

10

Busqueda y ordenamiento binario

Búsqueda binaria

La idea principal de la búsqueda binaria es llevar un registro del rango actual de intentos razonables. Digamos que estoy pensando en un número entre uno y 100. Si ya intentaste decir 25 y te dije que mi número es más grande, y ya intentaste decir 81 y te dije que mi número es más chico, entonces los números en el rango de 26 a 80 son los únicos intentos razonables. Aquí, la sección roja de la recta numérica contiene los intentos razonables, y la sección negra muestra los intentos que hemos descartado:



En cada turno, haces un intento que divide el conjunto de intentos razonables en dos rangos de aproximadamente el mismo tamaño. Si tu intento no es correcto, entonces te digo si es muy alto o muy bajo, y puedes eliminar aproximadamente la mitad de los intentos razonables. Por ejemplo, si el rango actual de los intentos razonables es de 26 a 80, intentarías adivinar a la mitad del camino, $(26+80)/2 = 53$. Si después te digo que 53 es demasiado alto, puedes eliminar todos los números de 53 a 80, dejando 26 a 52 como el nuevo rango de intentos razonables, reduciendo a la mitad el tamaño del rango.



Para el juego de adivinar, podemos llevar un registro del conjunto de intentos razonables al usar unas cuantas variables. Sea la variable el intento razonable mínimo actual para esta ronda, y sea la variable el intento razonable máximo actual. La *entrada* (o *input* en inglés) al problema es el número, el mayor número posible que tu oponente está pensando. Suponemos que el menor número posible es uno, pero sería fácil modificar el algoritmo para tener el menor número posible como una segunda entrada.

Aquí está una descripción paso a paso de cómo usar la búsqueda binaria para jugar el juego de adivinar:

1. Sea `min=1` y `max=n`.
2. Haz `(max+min)/2` tu estimación, redondeado hacia abajo de modo que sea un entero.
3. Si adivinaste el número, **detente**. ¡Lo encontraste!
4. Si el intento fue demasiado **bajo**, haz que `min` sea uno más grande que el intento.
5. Si el intento fue demasiado **alto**, haz que `max` sea uno menos que el intento.
6. Regresa al paso **dos**.

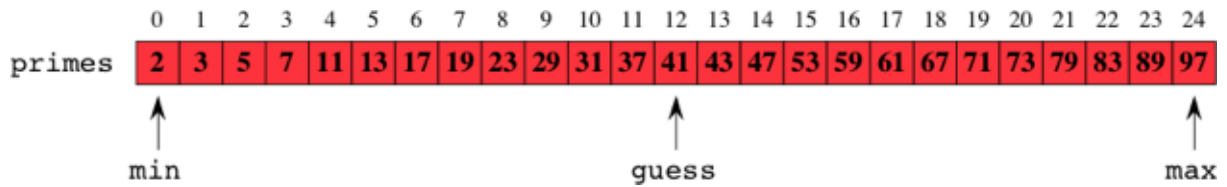
Podríamos hacer que la descripción fuera todavía más precisa al describir claramente las entradas y las salidas del algoritmo y al clarificar qué queremos decir con instrucciones como "adivina un número" y "detente". Pero esto es suficiente detalle por ahora.

