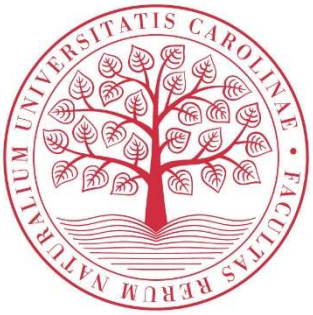


PŘÍRODOVĚDECKÁ
FAKULTA
Univerzita Karlova

Algoritmy v počítačové kartografii

Úloha č. 2
Konvexní obálky a jejich konstrukce

Čelonk Marek, Sýkora Matúš

Úloha č. 2: Konvexní obálky a jejich konstrukce

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

Výstup: $\mathcal{H}(P)$.

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

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

Vstupní množiny bodů včetně vygenerovaných konvexních obálek vhodně vizualizujte. Pro množiny $n \in < 1000, 1000000 >$ 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, Seep 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.

1. **Popis a rozbor problémov + vzorce**.

   V úlohe je riešenie konvexných obálok pre jednotlivé množiny bodov pomocou troch 4 algoritmov:

   - Jarvis Scan
   - Quick Hull
   - Sweep Line
   - Graham Scan

   Konvexne obálky patri k najpoužívanejším geometrickým štruktúram, ako pomocné štruktúry pre mnoho algoritmov. Používajú sa ako prvotný odhad tvaru priestorových dát (javov).

   V prostredí GIS a kartografií sa hlavne využívajú na detekciu tvaru a natočenie budov, analýzu tvarov objektov a pri analýze zhlukov ako vlastnosti jednotlivých clustrovaných dát.

   Dôležitá vlastnosť algoritmov by mala byť rýchlosť spracovania pre rôzne usporiadane datasety, ako je náhodná množina bodov, usporiadaná množina bodov (raster, Grid) a množina bodov na kružnici.

   Daný problém je možne riešiť s vzužitím rôzných algoritmov. Napríklad s algoritmami ktoré sú uvedené výššie ako Jarvis Scan, Quick Hull, Sweep Line, Graham Scan a dalšími napríklad aplikovaním Inkrementálnej konštrukcie alebo Divide and Conquer princípu.

2. **Popisy algoritmov formálnym jazykom.**

   V úlohe sú použité štyri algoritmy Jarvis Scan, Quick Hull, Sweep Line a Graham Scan. Pre dané algoritmy je nutné si vopred vypočítať niektoré hodnoty pomocou dalších funkcií.

2.1. Jarvis Scan

Algoritmus pripomína balenie darčekov do papiera označovaný tiež ako gift wrapping algorithm. Predpoklad pre tento algoritmus je že v množine bodov sa

nenachádzajú tri kolineárne body nieje vhodný pre veľké množiny bodov. Princíp algoritmu je maximalizovanie uhlov, ktorý zvierajú jednotlivé dva vektory. Algoritmus najprv vyhľadá bod s minimálnou hodnotou y, Ktorou sa preloží priamka rovnobežná s osou x, tento bod tvorí konvexnú obálku. Následne sa vyhľadávajú body, ktoré zvierajú s danou stranou najväčší uhol. Ak sa taký bod nájde je pridaný do množiny bodov tvoriacich konvexnú obálku. Tento postup sa opakuje s novo vytvorenou hranou množinou bodov, ktoré nepatria do obálky. Algoritmus využíva smer proti hodinovým ručičkám.

Kroky algoritmu:

- Nájdenie pivota (bod s minimálnou hodnotou y)
- Pridanie pivota do listu
- Inicializácia bodov
- Cyklus pre vyhľadávanie bodov konvexnej obálky
    - Nájdenie bodu, ktorého vektor zviera najväčší uhol s predchádzajúcou hranou
    - Pridanie bodu do listu bodov konvexnej obálky
    - Inicializácia bodov
- Vytvorenie konvexnej obálky

2.2. Quick Hull

Využíva analógiu Quicksortu. Najprv sa množina bodov usporiada podľa hodnôt x. Body s najnižšou a najvyššou hodnotu automaticky tvoria body konvexnej obálky. Tieto dva body tvoria hranu ktorá rozdeľuje množinu na 2 časti lower a upper. Tieto množiny bodov sú spracovávané jednotlivo rekurzívne. Od danej hrany sa v každej množine nájde najvzdialenejší bod, ktorý vytvorí trojuholník. Daný bod patrí do množiny bodov tvoriacich konvexnú obálku. Body vnútri trojuholníka môžeme ignorovať lebo nikdy nebudú tvoriť obálku. Opäť sa vyhľadajú body najvzdialenejšie od novovytvorených hrán a postup sa opakuje pre obe množiny . Algoritmus sa rozdeľuje na dve procedúry.

Kroky algoritmu

- Lokálna procedúra
  - Nájdenie min a max hodnôt x
  - Pridanie daných bodov do listu
  - Rekurzívne volanie globálnej procedúry pre lower a upper hull
  - Vytvorenie konvexnej obálky.
- Globálna procedúra
  - Nájdenie najvzdialenejšieho bodu od úsečky tvorenej bodmi s hodnotou min a mx x.
  - Uloží bod do listov (lower, uupper)
  - Pridanie bodov do upper resp lower hull pomocou cyklu

2.3. Sweep Line

Metóda zametacej priamky je realizáciou inkrementálnej konštrukcie. Nad rovina rozdeľuje množinu na spracovanú a nespracovanú časť. Najprv sa zotriedi množina bodov podľa súradnice x. Nad množinou zotriedených bodov sa pohybuje nad rovina cez všetky body. Iniciálne riešenie je tvorené trojuholníkom. Pre pridávanie bodu do obálky, expanduje v kladnom smere x. Algoritmus používa 3 zoznamy. Prvá hrana je tvorená 2 bodmi s minimálnou hodnotou x. Tretí bod je v pravej alebo ľavej polovine, nutne je podmienka zachovať CCW orientáciu. Následne sa vytvorí aproximácia obálky, ktorá nemusí byť konvexná. Obálka sa rozpojí v bodoch s ktorými tvorí nasledujúci bod dotyčnice s danou obálkou a nahradí sa sekvencia vrcholov. Nekonvexná obálka sa prevedie na konvexnú, čiže sa vynechajú nekonvexné vrcholy podľa dotyčnice či je konvexná alebo nie. Ak je nekonvexná daný bod sa vynechá.

