konvexní obálky body, jejichž spojnice svírá se známou spojnicí největší úhel. Tento cyklus končí tehdy, když algoritmus znovu narazí na pivota.

Algoritmus: Jarvis Scan (P, H)

- 1: Nalezení pivota q , $q = \min(y_i)$
- 2: Přidej $q \rightarrow H$
- 3: Inicializuj: $p_{j-1} \in X$, $p_j = q$, $p_{j+1} = p_{j-1}$
- 4: Opakuj, dokud $p_{j+1} \neq q$:
- 5: Nalezní $p_{j+1} = \arg \max_{p_i \in P} \angle(p_{j-1}, p_j, p_i)$
- 6: Přidej $p_{j+1} \rightarrow H$.
- 7: $p_{j-1} = p_j$; $p_j = p_{j+1}$



Obr.2

Problematická situace:

Singulární případ:

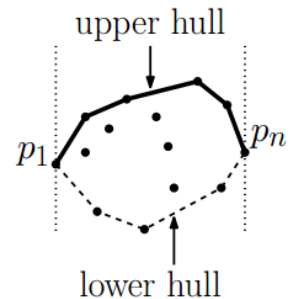
Problémem je singulární případ, pokud se vyskytují kolineární body na konvexní obálce. Tento problém je ošetřen přidáním bodu, jež má maximální vzdálenost od posledního bodu konvexní obálky. To jest, pokud máme tři kolineární body, přidáváme vždy ten nejvzdálenější.

Více bodů s minimální souřadnicí:

Dalším problémem může být vybrání pivota, pokud máme dva nebo více bodů s minimální Y souřadnicí. Tento problém lze řešit další podmínkou, vybereme tedy z těchto bodů, ten s maximální X souřadnicí.

Quick Hull

Jedná se o rekurzivní algoritmus, jež rozděljuje řešení na dvě poloviny. Nejprve přímkou určenou body s nejnižší a s nejvyšší souřadnicí ve zvoleném směru. Dále vždy najde nejvzdálenější bod na každé straně a zjišťuje, zda existuje další bod za touto spojnici, pokud ano, opakuje se řešení s tímto bodem. Nejprve probíhá výpočet pro horní a následně pro spodní polovinu, aby docházelo ke správnému řazení bodů. Nakonec dojde ke spojení bodů z obou polovin.



Obr.3

Algoritmus: Quick Hull (S, H), globální procedura

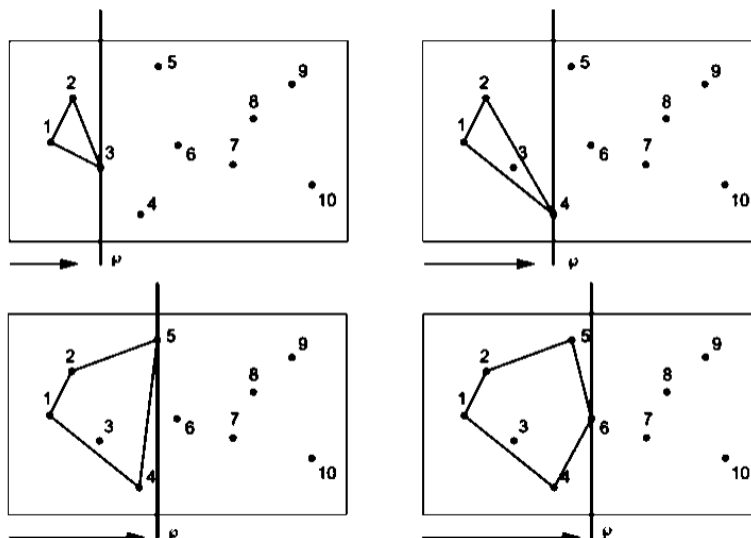
```
1:  $H = \emptyset, S_U = \emptyset, S_L = \emptyset$ 
2:  $q_1 = \min_{p_i \in S} (x_i), q_3 = \max_{p_i \in S} (x_i)$ 
3:  $S_U \leftarrow q_1, S_U \leftarrow q_3$ 
4:  $S_L \leftarrow q_1, S_L \leftarrow q_3$ 
5: for  $\forall p_i \in S$ 
6:   if  $(p_i \in \sigma_1(q_1, q_3)) S_U \leftarrow p_i$ 
7:   else  $S_L \leftarrow p_i$ 
8:  $H \leftarrow q_3$ . //Přidej bod c do H
9: Quick Hull(1, 0,  $S_U, H$ ) //Upper Hull
10:  $H \leftarrow q_1$ . //Přidej bod c do H.
11: Quick Hull(0, 1,  $S_L, H$ ) //Lower Hull
```