Implementar la Búsqueda Binaria de un Arreglo

Aquí hay un arreglo en C++ de los primeros 25 números primos, en orden:

```
int primes[] = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 187, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 473, 479, 487, 491, 499, 503, 509, 521, 523, 527, 531, 539, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 623, 629, 631, 637, 641, 643, 647, 653, 659, 661, 667, 671, 673, 677, 683, 687, 691, 697, 701, 709, 713, 719, 727, 733, 739, 743, 749, 757, 761, 769, 773, 779, 787, 791, 797, 809, 811, 817, 821, 823, 827, 829, 833, 837, 839, 843, 847, 853, 857, 859, 863, 867, 877, 881, 883, 887, 893, 897, 901, 907, 911, 913, 917, 919, 923, 927, 931, 937, 941, 943, 947, 953, 959, 967, 971, 973, 977, 983, 989, 991, 993, 997, 1001, 1003, 1007, 1009, 1013, 1017, 1019, 1021, 1023, 1027, 1029, 1031, 1033, 1037, 1039, 1043, 1047, 1049, 1051, 1053, 1057, 1059, 1063, 1067, 1069, 1073, 1077, 1079, 1081, 1083, 1087, 1091, 1093, 1097, 1101, 1103, 1107, 1109, 1113, 1117, 1119, 1121, 1123, 1127, 1129, 1133, 1137, 1139, 1143, 1147, 1149, 1153, 1157, 1159, 1163, 1167, 1169, 1173, 1177, 1179, 1181, 1183, 1187, 1191, 1193, 1197, 1201, 1203, 1207, 1209, 1211, 1213, 1217, 1219, 1223, 1227, 1229, 1231, 1233, 1237, 1239, 1243, 1247, 1249, 1253, 1257, 1259, 1263, 1267, 1269, 1271, 1273, 1277, 1279, 1283, 1287, 1289, 1291, 1293, 1297, 1301, 1303, 1307, 1309, 1313, 1317, 1319, 1321, 1323, 1327, 1329, 1331, 1333, 1337, 1339, 1343, 1347, 1349, 1351, 1353, 1357, 1359, 1363, 1367, 1369, 1371, 1373, 1377, 1379, 1381, 1383, 1387, 1391, 1393, 1397, 1401, 1403, 1407, 1409, 1411, 1413, 1417, 1419, 1421, 1423, 1427, 1429, 1431, 1433, 1437, 1439, 1443, 1447, 1449, 1451, 1453, 1457, 1459, 1463, 1467, 1469, 1471, 1473, 1477, 1479, 1481, 1483, 1487, 1491, 1493, 1497, 1501, 1503, 1507, 1509, 1511, 1513, 1517, 1519, 1521, 1523, 1527, 1529, 1531, 1533, 1537, 1539, 1543, 1547, 1549, 1551, 1553, 1557, 1559, 1563, 1567, 1569, 1571, 1573, 1577, 1579, 1581, 1583, 1587, 1591, 1593, 1597, 1601, 1603, 1607, 1609, 1611, 1613, 1617, 1619, 1621, 1623, 1627, 1629, 1631, 1633, 1637, 1639, 1643, 1647, 1649, 1651, 1653, 1657, 1659, 1663, 1667, 1669, 1671, 1673, 1677, 1679, 1681, 1683, 1687, 1691, 1693, 1697, 1701, 1703, 1707, 1709, 1711, 1713, 1717, 1719, 1721, 1723, 1727, 1729, 1731, 1733, 1737, 1739, 1743, 1747, 1749, 1751, 1753, 1757, 1759, 1763, 1767, 1769, 1771, 1773, 1777, 1779, 1781, 1783, 1787, 1791, 1793, 1797, 1801, 1803, 1807, 1809, 1811, 1813, 1817, 1819, 1821, 1823, 1827, 1829, 1831, 1833, 1837, 1839, 1843, 1847, 1849, 1851, 1853, 1857, 1859, 1863, 1867, 1869, 1871, 1873, 1877, 1879, 1881, 1883, 1887, 1891, 1893, 1897, 1901, 1903, 1907, 1909, 1911, 1913, 1917, 1919, 1921, 1923, 1927, 1929, 1931, 1933, 1937, 1939, 1943, 1947, 1949, 1951, 1953, 1957, 1959, 1963, 1967, 1969, 1971, 1973, 1977, 1979, 1981, 1983, 1987, 1991, 1993, 1997, 2001, 2003, 2007, 2009, 2011, 2013, 2017, 2019, 2021, 2023, 