

Robustní testy mřížového

(17)

Co se představuje, když dala nezávazný a normálního rozdělení?

Necká $\left(\frac{x_1}{y_1}, \dots, \frac{x_n}{y_n} \right)$ je mřížový řízec a mřížového rozdělení,

R_1, \dots, R_m jsou pořadí veličin x_1, \dots, x_n

Q_1, \dots, Q_m — " — y_1, \dots, y_n

Doplete se test. Gearmanova korelační koeficient.

$$r_s = 1 - \frac{6}{m(m^2-1)} \sum_{i=1}^m (R_i - Q_i)^2$$

Teorie testu hypoteze $H_0: \rho = 0$ mříží malející hodnoty r_s , protože pokud by si byla „pořadí“ X-ových a Y-ových veličin hodně podobná, byly by ani mřížové.

H_0 se zamítá, když $|r_s| \geq r_s(\alpha)$, kde $r_s(\alpha)$ je tabulková kritická hodnota.

Příklad: Spolužití alkoholu a úmrtnost, Anděl ob. 20%

Na tomto lednu je mříž sympatické, ne ani pěšné hodnoty sledovány, jde o veličinu mřížového řízce, sloučí ta pořadí.

Např. nám sloučí pořadí nemůže produkto alkoholu a pořadí mřížového řízce na cizoxiu jader a nevolitkové mřížek, kdežto se bude pěšně rýzilo a kolik lidí umřelo.

Příklad: (T. Nádvorníkova, diplomové práce, 1997)

2. Shell sort

Shell sort je klasický algoritmus na setřídění posloupnosti čísel. Byl navržen v roce 1959 Donaldem L. SHELLEM [SH]. Občas bývá nazýván také „diminishing increment sort“. Algoritmus je založen na **přírůstkové sekvenci** přirozených čísel $H = h_1, h_2, \dots, h_{t-2}, h_{t-1}, h_t$. Přírůstková sekvence bývá také nazývána **sekvencí velikostí skoku k** . Vstupní posloupnost čísel a_1, a_2, \dots, a_N o velikosti N prvků je tříděna v t fázích algoritmu pomocí tzv. **h_j -sortu** (neboli **k -sortu**, kde $k = h_j$) pro každý přírůstek h_j ze sekvence přirozených čísel H , počínaje h_t až k h_1 . Proto budeme pro jednoduchost uvádět v dalším textu sekvenci velikostí skoku od zadu jako $H = h_t, h_{t-1}, \dots, h_2, h_1$. Pro Shell sort může být použita libovolná sekvence velikostí skoku splňující dvě podmínky: posloupnost $h_t, h_{t-1}, \dots, h_2, h_1$ je klesající a poslední prvek sekvence h_1 je roven 1.

Každý k -sort vybere z prvků vstupní posloupnosti prvky do právě tříděné podposloupnosti, která je poté setříděna pomocí Insert sortu (třídění vkládáním).

Připomeňme si princip tohoto algoritmu. Insert sort vezme vstupní posloupnost čísel a_1, a_2, \dots, a_N a posunuje doleva každý prvek, počínaje a_2 a konče a_N , který je menší než jeho levý sousední prvek. Poté co jsou dokončeny přesuny prvku a_j , jsou všechny prvky po jeho levé straně setříděny.

INSERT ($a[k], k=1..N$)

```
for i:=2 to N do
begin
  for j:=i-1 downto 0 do
    begin
      vymen_prvek:=a[i];
      if vymen_prvek<(a[j]) then a[j+1]:=a[j]
      else begin a[j+1]:=vymen_prvek; break; end;
    end;
end;
```

Jak již bylo řečeno, nutnou podmínkou Shell sortu je, že poslední prvek sekvence skoků $h_1 = 1$. Protože prvky sekvence velikostí skoku k se postupně od h_t až k h_1 zmenšují, utříďovaná podposloupnost se zvětšuje (obsahuje stále více prvků). To znamená, že Shell sort v každé fázi utřídíuje pomocí Insert sortu stále větší a větší dílčí části vstupní posloupnosti čísel, až v poslední fázi algoritmu setřídí celou posloupnost čísel najednou.