2.4. Graham Scan

Konštrukcia star – shaped polygónu z pivotu q, tvorený usporiadanými vrcholmi podľa uhlov. Následne sa prevedie star – shaped polygón na konvexnú obálku. Využíva kritérium ľavotočivosti . Využíva zásobník pre rýchle pridávanie a odoberanie bodu. Algoritmus najprv vyhľadá bod s najmenšou hodnotou y. Následne zotriedi body podľa uhlov. Inicializuje prvky a pridá prvé dva body

pridá do obálky. Ak ďalší bod leží vľavo daný bod pridá inak ho odoberie z konvexnej obálky.

Kroky algoritmu:

- Nájdenie pivota q s minimálnou hodnotou
- Zotriedenie množiny bodov podľa uhlu
- Inicializácia prvkov
- Pridanie q a p1 do zásobníka
- Cyklus cez všetky body
    - Ak je vľavo pridá bod
    - Ak je vpravo odoberie bod
- Vytvorenie konvexnej obálky

3. Problematické situácie a ich rozbor (tj. simplexy) + ošetrenie týchto situácii v kóde.

V Algoritmoch pre tvorbu konvexných obálok vzniká problém pri kolineárnych bodoch v spracovávanom datasete. Tento problém bol vyriešený v algoritme Jarvis scan. Ak sa uhly dvoch nasledujúcich bodov rovnajú tak sa vypočítajú dĺžky daných vektorov a do úvahy sa zoberie bod ktorý je vzdialenejší.

4. Vstupné dáta, formát vstupných dát, popis.

Pre funkčnosť aplikácie je potrebná množina bodov na základe ktorej sa následne vytvorí konvexná obálka po použití jednotlivých algoritmov. Množiny bodov sa priamo generujú v aplikácii kde si užívateľ môže vybrať z rôznych tvarov množín bodov. Aplikácia podporuje generovanie množín bodov pre tieto tvary:

- Points – množina náhodne vygenerovaných bodov
- Circle – množina bodov vygenerovaná v tvare kruhu
- Ring – množina bodov vygenerovaná v tvare kružnice
- Ellipse – množina bodov vygenerovaná v tvare elipsy
- Square – množina bodov vygenerovaná v tvare štvorca

- Heart – množina bodov vygenerovaná v tvare srdca
- Triangle – množina bodov vygenerovaná v tvare trojuholníka
- Line Star – množina bodov vygenerovaná v tvare hviezdy

5. Výstupní dáta, formát výstupných dát, popis.

Výstupom aplikácie je vytvorená konvexná obálka z vygenerovaného datasetu pomocou vybraného algoritmu, ktorá je zobrazená červenou líniou priamo v aplikácií. Užívateľ taktiež ma možnosť si overiť rýchlosť jednotlivých algoritmov pomocou tlačidla Benchmark. Po jeho stlačení sa zobrazia možnosti kde si užívateľ vyberie veľkosť generovaného datasetu, tvar daného datasetu (Random, Grid, Circle) a algoritmy pre ktoré chce zobraziť čas za ktorý daný algoritmus spracoval zvolenú množinu datasetu. Pre algoritmus Jarvis Scan pre množinu bodov v tvare kruhu je zobrazený čas v mili sekundách, pretože je pomalší o niekoľko rádov a pre ostatné algoritmy je čas zobrazený v mikrosekundách.

Našou úlohou bolo otestovať rýchlosť algoritmov Jarvis Scan. Quick Hull, a Sweep line pre rôzne veľkosti datasetov, kde body boli usporiadané v tvare gridu, kružnice a náhodne. V Tabuľkách č.1 až č.3 sú uvedené hodnoty z meranie pre náhodne usporiadané body, pre vybrané algoritmy.

Tabuľka č.1: Hodnoty časov algoritmu Jarvis Scan pre náhodné generované datasety

	Jarvis [ms]											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Průměr**	**Sm. odchylka**
1000	6	5	6	7	4	6	4	6	4	5	5,3	1,005
5000	38	42	41	35	37	37	34	49	44	42	39,9	4,346
10000	94	101	109	87	83	67	93	89	77	78	87,8	11,660
15000	86	104	135	139	123	158	122	141	159	118	128,5	21,676
50000	653	480	458	373	473	578	558	488	596	463	512	78,210
100000	1214	1298	1302	1234	909	843	1189	1166	1465	1061	1168,1	177,239
150000	2120	1971	2169	1633	1802	1755	1448	1990	1461	1712	1806,1	240,835
500000	7296	8357	6048	6325	6337	5730	7574	6608	5422	6230	6592,7	851,508
750000	12702	11562	10192	7977	10395	11888	11378	9796	9050	10946	10588,6	1336,564
1000000	13304	13877	15361	16213	17088	12774	14133	12099	16108	15042	14599,9	1544,350

