

### Case: $h = 0 \pmod 3$

Baseline Riesel test numbers			
h * 2^n-1			
[ n, n+1000 )		h = 3*base_n [ h, h+6000 )	
base_n	n_beyond	hase_h	h_beyond
4194304	4195304	12582912	12588912
4331116	4332116	12993348	12999348
4885002	4886002	14655006	14661006
5209020	5210020	15627060	15633060
6286862	6287862	18860586	18866586
7676777	7677777	23030331	23036331
8388608	8389608	25165824	25171824
Standard Deviation			
Average Jacobi ops to find 1st v(1)			

Baseline No Jacobi cache					
Average Jacobi ops to find 1st v(1)					
search starting at 3		sorted by v(1)		reverse sort by freq	
integer search 1st v(1)	odd search 1st v(1)	known 1st v(1)	odd known 1st v(1)	known 1st v(1)	odd known 1st v(1)
9.878	5.900	4.926	4.875	4.864	4.852
9.884	5.898	4.929	4.875	4.868	4.852
9.873	5.892	4.923	4.871	4.862	4.847
9.867	5.891	4.922	4.871	4.861	4.847
9.891	5.901	4.930	4.877	4.869	4.853
9.878	5.897	4.925	4.874	4.863	4.849
9.880	5.901	4.926	4.876	4.865	4.852
0.008		0.004		0.003	
9.879	5.897	4.926	4.874	4.865	4.850

Baseline Jacobi cache					
Average Jacobi ops to find 1st v(1)					
search starting at 3		sorted by v(1)		reverse sort by freq	
integer search 1st v(1)	odd search 1st v(1)	known 1st v(1)	odd known 1st v(1)	known 1st v(1)	odd known 1st v(1)
7.559	4.620	4.433	4.364	4.369	4.350
7.561	4.618	4.435	4.363	4.371	4.349
7.555	4.615	4.431	4.359	4.367	4.345
7.551	4.614	4.430	4.359	4.366	4.345
7.565	4.620	4.436	4.363	4.373	4.349
7.557	4.617	4.432	4.361	4.367	4.346
7.559	4.621	4.434	4.364	4.370	4.350
0.004		0.003		0.002	
7.558	4.618	4.433	4.362	4.369	4.348

Baseline Cache advantage											
Jacobi cache / No Jacobi cache											
search starting at 3		sorted by v(1)		reverse sort by freq		search starting at 3		sorted by v(1)		reverse sort by freq	
integer search 1st v(1)	odd search 1st v(1)	known 1st v(1)	odd known 1st v(1)	known 1st v(1)	odd known 1st v(1)	integer search 1st v(1)	odd search 1st v(1)	known 1st v(1)	odd known 1st v(1)	known 1st v(1)	odd known 1st v(1)
1.3068	1.2771	1.1112	1.1171	1.1133	1.1154	1.3072	1.2772	1.1114	1.1174	1.1137	1.1157
1.3068	1.2767	1.1110	1.1175	1.1134	1.1155	1.3067	1.2768	1.1111	1.1175	1.1134	1.1155
1.3075	1.2773	1.1114	1.1178	1.1134	1.1159	1.3071	1.2772	1.1112	1.1176	1.1136	1.1157
1.3071	1.2770	1.1110	1.1173	1.1133	1.1154	1.3071	1.2770	1.1110	1.1173	1.1133	1.1154
0.0003		0.0002		0.0002		0.0002		0.0002		0.0002	
1.3070	1.2770	1.1112	1.1174	1.1134	1.1156	1.3070	1.2770	1.1112	1.1174	1.1134	1.1156

3, 5, 9, 11, 15, 17, <b>20</b> , 21, 27, 29, <b>32</b> , 35, <b>36</b> , 39, 41, <b>44</b> , 45, 49, 51, 55, <b>56</b> , 57, 59, 65, <b>66</b> , 67, 69, 71, <b>72</b> , <b>74</b> , 77, <b>80</b> , 81, <b>84</b> , 87, <b>90</b> , 95, 99, 101, <b>104</b> , 105, 109, 111, <b>116</b> , 125, 135											
3, 5, 9, 11, 15, 17, 21, 27, 29, 31, 35, 39, 41, 45, 49, 51, 53, 55, 57, 59, 65, 67, 69, 71, 77, 81, 83, 85, 87, 95, 99, 101, 105, 107, 109, 111, 125, 129, 135, 139, 141, 149, 155, 165											
3, 5, 9, 11, 15, 17, 21, 29, <b>20</b> , 27, 35, 36, 39, 41, 45, <b>32</b> , 51, <b>44</b> , 56, <b>49</b> , 59, <b>57</b> , 65, <b>55</b> , 69, 71, 77, <b>66</b> , 81, 95, <b>72</b> , 80, <b>67</b> , 99, <b>84</b> , <b>74</b> , 90, 104, 105, <b>87</b> , 116, <b>101</b> , 109, 125, <b>111</b> , 135											
3, 5, 9, 11, 15, 17, 21, 29, <b>27</b> , 35, 39, 41, <b>31</b> , 45, 51, 55, <b>49</b> , 59, 69, 71, <b>65</b> , <b>57</b> , 85, <b>81</b> , 95, 99, <b>77</b> , <b>53</b> , 67, 105, <b>101</b> , 109, 125, <b>87</b> , 129, <b>83</b> , 111, 155, <b>107</b> , 135, 139, 141, 149, 165											
3, 5, 9, 11, 15, 17, 21, 27, 29, 35, 39, 41, 45, 51, 57, 59, 65, 69, 81											
3248337 4331118 ++++++0++-0+-0											