SHELLSORT ($a[j], j=1..N$), pro sekvenci $H=h_t, h_{t-1}, \dots, h_1$

```
for s:=t downto 1 do
begin
  k:=h_s;
  for i:=1 to k do INSERT(a[r], r=i+jk; j=0.. $\left\lfloor \frac{N-i+1}{k} \right\rfloor$ );
end;
```

V jedné fázi třídí algoritmus k podposloupností, kdy první podposloupnost obsahuje prvky $a[1], a[1+k], a[1+2k], \dots$, druhá podposloupnost prvky $a[2], a[2+k], a[2+2k], \dots$ atd. Je-li všech těchto k podposloupností již setříděno, říkáme, že vstupní posloupnost je **k -setříděná posloupnost**.

2.3 Různé verze Shell sortu

Jako sekvence velikostí skoku může být využita libovolná klesající posloupnost přirozených čísel, jejíž poslední prvek je roven 1 (čímž je zajištěno, že v poslední fázi algoritmu se třídí celá posloupnost čísel najednou).

Ve své diplomové práci jsem se zaměřila na porovnání většiny verzí, které jsem nalezla v použité literatuře nebo které jsem navrhla.

Vzhledem k tomu, že Shellova varianta se v literatuře vyskytovala ve dvou podobách (jako stejnomořná i jako nestejnoměrná verze), zahrnula jsem ji do experimentů dvakrát. V prvním případě pod názvem Shell, v druhém pod názvem 1/2 Shell.

Tabulka 1 - Výčet všech verzí

Název verze	Sekvence velikostí skoku	Literatura
Shell	$h_i = 2^i$	[SED]
Papernov-Stasevich	$h_i = 2^i + 1$	[SED] [KN]
Hibbard	$h_i = 2^i - 1$	[KN] [WE1], [WE2] [SED]
Knuth	$h_i = \lfloor (3^i - 1)/2 \rfloor$	[KN] [WE1] [SED]
PR	$h_i = \lfloor (2^i - (-1)^i)/3 \rfloor$	[KN]
Sedgewick 1	$h_i = 4^{i+1} + 3 \cdot 2^i + 1$	[SED]
Sedgewick 2	$h_i = 2 \cdot 4^i - 9 \cdot 2^i + 9$	[SED]
Sedgewick SM	$h_i = 9 \cdot 4^i - 9 \cdot 2^i + 1 \cup 4^i - 3 \cdot 2^i + 1$	[WE1] [WE2]
Pratt	$h_{ij} = 2^i \cdot 3^j$	[KN] [PR]
F	$h_i = h_{i-1} + h_{i-2}$	[KN]
1/2 Shell	$h_i = \lfloor 1/2 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	[KN] [WE1] [WE2]
2/3	$h_i = \lfloor 2/3 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	
3/4	$h_i = \lfloor 2/3 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	[IN] [DOB]
4/5	$h_i = \lfloor 4/5 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	
5/6	$h_i = \lfloor 5/6 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	
6/7	$h_i = \lfloor 6/7 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	
7/8	$h_i = \lfloor 7/8 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	
8/9	$h_i = \lfloor 8/9 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	
9/10	$h_i = \lfloor 9/10 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	
3/5	$h_i = \lfloor 3/5 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	
5/7	$h_i = \lfloor 5/7 h_{i+1} \rfloor, h_t = \lfloor N/2 \rfloor, h_1 = 1$	

Pozn. Prattova verze není v této práci dále vyšetřována.

2.4 Výpočetní složitosti jednotlivých verzí

V tomto odstavci uvádím přehled dokázaných složitostí.

