

从算法分析的角度理解哥德巴赫猜想

本文大概分为以下几部分

- 1，以自然数数列为起点，计算素数数列
- 2，逐渐逼近素数数列
- 3，筛法执行 n 次和 n+1 次，素数间距离的变化
- 4，已知纯粹素数数列的循环节结构
- 5，对向列车行进模型
- 6，从素数序列得到完整偶数序列

首先声明，本文不参考在我之前的关于哥德巴赫猜想的论述，因为他们的公式我看不懂。  
本文的思路源于 2008 年冬天，读过徐迟写的哥德巴赫猜想的节选。偶然想到关于研究素数间距离的一个思路，于是开始动手计算 10000 以内的素数。

## 以自然数数列为起点，计算素数数列

第一步，10 进制数，按 10 个数为一段进行分段。每一段内尾数 2,4,6,8,0,5 的数删除。剩余数字的尾数是 1,3,7,9。  
第二步，计算尾数为 1,3,7,9 的数列，两两相乘。写成图表就是

	1	3	7	9	11	13	17	19	21	23	27	29	31	33	37	39
1	1	3	7	9	11	13	17	19	21	23	27	29	31	33	37	39
3	3	9	21	27	33	39	51	57	63	69	81	87	93	99	111	117
7	7	21	49	63	77	91	119	133	147	161	189	203	217	231	259	273
9	9	27	63	81	99	117	153	171	189	207	243	261	279	297	333	351
11	11	33	77	99	121	143	187	209	231	253	297	319	341	363	407	429
13	13	39	91	117	143	169	221	247	273	299	351	377	403	429	481	507
17	17	51	119	153	187	221	289	323	357	391	459	493	527	561	629	663
19	19	57	133	171	209	247	323	361	399	437	513	551	589	627	703	741
21	21	63	147	189	231	273	357	399	441	483	567	609	651	693	777	819
23	23	69	161	207	253	299	391	437	483	529	621	667	713	759	851	897
27	27	81	189	243	297	351	459	513	567	621	729	783	837	891	999	1053
29	29	87	203	261	319	377	493	551	609	667	783	841	899	957	1073	1131
31	31	93	217	279	341	403	527	589	651	713	837	899	961	1023	1147	1209
33	33	99	231	297	363	429	561	627	693	759	891	957	1023	1089	1221	1287
37	37	111	259	333	407	481	629	703	777	851	999	1073	1147	1221	1369	1443
39	39	117	273	351	429	507	663	741	819	897	1053	1131	1209	1287	1443	1521

图 1

第三步，从自然数数列中删除合数，得到素数数列。图表是

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270
271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330
331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390
391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420
421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480
481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510
511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540
541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570
571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630
631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660
661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690
691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720
721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750
751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780
781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810
811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840
841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870
871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930
931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960
961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990
991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020
1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050
1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080
1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110
1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140
1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170
1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200
1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230
1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260
1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290
1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320
1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350
1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380
1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410
1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440
1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	

第 3 步



图 5

第 1 步到第 3 步，在最小范围内呈现了素数的分布规律。  
这一分布规律用文字描述就是：  
自然数按等长分段，自第二段起，每段最右侧位置是合数。

也可以理解为这是素数定义的另外一种表现形式。素数定义为只能被 1 和本身整除的整数。第二段最右侧是分段长度的 2 倍，第 3 段是分段长度的 3 倍。

第 4 步

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

图 6

第 4 步，进一步呈现了素数分布规律。  
用文字描述就是，在以素数阶乘为分段的段落内重复出现。

素数的阶乘， $2 \times 3 = 6$ 。  
以 6 为分段，素数的位置 7,13,19，是重复出现的。5,11,17,23,29，是重复出现的。

已知小于 30 的全部素数，30 到 210 的素数可以写成

$P = P + 30n$

图表是

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210

图 7

第 5 步  
已知  $p < 30$ ,  
 $P = P + 30n$ ,  
 $P = P + 30n < 210$   
 $P < 210$  删除  $P1 \times P2$   
得到 210 以内全部素数

第 6 步，  
 $P=P+210n$   
 $P=P+210n<2310$

至此可知，计算素数数列是一个递归执行的不断逼近完整素数数列的算法。

## 执行第 n 步和第 n+1 步，素数间距离的变化

继续以 30 以内素数为例，首先来看相邻两个素数的距离，也就是两个素数的差。

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		1		2		2				4		2				4		2				4							6

图 8  
执行第 1 步，删除 2 以上全部偶数。剩余数的间距全都是 2。  
执行第 2 步，删除  $3*3=9,3*5=15,3*7=21,3*9=27$ ，  
执行第 3 步，删除  $5*5=25$ ，间距就变成了

		1		2		2				4		2				4		2				4							6
--	--	---	--	---	--	---	--	--	--	---	--	---	--	--	--	---	--	---	--	--	--	---	--	--	--	--	--	--	---

图 9  
图 9 最右数字 6，  
 $6=2+4$   
删除了  $3*9=27$ ，又删除了  $5*5=25$ ，合并间距  $2+4=6$

执行第 n 步和第 n+1 步，相邻两个素数的距离会因为删除一个合数而叠加。这是在逼近素数数列时必然发生的。

写成公式就是  
 $(P_{n+1}-P_n)=(P_{m+1}-P_m)+(P_m-P_{m-1}),n>m$

## 已知纯粹素数数列的循环节结构

$$P_{(30\sim210)}=P_{(1\sim30)}+30n$$
$$P_{(210\sim2310)}=P_{(1\sim210)}+210n$$
$$P_{(2310\sim30030)}=P_{(1\sim2310)}+2310n$$

每执行一次递归，就有合数被删除。素数 P 的位置重复出现，循环节逐步扩大。

相邻两个素数间距 $(P_{n+1}-P_n)$ 伴随素数数列的循环节结构依次分布。

6<P<30.