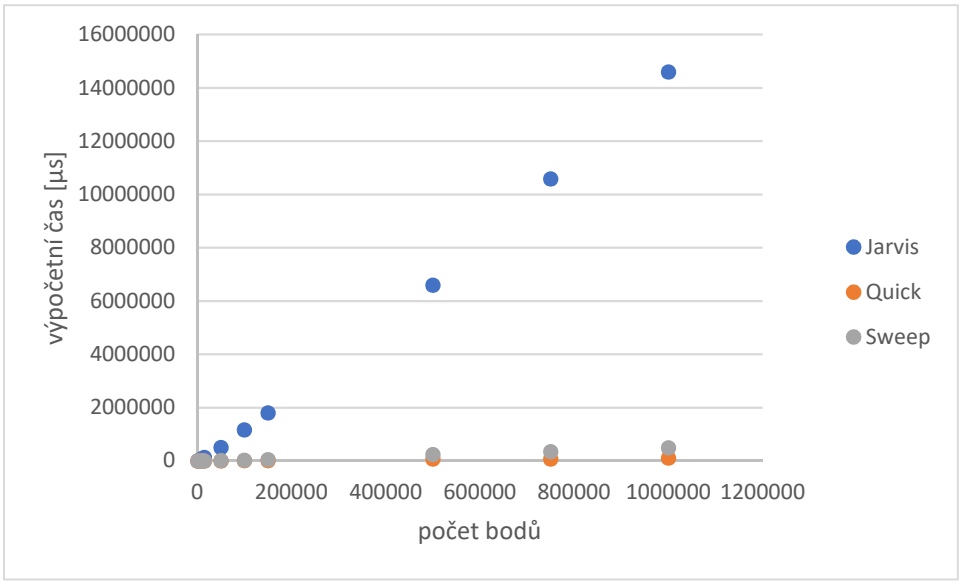
Tabuľka č.2: Hodnoty časov algoritmu Quick Hull pre náhodné generované datasety

	Quick [µs]											
Dataset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Průměr**	**Sm. odchylka**
1000	217	216	191	198	203	217	200	201	191	208	204,2	9,474
5000	1102	807	778	826	820	809	864	585	511	523	762,5	170,118
10000	810	1063	790	1614	726	615	721	741	731	712	852,3	276,865
15000	1118	968	1141	1158	998	1541	1010	1007	1031	1033	1100,5	159,272
50000	3306	3369	3121	3180	8015	3490	5854	2964	3387	3071	3975,7	1563,607
100000	6767	7485	13959	6982	7458	6381	7758	6913	6659	7289	7765,1	2103,297
150000	11646	16628	12316	11238	11101	11379	11165	11770	11410	20120	12877,3	2876,701
500000	65912	66586	296393	42341	46398	68320	43370	37536	42115	43261	75223,2	74558,602
750000	68240	95690	64207	61240	67311	55157	92886	65365	64978	97959	73303,3	14979,907
1000000	87612	107222	145560	121569	82665	136005	80014	140310	82451	88509	107191,7	25080,986

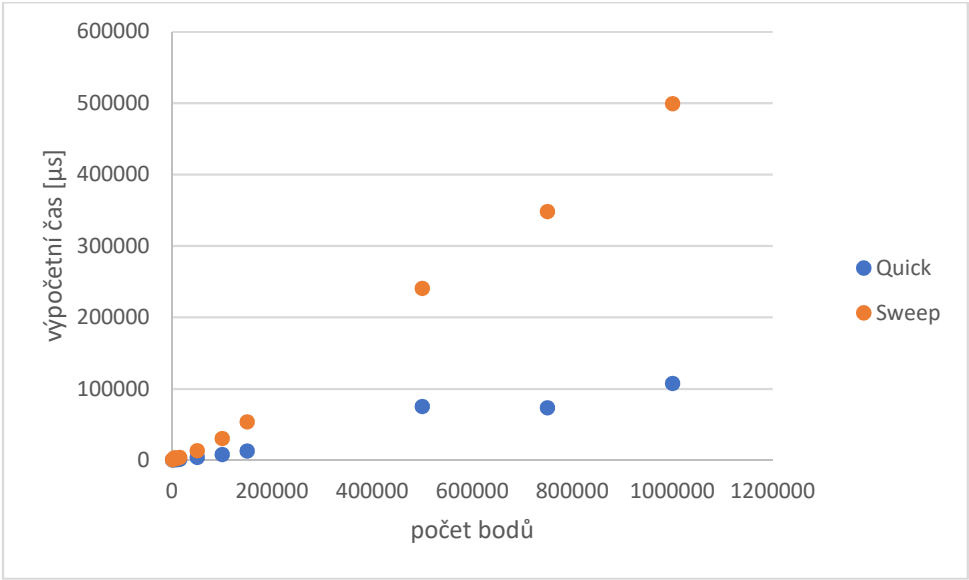
Tabuľka č.3: Hodnoty časov algoritmu Sweep line pre náhodné generované datasety

	Sweep [µs]											
Dataset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Průměr**	**Sm. odchylka**
1000	1142	1040	853	562	491	452	535	456	458	561	655	245,560
5000	3630	3732	5815	4347	3119	3802	3782	3168	2519	1395	3530,9	1093,495
10000	3092	2368	2451	2675	2549	2490	2480	2448	2238	2195	2498,6	238,363
15000	3413	3455	5200	4491	3409	3397	3828	3944	3397	3418	3795,2	578,894
50000	12640	12616	18236	12578	12595	12491	12312	12588	12409	16300	13476,5	1946,941
100000	36011	28768	30102	28838	33145	28534	29557	28565	28846	30012	30237,8	2325,352
150000	49032	48901	62222	58816	60335	52611	49894	53155	52952	48467	53638,5	4811,615
500000	234701	383546	242064	205513	243185	221954	226311	225708	200658	222961	240660,1	49384,575
750000	355773	357558	360992	332035	328622	355763	362630	358886	312130	358980	348336,9	16582,652
1000000	502418	505738	504254	497419	504558	492900	495988	497474	496139	495661	499254,9	4305,510

Pre lepšiu interpretáciu sú výsledné hodnoty zobrazené v Grafe č.1. a č.2.