Tabulka 2 - Tabulka výpočetních složitostí dokázaných teoreticky

Název verze	Nejhorší případ	Průměrný případ	Literatura
Shell	$\Theta(N^2)$	$\Theta(N^{3/2})$	[KN str. 90] [WE2 str. 218]
Hibbard	$O(N^{3/2})$ $\Omega(N^{3/2})$		[KN str. 91] [SED str. 162]
Sedgewick 1	$O(N^{4/3})$		[SED str. 165]
Sedgewick 2	$O(N^{4/3})$		[SED str. 168]
Pratt	$O(N \cdot \log^2 N)$	$O(N \cdot \log^2 N)$	[PR str. 32]

Tabulka 3 - Tabulka výpočetních složitostí zjištěných experimentálně

Název verze	Průměrný případ	Literatura
Papernov-Stasevich	$1.09N^{1.27}$ či $0.30N \cdot \ln^2 N - 1.35N \cdot \ln N$	[KN str. 93]
Hibbard	$1.22N^{1.26}$ či $0.29N \cdot \ln^2 N - 1.26N \cdot \ln N$ $\Theta(N^{5/4})$	[KN str. 93] [WE1]
PR	$1.12N^{1.28}$ či $0.36N \cdot \ln^2 N - 1.73N \cdot \ln N$	[KN str. 93]
Knuth	$1.66N^{1.25}$ či $0.33N \cdot \ln^2 N - 1.26N \cdot \ln N$ $\Theta(N^{5/4})$	[KN str. 93] [WE1]
Sedgewick SM	$\Theta(N^{7/6})$	[WE1]

Pozn. Na straně 93 v [KN] jsou uvedeny tyto výsledky s odkazem na experimenty Jamese PETERSONA a Davida L. RUSSELLA na Stanfordské universitě v roce 1971.

3.4 Naměřené výsledky (počet výměn)

Posloupnosti generované náhodným generátorem jsem třídila postupně za použití všech verzí Shell sortu. Při třídění jsem jako první ukazatel výpočetní složitosti měřila počty výměn jednotlivých prvků permutace. Počet výměn se často používá pro simulaci teoretické analýzy výpočetní složitosti. Výměnou je méněna operace algoritmu, kdy dva prvky musí být při třídění prohozeny, protože druhý z nich je menší než první. Počet výměn prvků je ukazatel výpočetní složitosti, který se nemění v závislosti na konfiguraci počítače, operačním systému a překladači, je tedy vhodný pro porovnání více různých studií. Sám o sobě ovšem nemusí přesně vypovídat o kvalitě algoritmu. Protože výměna prvků není jedinou operací, kterou algoritmus provádí (další je například porovnávání), můžeme říci, že váha tohoto ukazatele vzhledem k ostatním výpočetním ukazatelům (zejména času) závisí na tom, jak náročnou operací je výměna prvků pro konkrétní procesor. Může existovat prostředí, ve kterém bude výměna prvků podstatně náročnější než např. porovnání prvků, nebo opačně prostředí, kde náročnost obou operací nebude tolik rozdílná.

Tabulka 4 - Průměrný počet výměn prvků při třídění permutací

V tabulce jsou uvedeny průměrné počty výměn prvků a příslušné směrodatné odchylky pro jednotlivé verze a jednotlivé délky permutací.