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

图 6

30<P<210,

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210

图 7

本文前半部分到此结束。算法分析得到的结论基本够用了。下一步将要利用已经得到的结论，构建新的算法，并且论证新算法的可行性。  
新算法经过论证，可以无限次递归执行，并且输出符合预期，就证明命题成立。

## 对向列车行进模型

原命题

$2n= P_1+P_2, \quad 2n>6$

变形为

$P_1+P_2=2n,2n>6$

进一步拆解，变形

$P_1+P_2=3+3+(P_{n+1}-P_n) + (P_{m+1}-P_m)$

中小学数学题常见一种描述就是，两列火车从 A 和 B 两地分别出发，对向而行，问两车几时见面。  
下面对素数数列进行对向列车布置。两列数相加，随着数字增大，谋求填补全部偶数数列。看算法能否无限次执行下去，并且没有偶数遗漏。

1	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
2			1	2	2	4	2	4	2	4	6	2	6	4	2	4	6	6	
3	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
4			3																
5			6																
6	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
7			5	3															
8			8	8															
9	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
10			7	5	3														
11			10	10	10														
12	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
13			11	7	5	3													
14			14	12	12	14													
15	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
16			13	11	7	5	3												
17			16	16	14	16	16												
18	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
19			17	13	11	7	5	3											
20			20	18	18	18	18	20											
21	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
22			19	17	13	11	7	5	3										
23			22	22	20	22	20	22	22										
24	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
25			23	19	17	13	11	7	5	3									
26			26	24	24	24	24	24	24	26									
27	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
28			29	23	19	17	13	11	7	5	3								
29			32	28	26	28	26	28	26	28	32								
30	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
31			31	29	23	19	17	13	11	7	5	3							
32			34	34	30	30	30	30	30	30	34	34							
30	1	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	60
31			37	31	29	23	19	17	13	11	7	5	3						
32			40	36	36	34	32	34	32	34	36	36	40						

图 10

图 10 呈现了  $P_1+P_2=2n,2n>6$ ，这个算式的计算结果和数据分布规律。

下一步需要论证的是在素数趋于无穷大的方向上，在更大的范围里，由已知素数的分布规律为基础是否能够确认作为计算结果的偶数数列是连续的。如果出现不连续，由怎样的计算步骤可以找到缺失的偶数对应的素数的解。算法演绎能够确认缺失的偶数对应的解必然存在，则算法可以在无穷大范围内成立。算法成立，命题证明结束。

图 10 第 3 到第 11 行，  
素数 3,5,7 两两相加的计算结果是 6,8,10。结果是连续的偶数数列。呈现这一结果的原因是  
 $5-3=2$ ，  
 $7-5=2$

图 10 第 14 行。  
 $11+3=14$ ，  
这一结果与  
 $7+3=10$  出现偶数断点，缺少 12。

缺少 12 的原因是 7 和 11 中间删除了 9,  $9=3*3$ 。

图 6 表明了素数 7 和 11 距离变大的原理。

11-7=4,  
4=2+2

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

图 6

3+11=14，出现偶数断点。寻找一组素数解填补断点的算法是

以算式中小素数 3 为起点向右，间距为+2 的下一个素数是 5。以 11 为起点向左，最近的素数是 7，间距为-4。5+7=12，得到距离 14 为-2 的偶数 12。

## 从素数序列得到完整偶数序列

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		1		2		2				4		2				4		2				4							6
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
2						6				4		2				4						6							6
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
2						6				4		2						6				4							6
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
						8				4		2				4		2				4							
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
						14				4						6		2											10
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
2						6						6				4						6							6
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210
2										10		2				4		2											

(P<sub>n+1</sub>-P<sub>n</sub>)为相邻两个素数的间距。

素数数列成对向列车布置。

素数 P<sub>n</sub> 与不大于自身的所有素数相加。结果等于得到位于 P<sub>n</sub> 右侧的一个重复自身左侧依次间距的数列，记为 P<sub>n</sub>+P<sub>(1~n)</sub>

素数 P<sub>n+1</sub> 与不大于自身的所有素数相加，结果等于得到位于 P<sub>n+1</sub> 右侧的一个重复自身左侧依次间距的数列。记为 P<sub>n+1</sub>+P<sub>(1~n+1)</sub>

数列 P<sub>n</sub>+P<sub>(1~n)</sub> 与数列 P<sub>n+1</sub>+P<sub>(1~n+1)</sub> 叠加的结果是间距(P<sub>n+1</sub>-P<sub>n</sub>)被素数 P<sub>n</sub> 左侧素数间距(P<sub>m+1</sub>-P<sub>m</sub>)与(P<sub>m</sub>-P<sub>m-1</sub>)覆盖。

前文算法分析已经得出结论。素数间距的变化规律是

$$(P_{n+1}-P_n)=(P_{m+1}-P_m)+(P_m-P_{m-1}),n>m$$

或者说

$$P_{n+1}-P_{m+1}=P_n-P_{m-1},n>m$$

素数位置的循环重复分布造成存在无穷多形如 P<sub>n+1</sub>-P<sub>m+1</sub>=P<sub>n</sub>-P<sub>m-1</sub> 的等差素数存在。

小于素数 P<sub>n</sub> 的全部素数两两相加的结果是素数间距(P<sub>m+1</sub>-P<sub>m</sub>)与(P<sub>m</sub>-P<sub>m-1</sub>)全部叠加，或者说递归拆分。素数间距递归拆分的极限是 2。所以小于素数 P<sub>n</sub> 的全部素数两两相加计算结果的最小间距极限也是 2。最终结论，全部素数两两相加的结果是全部偶数序列。