Algoritmus: Quick Hull (s, e, S, H), lokální procedura

```
1: Najdi bod  $p = \arg \max_{p_i \in S} ||p_i - (p_s, p_e)||, p \in \sigma_r(p_s, p_e)$ .
2: If  $p \neq O$  ; //Existuje bod vpravo od hrany.
3: Quick Hull(s, i, S, H) //Upper Hull, i index p
4:  $H \leftarrow p$ . //Přidej bod c do H.
5: Quick Hull(i, e, S, H) //Lower Hull, i index p
```

Sweep Line

Metoda zametací přímky (anglicky Sweep Line) prochází body postupně podle určené souřadnice a tvoří obálku pomocí metody inkrementální konstrukce. Body jsou nejprve seříděny a poté se všechny postupně vyhodnocují. Vyhodnocování probíhá na základě polohy vůči tečně současné konvexní obálky, z té zjišťujeme, zda se bod nachází uvnitř nebo je její součástí.



Obr.4

Problematické situace:

Duplicitní body:

Problém může nastat, pokud budou na vstupu duplicitní body. Při výpočtu vznikají singularity, které zabrání správnému průběhu algoritmu. Tento problém je nutno ošetřit hned na začátku odstraněním všech duplicitních bodů.

Algoritmus 1: Sweep Line (P, H)

```
1: unique(P) //Odstraň duplicitní body
2: Sort Ps = sort(P) by x
3: H(Ps(3)) ← 4(p1, p2, p3) //Iniciální řešení pro i ≤ 3
4: i ← 4.
5: Opakuj pro ∀pi ∈ Ps
6: H(Ps(m)) = H(Ps(m - 1)) + ΔH(pi) //Aktualizace řešení
```

Výsledky:

Jarvis Scan:

Náhodná množina										
	1000	2500	5000	10000	25000	50000	100000	250000	500000	1000000
1	0.002	0.004	0.004	0.004	0.009	0.023	0.069	0.066	1.161	0.199
2	0.001	0.000	0.002	0.003	0.011	0.030	0.057	0.289	1.262	4.043
3	0.001	0.001	0.002	0.004	0.010	0.024	0.084	0.296	1.211	4.309
4	0.001	0.000	0.001	0.003	0.008	0.018	0.077	0.346	1.204	0.213
5	0.000	0.001	0.001	0.003	0.009	0.024	0.070	0.378	1.072	4.753
6	0.000	0.000	0.002	0.003	0.010	0.021	0.083	0.375	1.175	0.240
7	0.000	0.001	0.002	0.003	0.009	0.027	0.087	0.362	1.323	0.213
8	0.000	0.001	0.001	0.004	0.010	0.028	0.022	0.320	1.146	3.957
9	0.000	0.001	0.001	0.004	0.010	0.027	0.045	0.351	1.348	4.626
10	0.000	0.000	0.001	0.004	0.010	0.023	0.066	0.290	1.153	4.160
Grid										
	1000	2500	5000	10000	25000	50000	100000	250000	500000	1000000
1	0.000	0.016	0.009	0.024	0.054	0.099	0.254	0.554	1.047	2.108
2	0.002	0.000	0.015	0.015	0.067	0.094	0.406	0.646	1.241	2.092
3	0.000	0.000	0.015	0.032	0.046	0.113	0.296	0.607	1.044	2.095
4	0.000	0.016	0.015	0.047	0.067	0.206	0.383	0.617	1.199	2.080
5	0.000	0.004	0.000	0.015	0.046	0.094	0.395	0.597	1.065	2.093
6	0.000	0.016	0.016	0.032	0.056	0.098	0.244	0.583	1.051	2.101
7	0.003	0.005	0.010	0.016	0.047	0.094	0.247	0.624	1.057	2.097
8	0.000	0.000	0.016	0.031	0.047	0.093	0.235	0.585	1.062	2.104
9	0.016	0.000	0.016	0.031	0.051	0.094	0.266	0.608	1.276	2.098
10	0.000	0.015	0.015	0.016	0.072	0.094	0.242	0.598	1.063	2.087