Graf č.1. Zobrazenie rýchlosti algoritmov pre náhodne generované datasety



Graf č.2. Zobrazenie rýchlosti Quick hull a Sweep line pre náhodne generované datasety

S predchádzajúcich grafov môžeme vidieť že pri náhodne generovanom datasete je najrýchlejši algoritmus Quick Hull. Jarvis scan je pomalši voči Quick hull a Sweep line algoritmov o niekoľko radov.

V Tabuľkách č.4 až č.6 sú uvedené hodnoty z meranie pre body usporiadane v tvare gridu, pre vybrané algoritmy.

Tabuľka č.4: Hodnoty časov algoritmu Jarvis Scan pre body usporiadané v tvare gridu

	Jarvis [ms]											
Dataset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Průměr**	**Sm. odchylka**
1000	2	1	1	2	1	1	2	1	1	2	1,4	0,490
5000	8	9	8	8	8	8	8	8	8	8	8,1	0,300
10000	17	17	17	17	17	17	17	17	17	17	17	0,000
15000	26	25	26	26	25	25	26	26	26	26	25,7	0,458
50000	86	86	87	87	87	87	87	87	87	87	86,8	0,400
100000	174	174	175	177	176	175	178	177	174	176	175,6	1,356
150000	262	265	266	265	265	286	265	265	264	268	267,1	6,457
500000	874	879	879	881	876	872	876	871	871	869	874,8	3,842
750000	1303	1305	1305	1301	1303	1300	1319	1303	1316	1313	1306,8	6,337
1000000	1728	1733	1735	1738	1731	1735	1734	1747	1738	1734	1735,3	4,818

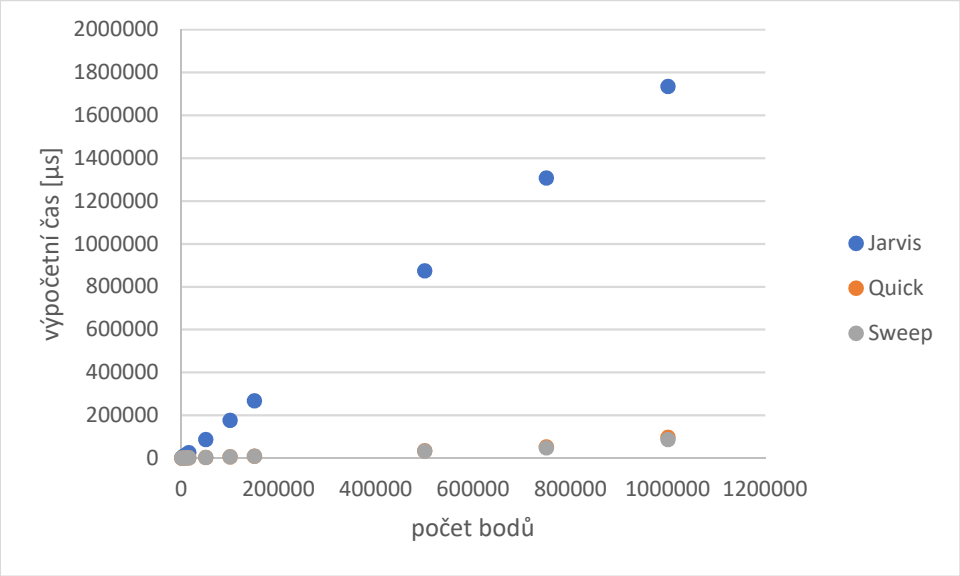
Tabuľka č.5: Hodnoty časov algoritmu Qucik hull pre body usporiadané v tvare gridu

	Quick [µs]											
Dataset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Průměr**	**Sm. odchylka**
1000	59	107	60	57	58	57	57	57	58	59	62,9	14,734
5000	285	320	292	281	298	272	273	309	331	273	293,4	19,754
10000	551	633	572	555	571	574	552	598	553	569	572,8	24,281
15000	847	857	818	837	820	853	817	852	832	859	839,2	15,785
50000	2799	2859	2823	2853	2867	2830	2888	2784	2809	2891	2840,3	35,114
100000	5811	5762	5829	5713	5731	6318	5902	5674	5878	5847	5846,5	172,050
150000	8716	8727	9237	8832	8814	9022	9079	8816	8845	8961	8904,9	158,070
500000	29735	29723	29678	29513	69626	29884	31712	29825	30497	29424	33961,7	11905,086
750000	43672	46037	76659	88486	44264	44028	44318	45560	44136	44751	52191,1	15434,111
1000000	60335	59004	59093	122826	58781	171244	61431	154334	163959	58727	96973,4	47292,144

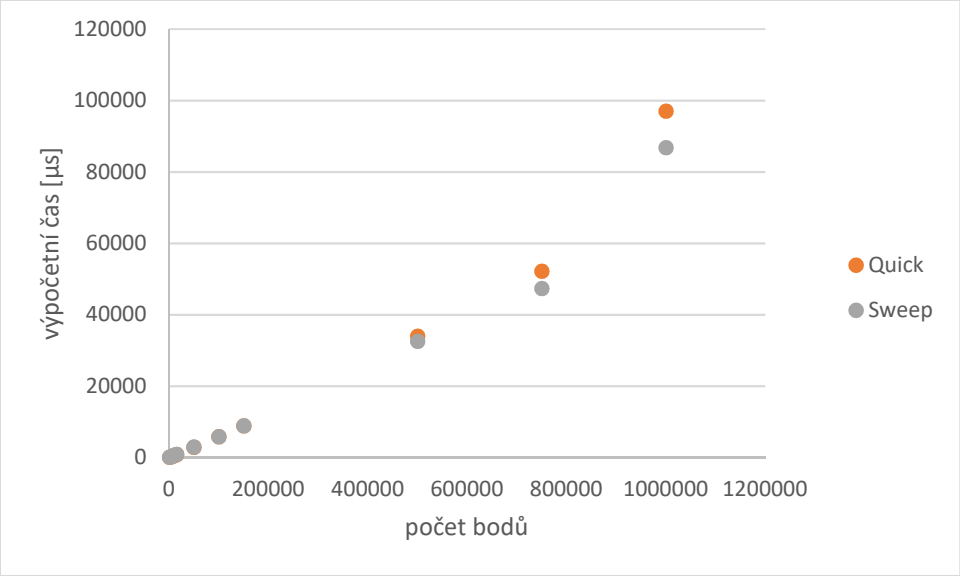
Tabuľka č.6: Hodnoty časov algoritmu Sweep line pre body usporiadané v tvare gridu