2027, 2029, 2031, 2033, 2037, 2039, 2043, 2047, 2049, 2051, 2053, 2057, 2059, 2063, 2067, 2069, 2071, 2073, 2077, 2079, 2081, 2083, 2087, 2091, 2093, 2097, 2101, 2103, 2107, 2109, 2111, 2113, 2117, 2119, 2121, 2123, 2127, 2129, 2131, 2133, 2137, 2139, 2143, 2147, 2149, 2151, 2153, 2157, 2159, 2163, 2167, 2169, 2171, 2173, 2177, 2179, 2181, 2183, 2187, 2191, 2193, 2197, 2201, 2203, 2207, 2209, 2211, 2213, 2217, 2219, 2221, 2223, 2227, 2229, 2231, 2233, 2237, 2239, 2243, 2247, 2249, 2251, 2253, 2257, 2259, 2263, 2267, 2269, 2271, 2273, 2277, 2279, 2281, 2283, 2287, 2291, 2293, 2297, 2301, 2303, 2307, 2309, 2311, 2313, 2317, 2319, 2321, 2323, 2327, 2329, 2331, 2333, 2337, 2339, 2343, 2347, 2349, 2351, 2353, 2357, 2359, 2363, 2367, 2369, 2371, 2373, 2377, 2379, 2381, 2383, 2387, 2391, 2393, 2397, 2401, 2403, 2407, 2409, 2411, 2413, 2417, 2419, 2421, 2423, 2427, 2429, 2431, 2433, 2437, 2439, 2443, 2447, 2449, 2451, 2453, 2457, 2459, 2463, 2467, 2469, 2471, 2473, 2477, 2479, 2481, 2483, 2487, 2491, 2493, 2497, 2501, 2503, 2507, 2509, 2511, 2513, 2517, 2519, 2521, 2523, 2527, 2529, 2531, 2533, 2537, 2539, 2543, 2547, 2549, 2551, 2553, 2557, 2559, 2563, 2567, 2569, 2571, 2573, 2577, 2579, 2581, 2583, 2587, 2591, 2593, 2597, 2601, 2603, 2607, 2609, 2611, 2613, 2617, 2619, 2621, 2623, 2627, 2629, 2631, 2633, 2637, 2639, 2643, 2647, 2649, 2651, 2653, 2657, 2659, 2663, 2667, 2669, 2671, 2673, 2677, 2679, 2681, 2683, 2687, 2691, 2693, 2697, 2701, 2703, 2707, 2709, 2711, 2713, 2717, 2719, 2721, 2723, 2727, 2729, 2731, 2733, 2737, 2739, 2743, 2747, 2749, 2751, 2753, 2757, 2759, 2763, 2767, 2769, 2771, 2773, 2777, 2779, 2781, 2783, 2787, 2791, 2793, 2797, 2801, 2803, 2807, 2809, 2811, 2813, 2817, 2819, 2821, 2823, 2827, 2829, 2831, 2833, 2837, 2839, 2843, 2847, 2849, 2851, 2853, 2857, 2859, 2863, 2867, 2869, 2871, 2873, 2877, 2879, 2881, 2883, 2887, 2891, 2893, 2897, 2901, 2903, 2907, 2909, 2911, 2913, 2917, 2919, 2921, 2923, 2927, 2929, 2931, 2933, 2937, 2939, 2943, 2947, 2949, 2951, 2953, 2957, 2959, 2963, 2967, 2969, 2971, 2973, 2977, 2979, 2981, 2983, 2987, 2991, 2993, 2997, 3001, 3003, 3007, 3009, 3011, 3013, 3017, 3019, 3021, 3023, 3027, 3029, 3031, 3033, 3037, 3039, 3043, 3047, 3049, 3051, 3053, 3057, 3059, 3063, 3067, 3069, 3071, 3073, 3077, 3079, 3081, 3083, 3087, 3091, 3093, 3097, 3101, 3103, 3107, 3109, 3111, 3113, 3117, 3119, 3121, 3123, 3127, 3129, 3131, 3133, 3137, 3139, 3143, 3147, 3149, 3151, 3153, 3157, 3159, 3163, 3167, 3169, 3171, 3173, 3177, 3179, 3181, 3183, 3187, 3191, 3193, 3197, 3201, 3203, 3207, 3209, 3211, 3213, 3217, 3219, 3221, 3223, 3227, 3229, 3231, 3233, 3237, 3239, 3243, 3247, 3249, 3251, 3253, 3257, 3259, 3263, 3267, 3269, 3271, 3273, 3277, 3279, 3281, 3283, 3287, 3291, 3293, 3297, 3301, 3303, 3307, 3309, 3311, 3313, 3317, 3319, 3321, 3323, 3327, 3329, 3331, 3333, 3337, 3339, 3343, 3347, 3349, 3351, 3353, 3357, 3359, 3363, 3367, 3369, 3371, 3373, 3377, 3379, 3381, 