\bar{X} - průměrný počet výměn

S - odhad směrodatné odchylky

délka	100		1 000		10 000		100 000		1 000 000	
verze	\bar{X}	S	\bar{X}	S	\bar{X}	S	\bar{X}	S	\bar{X}	S
shell	495	74	15 936	1 846	540 005	103 110	16 281 747	2 641 339	520 419 952	58 692 148
paper	326	30	6 813	332	127 204	4 837	2 448 493	123 677	45 492 252	2 072 956
hibbard	349	25	7 366	427	133 964	6 262	2 507 887	160 322	45 866 667	1 928 913
knuth	426	29	8 802	464	163 882	7 304	2 966 424	140 321	52 879 129	1 933 045
pr	332	28	7 238	347	144 804	5 486	2 785 420	188 092	52 311 184	2 574 749
sedg 1	569	56	11 262	481	169 501	3 020	2 350 127	20 628	31 076 064	208 154
sedg 2	752	103	21 101	1 454	344 538	12 554	4 430 086	55 292	53 057 575	278 060
sedg sm	465	39	7 675	145	109 564	1 122	1 408 508	4 881	17 429 830	43 853
f	317	28	7 380	485	148 627	9 218	2 937 997	140 230	55 798 998	3 330 348
1/2 shell	429	32	7 666	399	154 391	8 468	2 871 191	111 208	50 144 356	2 711 868
2/3	306	21	5 042	92	76 743	414	959 169	1 838	11 678 236	7 062
3/4	274	14	4 575	82	65 515	277	850 320	815	10 450 935	4 726
4/5	274	17	4 508	80	63 405	329	815 168	846	10 038 363	3 527
5/6	260	12	4 351	47	61 406	226	789 084	1 013	9 677 280	3 563
6/7	261	15	4 337	51	60 836	211	783 127	1 282	9 609 213	3 258
7/8	266	14	4 319	71	60 837	269	776 443	770	9 513 879	2 296
8/9	269	17	4 368	60	61 044	250	779 642	887	9 548 536	3 371
9/10	280	15	4 498	64	62 291	280	796 036	913	9 708 516	3 130
3/5	353	26	5 555	123	82 432	640	1 052 434	2 025	13 940 646	14 612
5/7	284	17	4 871	100	68 852	296	891 235	1 344	11 185 468	5 732

3.5 Naměřené výsledky (počet porovnání)

Jako druhý ukazatel výpočetní složitosti jsem při třídění měřila počty porovnání jednotlivých prvků permutace. Porovnáním je míněna operace algoritmu, kdy se pro dva prvky táží, zda první z nich je menší než druhý. V případě, že tomu tak není, následuje výměna prvků. Počet porovnání prvků je stejně jako počet výměn ukazatelem, který se nemění v závislosti na konfiguraci počítače. Také pro počet porovnání platí, že jeho váha vzhledem k ostatním ukazatelům (zejména času) závisí na tom, jak náročnou operací je výměna prvků pro konkrétní procesor. Pro některé procesory může být operace výměny prvku stejně náročná jako operace porovnání, pro jiné tu může být velký rozdíl.

Tabulka 5 - Průměrný počet porovnání prvků při třídění permutací

V tabulce jsou uvedeny průměrné počty porovnání prvků a příslušné směrodatné odchylky pro jednotlivé verze a jednotlivé délky permutací.

\bar{X} - průměrný počet porovnání

S - odhad směrodatné odchylky

délka	100		1 000		10 000		100 000		1 000 000	
verze	\bar{X}	S	\bar{X}	S	\bar{X}	S	\bar{X}	S	\bar{X}	S
shell	1 025	74	24 419	1 843	659 123	103 113	17 806 125	2 641 310	538 884 886	58 692 178
paper	841	29	15 210	332	245 601	4 853	3 964 865	123 639	63 886 815	2 072 757
hibbard	780	24	14 789	430	242 404	6 247	3 924 297	160 290	63 261 182	1 929 002
knuth	725	28	13 818	467	234 774	7 319	3 892 227	140 370	64 274 072	1 932 782
pr	815	29	15 356	341	258 495	5 499	4 258 573	188 100	70 402 355	2 574 947
sedg 1	824	56	15 483	475	229 701	3 021	3 122 765	20 617	40 349 774	208 206
sedg 2	953	102	24 273	1 455	391 347	12 555	5 049 610	55 271	60 931 044	278 075
sedg sm	756	39	13 341	147	197 792	1 111	2 623 087	4 866	32 785 202	44 000
f	1 030	27	19 241	496	315 395	9 215	5 085 815	140 256	82 054 987	3 330 383
1/2 shell	842	30	14 756	395	265 348	8 472	4 281 158	111 239	67 246 151	2 711 957
2/3	964	20	17 514	93	251 364	424	3 305 292	1 799	41 139 265	7 047
3/4	1 183	12	21 506	86	314 620	257	4 141 087	886	51 357 959	4 600
4/5	1 338	15	24 914	71	367 060	323	4 951 188	793	61 397 006	3 570
5/6	1 486	11	28 244	47	429 729	215	5 671 536	901	71 499 463	3 527
6/7	1 558	13	31 715	49	483 897	199	6 512 405	1 124	81 899 880	3 375
7/8	1 718	13	35 213	78	538 616	284	7 252 799	727	92 274 166	2 084
8/9	1 882	13	37 788	59	593 602	224	8 003 220	822	101 780 369	3 208
9/10	1 955	12	41 419	60	649 605	236	8 866 939	825	112 413 367	2 800
3/5	840	25	15 327	127	220 089	619	2 928 771	2 038	36 703 696	14 475
5/7	1 023	15	19 081	96	280 691	293	3 709 406	1 342	45 366 008	5 606