	Sweep [µs]											
Dataset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Průměr**	**Sm. odchylka**
1000	72	67	67	100	67	68	67	102	72	70	75,2	13,045
5000	301	305	288	290	297	288	289	291	310	295	295,4	7,338
10000	600	632	601	581	580	603	577	646	583	577	598	22,843
15000	856	892	858	883	855	929	848	854	853	871	869,9	23,935
50000	2894	2881	2904	2882	2950	3138	2957	2832	2897	2871	2920,6	80,225
100000	5758	5862	5840	5770	5944	5817	5783	5851	5849	5799	5827,3	51,935
150000	8842	8813	8832	8861	8862	8768	8900	8735	9023	8983	8861,9	84,067
500000	30249	29703	29818	29487	30560	29701	29546	56637	30313	29648	32566,2	8030,899
750000	44959	44306	44333	44037	43975	45034	74297	44013	44491	44188	47363,3	8984,778
1000000	59015	99410	59608	59798	61897	60423	146629	199465	60631	61112	86798,8	46187,702

Pre lepšiu interpretáciu sú výsledné hodnoty zobrazené v Grafe č.3. a č.4.



Graf č.3. Zobrazenie rýchlosti algoritmov pre body usporiadané v tvare gridu



Graf č.4. Zobrazenie rýchlosti Quick hull a Sweep line pre body usporiadané v tvare gridu

S predchádzajúcich grafov môžeme vidieť že pre body usporiadané v tvare gridu je najrýchlejší algoritmus Sweep line. Algoritmy Sweep line a Quick hull od menšieho počtu bodov pracovali skoro totožnou rýchlosť a však z grafu je zrejme, že pri väčšom počte bodov Sweep ine bol rýchlejši. V Tabuľkách č.7 až č.9 sú uvedené hodnoty z meranie pre body usporiadane v tvare gridu, pre vybrané algoritmy.

Tabuľka č.7: Hodnoty časov algoritmu Jarvis Scan pre body usporiadané v tvare kružnice

	Jarvis [ms]											
Dataset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Průměr**	**Sm. odchylka**
10	0,128	0,121	0,127	0,137	0,12	0,115	0,104	0,114	0,111	0,118	0,1195	0,009
50	2	2	1	2	1	2	2	2	1	2	1,7	0,458
100	6	5	5	4	6	7	5	4	4	6	5,2	0,980
500	123	132	123	146	123	121	135	162	129	119	131,3	12,814
1000	428	431	430	428	432	431	432	432	432	434	431	1,789
1500	1133	1115	1102	1105	1202	1074	1072	1073	1062	1073	1101,1	40,121
5000	10798	10743	10781	10761	10794	10750	10827	10796	10773	10778	10780,1	23,809
7500	26804	26785	26816	27024	26712	27179	26768	27027	27003	26854	26897,2	142,919
10000	43139	43121	43176	43136	43091	43153	43174	43126	43098	43137	43135,1	26,700
15000	97386	97196	97069	97115	97090	97282	97316	97087	97304	97302	97214,7	110,953

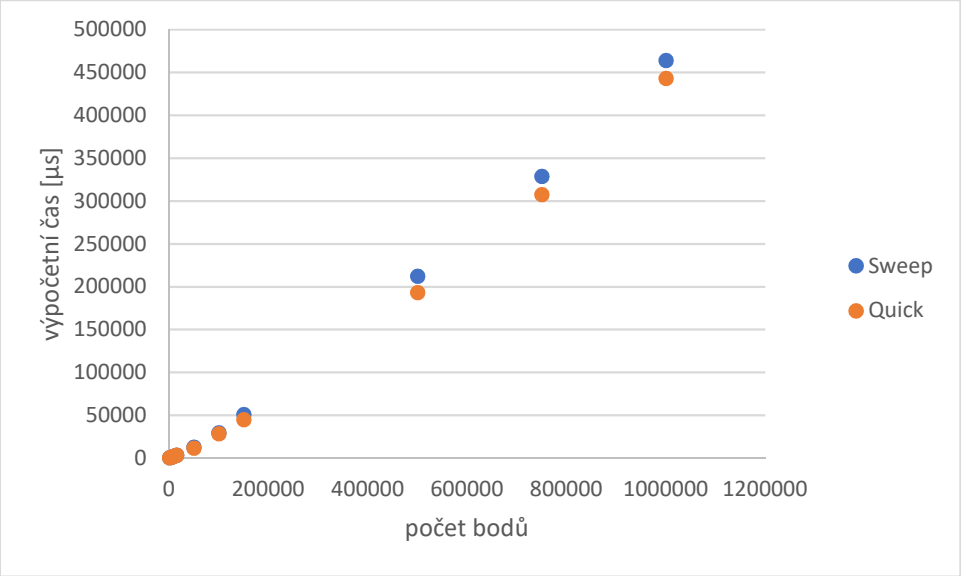
Tabuľka č.8: Hodnoty časov algoritmu Qucik hull pre body usporiadané v tvare kružnice