Kružnice										
	1000	2500	5000	10000	25000	50000	100000	250000	500000	1000000
1	0.098	0.091	0.187	0.445						
2	0.032	0.114	0.170	0.397						
3	0.023	0.088	0.237	0.398						
4	0.029	0.086	0.996	0.381						
5	0.016	0.144	0.719	0.457						
6	0.031	0.125	0.167	0.472						
7	0.017	0.068	0.203	0.376						
8	0.032	0.127	0.156	0.395						
9	0.016	0.078	0.223	0.395						
10	0.032	0.122	0.168	0.449						

Quick Hull:

Náhodná množina										
	1000	2500	5000	10000	25000	50000	100000	250000	500000	1000000
1	0.005	0.000	0.000	0.019	0.016	0.053	0.085	0.363	0.439	1.076
2	0.000	0.000	0.000	0.000	0.019	0.047	0.078	0.239	0.487	0.891
3	0.000	0.000	0.016	0.016	0.031	0.046	0.094	0.218	0.447	1.015
4	0.002	0.016	0.015	0.015	0.033	0.047	0.141	0.219	0.458	0.920
5	0.000	0.015	0.016	0.015	0.031	0.047	0.094	0.232	0.438	0.891
6	0.000	0.000	0.000	0.000	0.031	0.047	0.098	0.244	0.575	0.888
7	0.001	0.003	0.000	0.000	0.031	0.047	0.087	0.237	0.526	0.899
8	0.000	0.000	0.016	0.010	0.031	0.047	0.116	0.220	0.452	0.902
9	0.000	0.000	0.000	0.016	0.032	0.043	0.094	0.218	0.458	0.891
10	0.000	0.000	0.000	0.000	0.015	0.047	0.109	0.259	0.607	0.954

Grid										
	1000	2500	5000	10000	25000	50000	100000	250000	500000	1000000
1	0.000	0.000	0.000	0.000	0.016	0.031	0.062	0.281	0.349	0.779
2	0.000	0.000	0.000	0.000	0.015	0.031	0.070	0.168	0.357	0.786
3	0.000	0.002	0.002	0.006	0.016	0.031	0.063	0.419	0.516	0.779
4	0.000	0.000	0.000	0.000	0.016	0.031	0.197	0.156	0.356	0.769
5	0.000	0.000	0.000	0.016	0.057	0.031	0.052	0.226	0.522	0.784
6	0.000	0.000	0.000	0.000	0.016	0.032	0.062	0.171	0.517	0.777
7	0.000	0.000	0.015	0.016	0.015	0.031	0.063	0.156	0.484	0.777
8	0.000	0.015	0.000	0.000	0.022	0.031	0.235	0.152	0.355	0.773
9	0.000	0.000	0.000	0.000	0.030	0.031	0.062	0.157	0.515	0.779
10	0.000	0.000	0.016	0.000	0.016	0.031	0.066	0.156	0.504	0.795

Kružnice										
	1000	2500	5000	10000	25000	50000	100000	250000	500000	1000000
1	0.000	0.000	0.016	0.015	0.063	0.166	0.286	1.008	1.325	3.395
2	0.000	0.016	0.016	0.047	0.139	0.342	0.340	1.213	1.369	4.640
3	0.000	0.000	0.000	0.047	0.081	0.344	0.471	0.988	2.354	4.688
4	0.000	0.016	0.031	0.019	0.172	0.396	0.266	0.545	2.247	3.023
5	0.015	0.000	0.019	0.027	0.109	0.360	0.390	0.850	1.527	2.906
6	0.003	0.007	0.000	0.031	0.110	0.172	0.406	0.821	2.499	3.151
7	0.000	0.000	0.010	0.031	0.128	0.095	0.563	0.742	1.757	3.266
8	0.000	0.000	0.000	0.031	0.059	0.363	0.278	0.973	2.297	4.396
9	0.000	0.000	0.020	0.047	0.098	0.219	0.541	0.676	2.227	3.605
10	0.000	0.011	0.015	0.046	0.109	0.381	0.276	1.042	2.041	3.701