Na následujících stránkách jsou průměrné hodnoty vyneseny do grafů pro názornější představu naměřených hodnot. Výsledky všech měření jsou uvedeny v příloze 3.

3.6 Naměřené výsledky (čas)

V poslední řadě jsem při třídění měřila skutečně spotřebovaný čas běhu programu. Čas byl měřen v sekundách na dvě desetinná místa (x:xx sec). Měřila jsem pouze čas vlastního Shell sortu, včetně času výpočtu hodnoty funkce pro určení skoku k . Do tohoto času se nezapočítával čas pro vstupní a výstupní operace, tj. zejména pro načtení posloupnosti čísel do paměti. Při využití profileru gprof pro měření času rovněž nedochází ke zkreslení měření z důvodu zaneprázdněnosti procesoru jinou úlohou souběžně s naší, protože gprof přesně zaznamenává čas strávený v určené funkci programu. Pro další zpřesnění experimentů by bylo možné provést více měření času pro jednu stejnou úlohu a dále využívat například jejich aritmetický průměr. Vzhledem k tomu, že vícenásobné měření času pro stejnou úlohu by způsobilo zvýšení celkové doby trvání experimentální části práce, rozhodla jsem se každé měření provádět pouze jednou. Toto rozhodnutí je rovněž podloženo spolehlivostí informací z profileru gprof. Čas je vždy zajímavý praktický ukazatel výpočetní složitosti, ale vzhledem k tomu, že záleží na použité konfiguraci počítače, operačním systému a překladači, je pro srovnání použitelný pouze tehdy, když je pro všechna měření dodržena stejná situace.

Tabulka 6 - Průměrný čas běhu programu při třídění permutací

V tabulce jsou uvedeny průměrné časy běhu programu a příslušné směrodatné odchylinky pro jeho jednotlivé verze a jednotlivé délky permutací. Pro permutace délky 100 není čas uváděn, protože byl vždy změřen jako 0:00 sec.

\bar{X} - průměrný čas [sec]
 S - odhad směrodatné odchylinky [sec]