	Quick [µs]											
Dataset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Průměr**	**Sm. odchylka**
1000	215	244	211	211	254	221	211	234	211	211	222,3	15,186
5000	1069	1062	1061	1057	1074	1048	1062	1059	1046	1066	1060,4	8,188
10000	2134	2152	2357	2156	2160	2165	2272	2278	2254	2240	2216,8	70,023
15000	3414	3308	3281	3327	3288	3220	3357	3309	3246	3304	3305,4	51,638
50000	11357	11409	11280	11332	11458	11348	11422	11393	11420	11419	11383,8	50,687
100000	23941	24262	24337	23951	24238	45370	24369	24271	23958	45866	28456,3	8582,942
150000	41561	41751	60975	41655	40896	59856	42239	39271	39524	41658	44938,6	7796,655
500000	197814	189297	199365	189653	175512	182092	206411	194432	196889	199011	193047,6	8640,349
750000	295113	303834	316345	302787	306909	295125	314912	317315	306547	313584	307247,1	7806,211
1000000	441940	505491	469979	434127	454959	423547	440794	401962	427967	428176	442894,2	27145,711

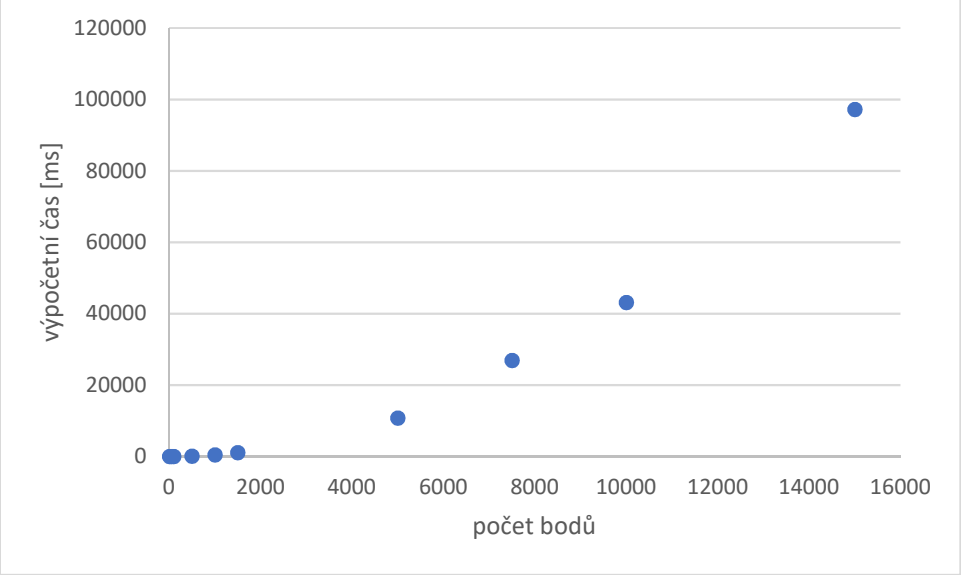
Tabuľka č.9: Hodnoty časov algoritmu Sweep line pre body usporiadané v tvare kružnice

	Sweep [µs]											
Dataset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Průměr**	**Sm. odchylka**
1000	200	184	183	232	228	189	207	186	181	182	197,2	18,236
5000	998	973	1002	994	994	980	1007	1043	1131	1001	1012,3	43,308
10000	2166	2117	2172	2169	2189	2180	2155	2139	2122	2145	2155,4	22,966
15000	3352	3380	3438	3325	3371	3350	3400	3391	3616	3334	3395,7	80,014
50000	12375	12613	12719	12430	12630	12858	12617	12501	12484	12526	12575,3	135,735
100000	30092	28887	29077	28880	29256	28825	30171	31034	29084	29625	29493,1	692,555
150000	49781	52498	50128	54040	51504	49816	48869	50756	50205	50671	50826,8	1429,294
500000	224354	219924	207593	208576	196665	211006	210102	210506	223712	207682	212012	8027,178
750000	293794	328096	335464	346178	328095	335722	332062	330474	332287	325363	328753,5	12876,816
1000000	459351	499173	464868	414588	464740	459168	478919	465259	466342	465539	463794,7	19891,058

Pre lepšiu interpretáciu sú výsledné hodnoty zobrazené v Grafe č.5. a č.6.



Graf č.5. Zobrazenie rýchlosti Quick hull a Sweep line pre body usporiadané v tvare kružnice