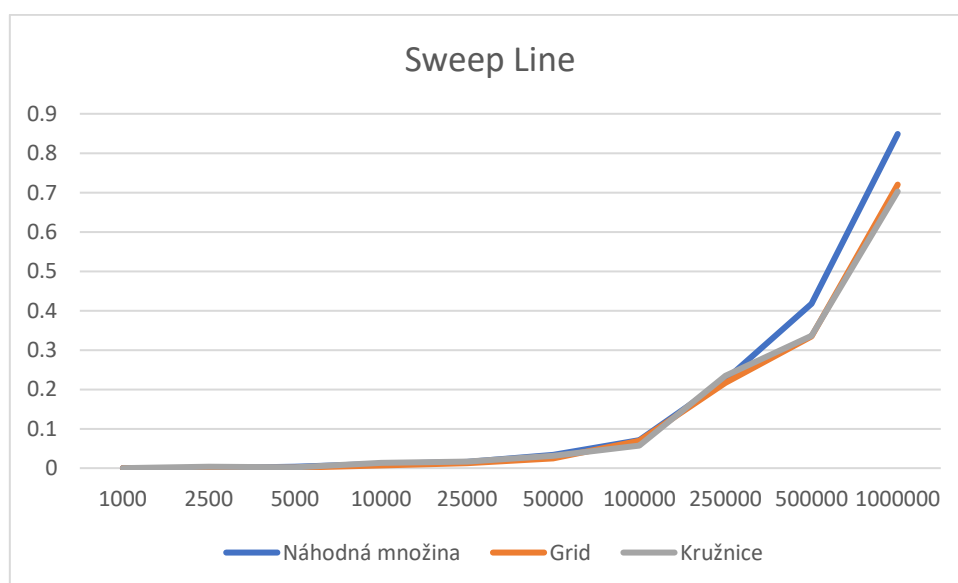
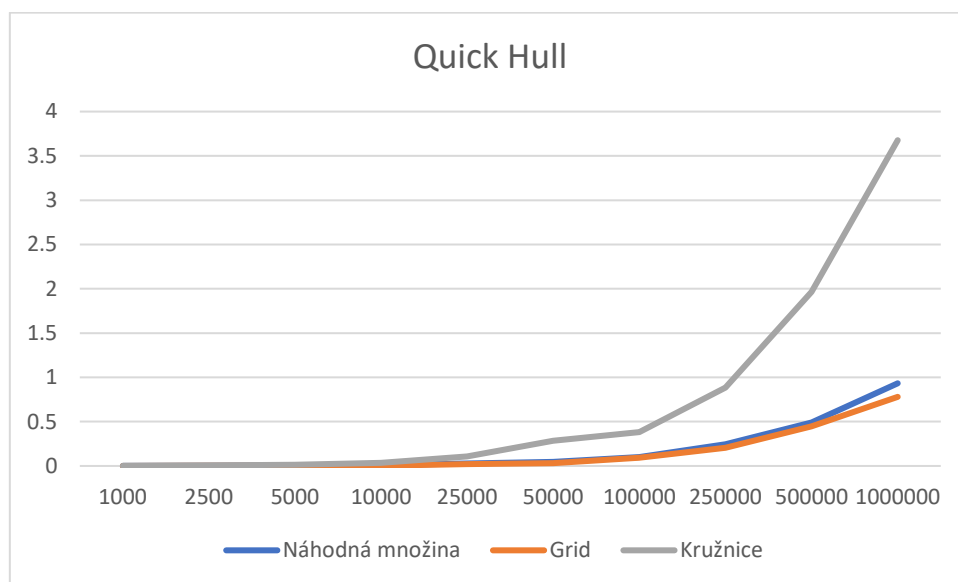
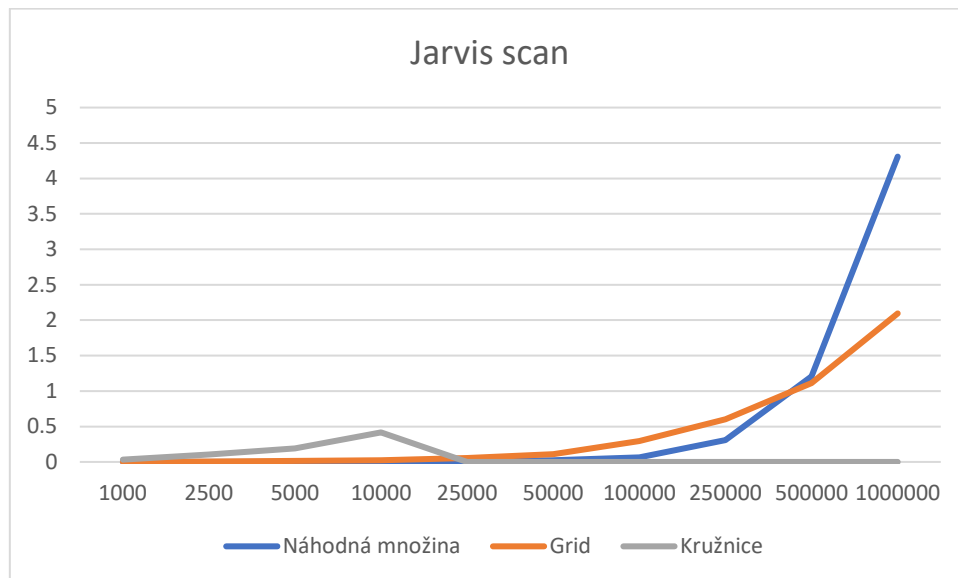
Sweep Line:

Náhodná množina										
	1000	2500	5000	10000	25000	50000	100000	250000	500000	1000000
1	0.001	0.000	0.000	0.017	0.023	0.033	0.079	0.215	0.436	0.899
2	0.001	0.002	0.003	0.017	0.016	0.043	0.084	0.202	0.423	0.928
3	0.001	0.001	0.005	0.000	0.018	0.033	0.084	0.219	0.443	0.880
4	0.000	0.002	0.016	0.016	0.017	0.033	0.076	0.298	0.429	0.887
5	0.001	0.000	0.000	0.017	0.017	0.033	0.073	0.211	0.596	0.892
6	0.001	0.002	0.016	0.019	0.006	0.033	0.066	0.267	0.416	0.909
7	0.001	0.001	0.003	0.000	0.022	0.053	0.083	0.303	0.428	0.983
8	0.001	0.000	0.005	0.017	0.016	0.033	0.068	0.301	0.445	0.990
9	0.001	0.002	0.000	0.016	0.017	0.033	0.084	0.204	0.453	0.913
10	0.001	0.000	0.003	0.000	0.017	0.038	0.068	0.200	0.444	0.886

Grid										
	1000	2500	5000	10000	25000	50000	100000	250000	500000	1000000
1	0.000	0.003	0.000	0.017	0.017	0.021	0.051	0.315	0.351	0.768
2	0.000	0.000	0.009	0.000	0.025	0.038	0.067	0.143	0.337	0.750
3	0.001	0.001	0.000	0.016	0.000	0.017	0.051	0.301	0.321	0.772
4	0.000	0.000	0.000	0.017	0.000	0.033	0.051	0.321	0.323	0.781
5	0.000	0.000	0.009	0.005	0.017	0.036	0.056	0.152	0.326	0.764
6	0.001	0.001	0.003	0.000	0.016	0.033	0.051	0.149	0.325	0.767
7	0.000	0.003	0.009	0.009	0.016	0.024	0.274	0.150	0.321	0.889
8	0.000	0.000	0.000	0.000	0.012	0.033	0.045	0.152	0.500	0.739
9	0.000	0.000	0.009	0.016	0.016	0.016	0.051	0.321	0.321	0.774
10	0.001	0.000	0.002	0.016	0.017	0.017	0.051	0.337	0.490	0.775