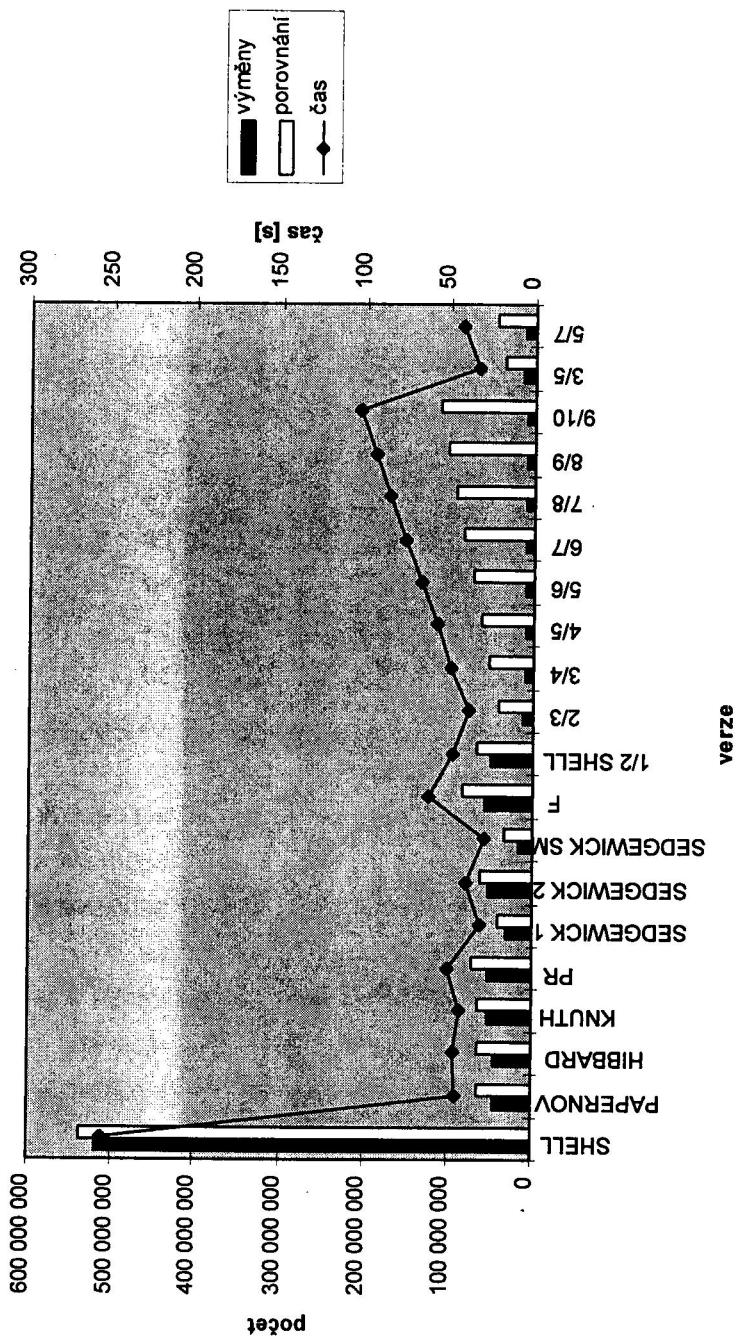
	1 000		10 000		100 000		1 000 000	
	\bar{X}	S	\bar{X}	S	\bar{X}	S	\bar{X}	S
shell	0.0100	0.0029	0.28	0.05	8.30	1.23	256.63	22.68
paperov	0.0060	0.0050	0.13	0.02	2.78	0.27	45.65	1.46
hibbard	0.0044	0.0051	0.12	0.02	2.42	0.40	45.95	1.64
knuth	0.0064	0.0049	0.11	0.02	2.50	0.27	42.47	1.83
pr	0.0068	0.0048	0.13	0.02	2.80	0.41	49.58	1.71
sedg 1	0.0080	0.0041	0.12	0.02	2.03	0.23	31.19	0.90
sedg 2	0.0104	0.0020	0.18	0.03	2.83	0.33	39.64	1.28
sedg sm	0.0060	0.0050	0.11	0.02	1.96	0.22	28.12	0.60
f	0.0080	0.0050	0.17	0.04	3.41	0.46	61.54	2.27
1/2 shell	0.0060	0.0050	0.13	0.02	2.84	0.36	48.00	1.29
2/3	0.0060	0.0050	0.14	0.03	2.61	0.34	38.19	1.05
3/4	0.0084	0.0037	0.17	0.03	3.27	0.46	48.32	0.58
4/5	0.0100	0.0000	0.19	0.02	3.99	0.54	57.23	0.59
5/6	0.0108	0.0028	0.24	0.04	4.59	0.58	66.84	0.82
6/7	0.0140	0.0050	0.26	0.05	5.21	0.67	76.09	0.80
7/8	0.0128	0.0046	0.29	0.06	5.91	0.68	85.08	0.86
8/9	0.0140	0.0050	0.33	0.06	6.39	0.82	93.21	1.11
9/10	0.0144	0.0051	0.33	0.04	6.87	1.01	102.82	1.94
3/5	0.0068	0.0048	0.12	0.03	2.31	0.27	33.33	0.98
5/7	0.0080	0.0041	0.15	0.02	2.94	0.41	42.59	0.50

Na následujících stránkách jsou průměrné hodnoty vyneseny do grafů pro názornější představu naměřených hodnot. Výsledky všech měření jsou uvedeny v příloze 4.