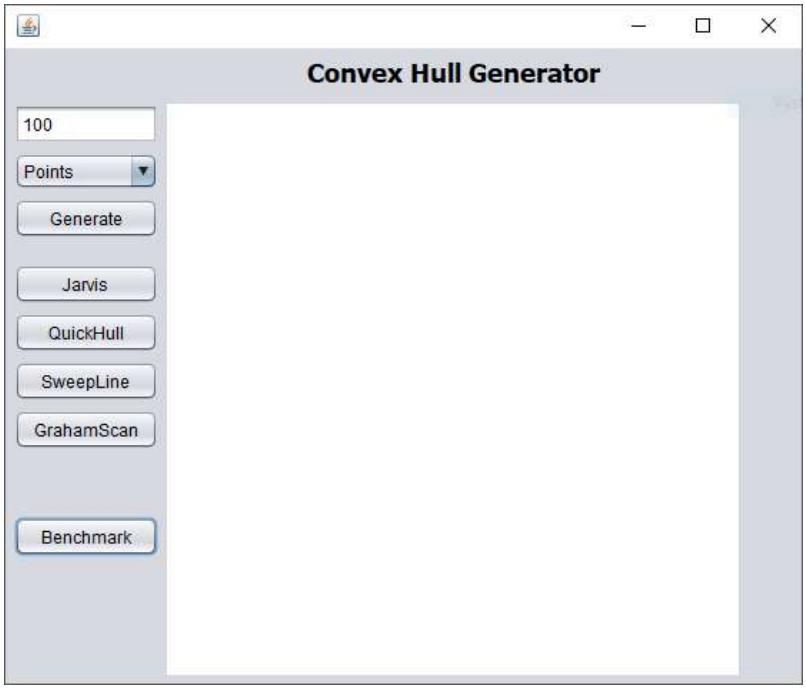


Graf č.6. Zobrazenie rýchlosti Jarvis Scan pre body usporiadané v tvare kružnice

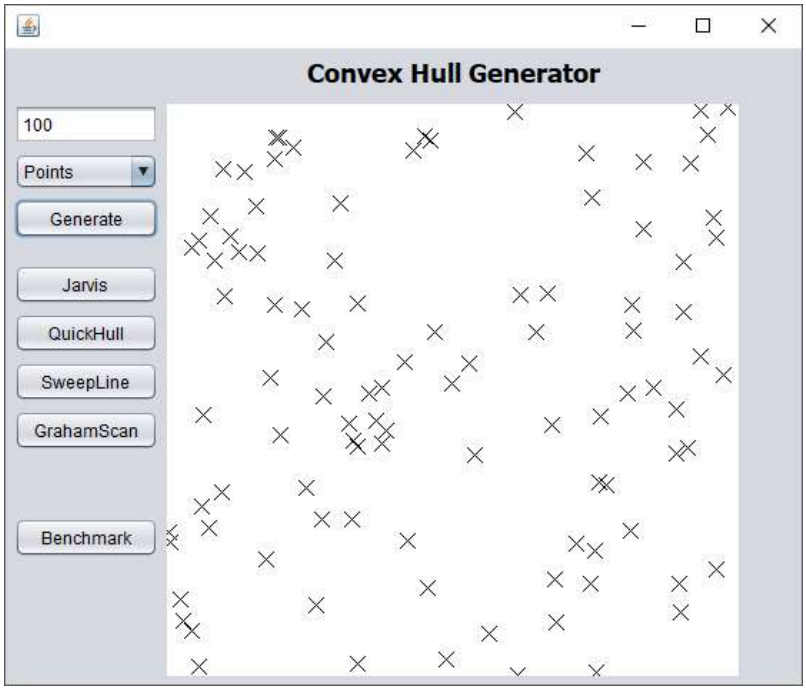
Kvôli pomalému spracovaniu Jarvis Scanu sme upravili pre body generovane na kružnici počet bodov na maximálne 15000. V tomto prípade bol najrýchlejší algoritmus Qucik hull.

6. Printscreen vytvorenej aplikácie.

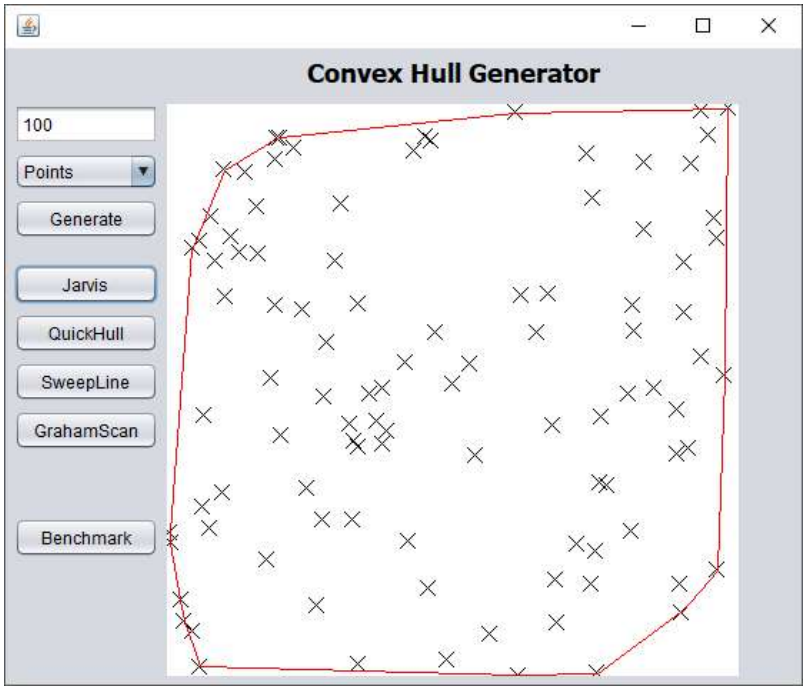
Na obrázku číslo 1 je zobrazená aplikácia po spustení. Obrázok 2 znázorňuje aplikáciu po vygenerovaní bodov. Obrázok 3 zobrazuje aplikáciu s konvexnou obálkou po aplikovaní algoritmu Jarvis Scan. Obrázok 4 zobrazuje okno aplikáciu s možnosťami merania času pre jednotlivé algoritmy po stlačení tlačidla Benchmark.