Kružnice										
	1000	2500	5000	10000	25000	50000	100000	250000	500000	1000000
1	0.000	0.000	0.002	0.016	0.016	0.034	0.068	0.161	0.358	0.802
2	0.000	0.000	0.000	0.068	0.016	0.034	0.068	0.172	0.359	0.749
3	0.000	0.001	0.002	0.000	0.017	0.033	0.085	0.259	0.363	0.736
4	0.000	0.002	0.000	0.016	0.016	0.034	0.051	0.282	0.362	0.849
5	0.000	0.000	0.002	0.007	0.016	0.035	0.051	0.295	0.375	0.753
6	0.000	0.000	0.016	0.000	0.017	0.035	0.050	0.274	0.375	0.735
7	0.000	0.002	0.000	0.007	0.017	0.035	0.055	0.277	0.362	0.763
8	0.000	0.002	0.000	0.000	0.017	0.034	0.062	0.277	0.380	0.753
9	0.000	0.017	0.002	0.022	0.031	0.034	0.069	0.283	0.352	0.759
10	0.000	0.016	0.000	0.017	0.017	0.035	0.069	0.277	0.377	0.760

Grafy:



Vstupní data

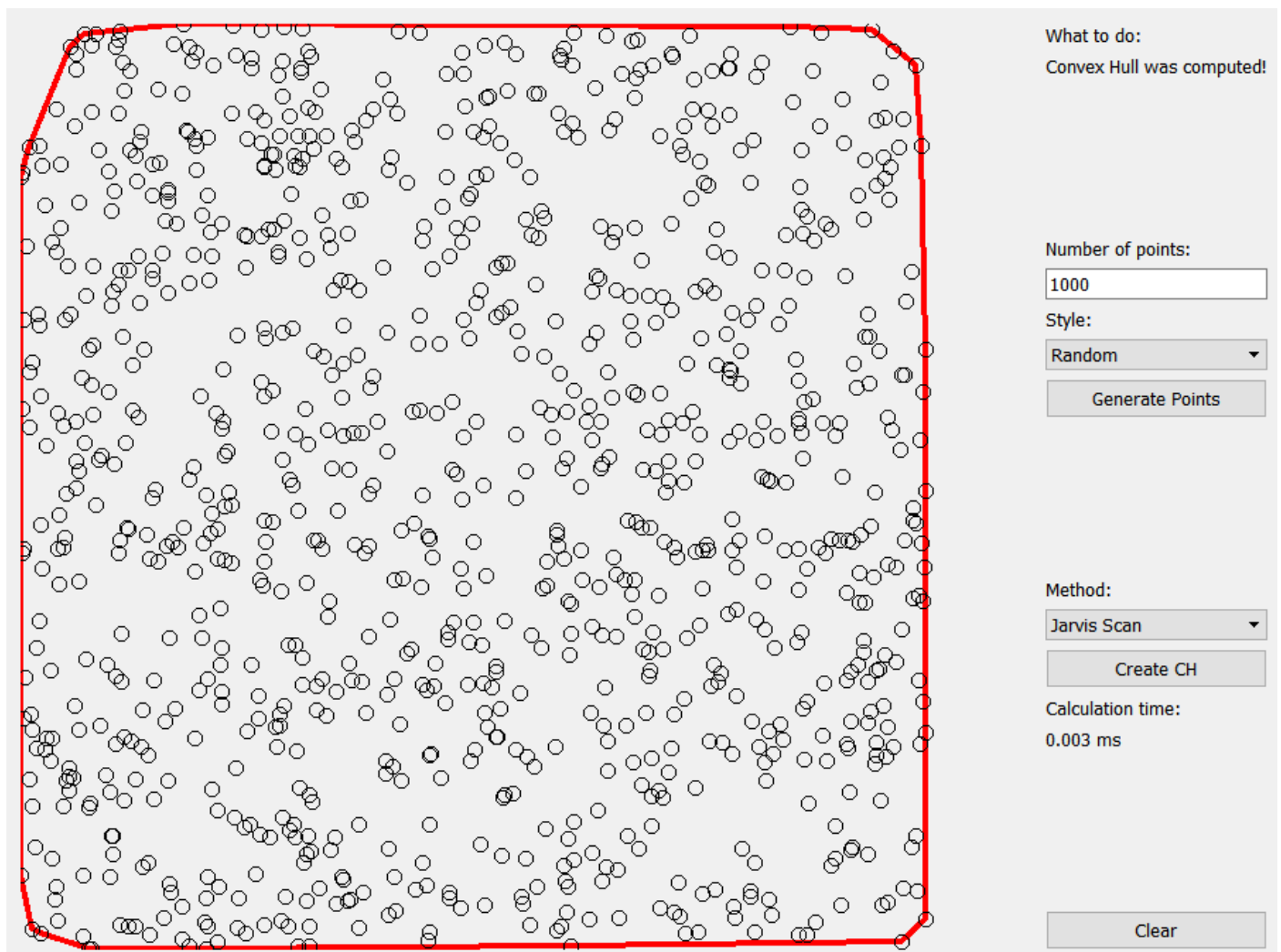
Množiny jsou generovány náhodně, lze volit s možností – náhodně, v mřížce, v kruhu, v elipse, ve čtverci či v obdélníku.