Graf 19 - Porovnání měření pro různé verze - permutace délky 1 000 000

Graf zobrazuje průměr naměřených hodnot. V grafu vidíme poměr počtu výměn prvků vůči počtu porovnání prvků a času. U verze Sedgewick 2 bylo vysoké procento prvků po porovnání vyměněno, naopak verze 8/9 a 9/10 prováděly zbytečně moc nevyužitých porovnání (tj. porovnání, po kterých nenašlo žádoucího výměnu). Naměřený čas koresponduje s počtem porovnání.

Naměřené výsledky pro permutace délky 1 000 000



Spiracorda Thellorhynchus (*dialex postlongirostris* $m = 1000\ 000$)

Shell	<u>gymnery</u>	<u>porosumini</u>	<u>eas</u>
	20	20	20
Papernov	13	10	8
Hibbard	14	9	9
Knuth	14	11	6
PR	16	13	12
Sedg 1	12	3	2
Sedg 2	18	2	5
Sedg sm	11	1	1
F	19	16	14
1/2 Shell	15	12	10
2/3	9	4	4
3/4	7	6	11
4/5	6	8	13
5/6	5	14	15
6/7	3	15	16
7/8	1	17	17
8/9	2	18	18
9/10	5	19	19
3/5	10	2	3
5/7	8	5	7

Test nerávnoslosti pravosti a výmen:

$$\text{Spurmanov korelační koeficient } r_s = 1 - \frac{6}{m(m^2-1)} \sum_{i=1}^m (R_i - Q_i)^2 =$$

$$= 1 - \frac{6}{20(20^2-1)} (0 + 3^2 + 5^2 + 6^2 + 3^2 + 9^2 + 11^2 + 10^2 + 3^2 +$$

$$+ 3^2 + 5^2 + 1^2 + 2^2 + 10^2 + 12^2 + 16^2 + 16^2 +$$

$$+ 15^2 + 8^2 + 3^2) =$$

$$= 1 - \frac{6}{20 \cdot 399} \cdot 1454 = 1 - 1,0932 = -\cancel{0,992} - 0,0932$$

kritická hodnota $r_{2,0}(0,05) = 0,4451$

$|r_s| < 0,4451 \Rightarrow$ hypotéza o nerávnoslosti se nezměnila

Test nerávnoslosti času a výmen:

$$r_s = 1 - \frac{6}{20 \cdot 399} (0 + 5^2 + 5^2 + 11^2 + 4^2 + 10^2 + 13^2 + 10^2 +$$

$$+ 5^2 + 5^2 + 5^2 + 4^2 + 4^2 + 11^2 + 13^2 + 16^2 +$$

$$+ 16^2 + 14^2 + 7^2 + 1) =$$

$$= 1 - \frac{6}{20 \cdot 399} \cdot 1744 = 1 - 1,3113 = -0,3113$$

$|r_s| < 0,4451 \Rightarrow$ hypotéza o nerávnoslosti se nezměnila

Test nerávnosti času a porovnání:

$$\begin{aligned}
 r_s &= 1 - \frac{6}{20 \cdot 399} (0 + 2^2 + 0 + 5^2 + 1 + 1 + 2^2 + 0 + 2^2 + 2^2 + \\
 &\quad + 0 + 5^2 + 5^2 + 1 + 1 + 0 + 0 + 0 + 1 + 2^2) = \\
 &= 1 - \frac{6}{20 \cdot 399} \cdot 100 = 1 - 0,0452 = 0,9548
 \end{aligned}$$

$r_s > 0,4451 \Rightarrow$ hypotéza o nerávnosti času na počtu porovnání se zamítlá

Poznámka: Výsledky jsou pravděpodobně silně ovlivněny druhou skupinou algoritmu (verze 2/3 až 5/7), kde jsou velice rozdílny
 ✓ porovnaním počtu porovnání a výměn.
 ✓ algoritmu skupiny Shell - $\frac{1}{2}$ Shell by se patrně proto zlepšila závislost času jen na počtu porovnání, tak na počtu výměn.