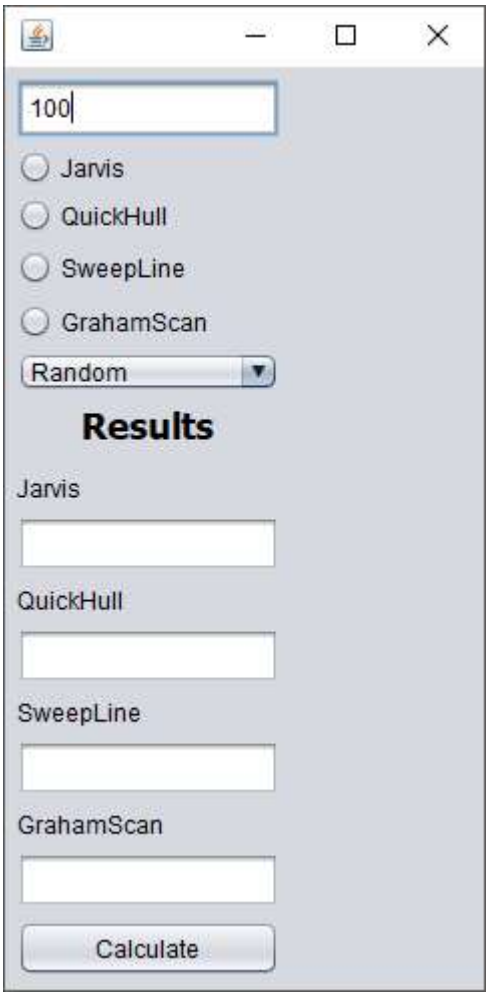
Obr. č.1 aplikácia po spustení



Obr. č.2 Aplikácia po vygenerovaní bodov



Obr. č.3 Aplikácia po vytvorení konvexnej obálky algoritmom Jarvis Scan



Obr. č.4 Okno pre určenie času jednotlivých algoritmov

7. Dokumentácia: popis tried, dátových položiek a jednotlivých metód..

Aplikácia sa skladá celkovo z troch tried. Trieda Algorithms, reprezentuje hlavne výpočty, ktoré sa dejú v aplikácii. Trieda obsahuje dve premenné. Jedna je typu double, ktorá udáva presnosť niektorých výsledkov v daných algoritmoch. Ďalšia je dátového typu enum, ktorá určuje smer jednotlivých algoritmov, medzi ktoré patri clockwise, counter clockwise a colinear. Obsahuje 13 metód z toho obsahuje krátke metódy, ktoré slúžia k doplnkovým výpočtom: OrientationEnum– metóda na určenia orientácie (clockwise, counter clockwise a colinear), dotProd – výpočet skalárneho súčinu,len – výpočet dĺžky skalárneho súčinu, angle – výpočet uhlu medzi vektormi, dotProdNorm – výpočet normovaného uhla, splitPoints – pridá body do listov podľa strany na ktorej ležia, distanceFromLine – výpočet vzdialenosti bodu od línie, fixConvexity – fixuje konvexitu lower a upper obálky.
Medzi hlavné metódy tento triedy patria: jarvisScan, sweepHull, quickHull a qh, ktoré zabezpečujú výpočet a jednotlive vytvorenie konvexných obálok. Metóda qh je globálna procedúra pre Quick Hull algoritmus.

Druha trieda drawPanel, reprezentuje triedu ktorá zabezpečuje vykresľovanie bodov a konvexných obálok. Obsahuje dve premenné, prvá je typu point, ktorá obsahuje pole bodov ktoré sa majú vykresliť a je dátového typu Point2D. Druhá premenná je dátového typu Path2D, v ktorej sa nachádza konvexná obálka. Táto trieda obsahuje iba jednu predefinovanú metódu PaintComponent. Metóda zabezpečuje vykreslenie generovaných bodov, ktoré zobrazí ako kríž a vykreslenie konvexnej obálky červenou farbou.

Tretia trieda GUI reprezentuje grafické rozhranie aplikácie. Premenné v tejto triede zväčša reprezentujú tlačidla, textové správy a textové polia v grafickom rozhraní. Trieda obsahuje viacero metód. Metóda main spúšťa cele grafické rozhranie. Metóda GenerateActionPerformed zabezpečuje generovanie množín bodov rôznych tvarov po vybratí z týchto možností Points, Circle, Ring, Ellipse, Square, Heart, Triangle, Line Star. Metódy qhButtonActionPerformed, jarvisButton1ActionPerformed,sweepButtonActionPerformed, grahamScanActionPerformed sú takmer totožné. V každej z nich sa po stlačení

tlačidla zavolá jednotlivý algoritmus pre vytvorenie konvexnej obálku, ktorú následne po zavolaní drawPanelu vykreslí. Metóda ToggleActionPerformed po stlačení tlačidla Benchmark zobrazí nové okno s možnosti, ktoré si užívateľ nastaví pre zistenie času pre vytvorenie konvexnej obálky jednotlivými algoritmami.Metódy benchmarktDataJarvis, benchmarktDataQuickHull, benchmarktDataSweepLine, benchmarktDataGrahamScan slúžia na výpočet času spracovanie vybraného datasetu podľa zadaných kritérií. Metóda generateCircle sa používa pri výpočte času pre dataset bodov v tvare kruhu, generateGrid pre dataset v tvare mriežky a generateRandom pre dataset náhodne usporiadaných bodov. Poslednou metódou je CalculateActionPerformed, ktorá po stlačení tlačidla Calculate výpočty časov poľa zadaných kritérií od užívateľa a vypíšu sa po spracovaní jednotlivé výsledne časy.

8. Záver, možné či neriešené problémy, námety na vylepšení.

Aplikácia je funkčná pre všetky algoritmy, ktoré boli uvedené v popisoch algoritmov formálnym jazykom a pre vygenerovane datasety rôznych tvarov. Možný problém je v algortime Jarvis Scan, kde pre dataset bodov v tvare kruhu trvá výpočet dlhšie ako u ostatných algoritmov. Aplikáciu je možne rozšíriť o generovanie striktne konvexných obálok a ošetrenie singularity kolineárnych bodov u algoritmoch Quick Hull, Sweep Line, a Graham Scan.

9. Zoznam literatúry.

BAYER, T. 2018. Konvexní obálka množiny bodu. Graham Scan. Jarvis Scan. Quick Hull. Inkrementální metoda. Divide and Conquer. Rotating Calipers. [online]. Praha. Dostupné z:

https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk4.pdf