Rovněž lze přidávat body manuálně kliknutím do aplikace.

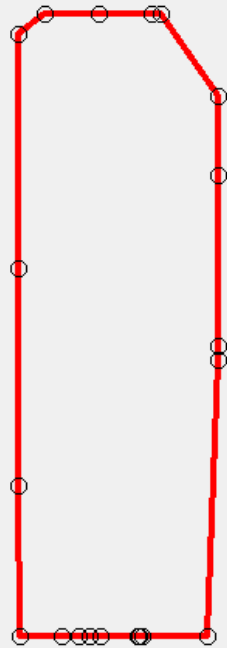
Výstupní data:

Aplikace zobrazuje vygenerované množiny, vypočtenou konvexní obálku a čas výpočtu konvexní obálky.

Náhledy aplikace:



What to do:
Convex Hull was computed!



Number of points:

20

Style:

Rectangle

Generate Points

Method:

QHull

Create CH

Calculation time:

0 ms

Clear

What to do:
Convex Hull was computed!

Number of points:

2000

Style:

Grid

Generate Points

Method:

QHull

Create CH

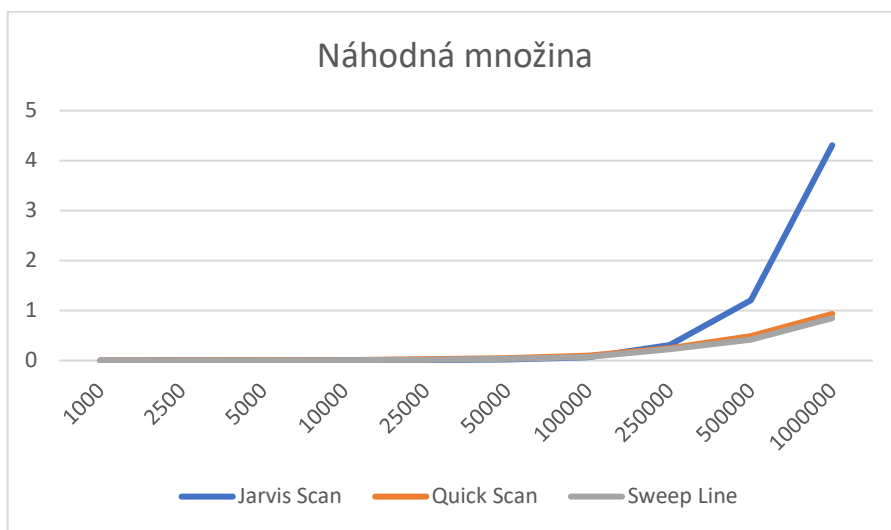
Calculation time:

0 ms

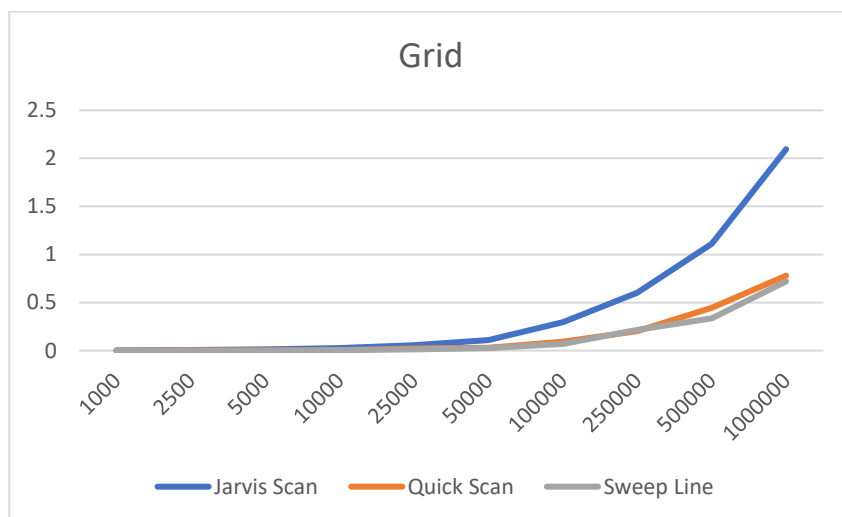
Clear

Porovnání metod:

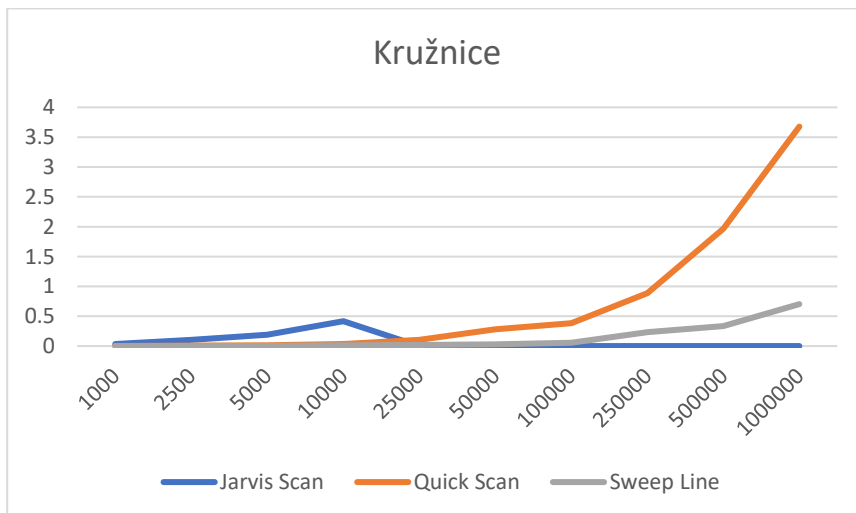
Na náhodné množině pracují Quick Hull a Sweep Line velice podobně. Nejhorší dopadl Jarvis Scan, který je kvůli své kvadratické složitosti při velkém počtu bodů významně pomalejší.



Obdobná situace je i pro body rozložené v pravidelné mřížce, ovšem rozdíl je menší než u předchozí varianty. V tomto případě bylo pracováno s Jarvis Scan, který má ošetřen singulární případ.



Pro body na kružnici vychází nejlépe Sweep Line, následovaná Quick Hull. Nejhorší se jeví Jarvis Scan, pro který se nepodařilo najít výsledek pro více jak 25000 bodů.



Závěr a zhodnocení:

Byl vytvořen program generující náhodné množiny bodů a následně pro ně počítající konvexní obálku pomocí zvolené metody. V rámci aplikace je možno volit rozmístění bodů na kružnici, na elipse, v gridu nebo náhodně.

Byly otestovány rychlosti jednotlivých metod na různě rozložených množinách. Porovnání je znázorněno na předchozích grafech. Pro všechna rozložení vychází značně nejhůře metoda Jarvis Scan, pro kterou se na kružnici ani nepovedlo dokončit výpočet pro více než 25000 bodů.

Metody Quick Hull a Sweep Line si stojí na náhodné množině i gridu podobně a jsou téměř stejně rychlá. Na kružnici je značně rychlejší metoda Sweep Line. Z tohoto důvodu mi vychází jako nejlepší metoda z námi testovaných metoda Sweep Line.

Zdroje:

Obr. 1 – <https://medium.com/@harshitsikchi/convex-hulls-explained-baab662c4e94>

Obr. 2 - <https://i.stack.imgur.com/lzdfs.png>

Obr. 3 - <https://medium.com/@harshitsikchi/convex-hulls-explained-baab662c4e94>

Obr. 4 - <https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk4.pdf>