3383, 3387, 3391, 3393, 3397, 3401, 3403, 3407, 3409, 3411, 3413, 3417, 3419, 3421, 3423, 3427, 3429, 3431, 3433, 3437, 3439, 3443, 3447, 3449, 3451, 3453, 3457, 3459, 3463, 3467, 3469, 3471, 3473, 3477, 3479, 3481, 3483, 3487, 3491, 3493, 3497, 3501, 3503, 3507, 3509, 3511, 3513, 3517, 3519, 3521, 3523, 3527, 3529, 3531, 3533, 3537, 3539, 3543, 3547, 3549, 3551, 3553, 3557, 3559, 3563, 3567, 3569, 3571, 3573, 3577, 3579, 3581, 3583, 3587, 3591, 3593, 3597, 3601, 3603, 3607, 3609, 3611, 3613, 3617, 3619, 3621, 3623, 3627, 3629, 3631, 3633, 3637, 3639, 3643, 3647, 3649, 3651, 3653, 3657, 3659, 3663, 3667, 3669, 3671, 3673, 3677, 3679, 3681, 3683, 3687, 3691, 3693, 3697, 3701, 3703, 3707, 3709, 3711, 3713, 3717, 3719, 3721, 3723, 3727, 3729, 3731, 3733, 3737, 3739, 3743, 3747, 3749, 3751, 3753, 3757, 3759, 3763, 3767, 3769, 3771, 3773, 3777, 3779, 3781, 3783, 3787, 3791, 3793, 3797, 3801, 3803, 3807, 3809, 3811, 3813, 3817, 3819, 3821, 3823, 3827, 3829, 3831, 3833, 3837, 3839, 3843, 3847, 3849, 3851, 3853, 3857, 3859, 3863, 3867, 3869, 3871, 3873, 3877, 3879, 3881, 3883, 3887, 3891, 3893, 3897, 3901, 3903, 3907, 3909, 3911, 3913, 3917, 3919, 3921, 3923, 3927, 3929, 3931, 3933, 3937, 3939, 3943, 3947, 3949, 3951, 3953, 3957, 3959, 3963, 3967, 3969, 3971, 3973, 3977, 3979, 3981, 3983, 3987, 3991, 3993, 3997, 4001, 4003, 4007, 4009, 4011, 4013, 4017, 4019, 4021, 4023, 4027, 4029, 4031, 4033, 4037, 4039, 4043, 4047, 4049, 4051, 4053, 4057, 4059, 4063, 4067, 4069, 4071, 4073, 4077, 4079, 4081, 4083, 4087, 4091, 4093, 4097, 4101, 4103, 4107, 4109, 4111, 4113, 4117, 4119, 4121, 4123, 4127, 4129, 4131, 4133, 4137, 4139, 4143, 4147, 4149, 4151, 4153, 4157, 4159, 4163, 4167, 4169, 4171, 4173, 4177, 4179, 4181, 4183, 4187, 4191, 4193, 4197, 4201, 4203, 4207, 4209, 4211, 4213, 4217, 4219, 4221, 4223, 4227, 4229, 4231, 4233, 4237, 4239, 4243, 4247, 4249, 4251, 4253, 4257, 4259, 4263, 4267, 4269, 4271, 4273, 4277, 4279, 4281, 4283, 4287, 4291, 4293, 4297, 4301, 4303, 4307, 4309, 4311, 4313, 4317, 4319, 4321, 4323, 4327, 4329, 4331, 4333, 4337, 4339, 4343, 4347, 4349, 4351, 4353, 4357, 4359, 4363, 4367, 4369, 4371, 4373, 4377, 4379, 4381, 4383, 4387, 4391, 4393, 4397, 4401, 4403, 4407, 4409, 4411, 4413, 4417, 4419, 4421, 4423, 4427, 4429, 4431, 4433, 4437, 4439, 4443, 4447, 4449, 4451, 4453, 4457, 4459, 4463, 4467, 4469, 4471, 4473, 4477, 4479, 4481, 4483, 4487, 4491, 4493, 4497, 4501, 4503, 4507, 4509, 4511, 4513, 4517, 4519, 4521, 4523, 4527, 4529, 4531, 4533, 4537, 4539, 4543, 4547, 4549, 4551, 4553, 4557, 4559, 4563, 4567, 4569, 4571, 4573, 4577, 4579, 4581, 4583, 4587, 4591, 4593, 4597, 4601, 4603, 4607, 4609, 4611, 4613, 4617, 4619, 4621, 4623, 4627, 4629, 4631, 4633, 4637, 4639, 4643, 4647, 4649, 4651, 4653, 4657, 4659, 4663, 4667, 4669, 4671, 4673, 4677, 4679, 4681, 4683, 4687, 4691, 4693, 4697, 4701, 4703, 4707, 4709, 4711, 4713, 4717, 4719, 4721, 4723, 4727, 4729, 4731, 4733, 4737, 4739, 4743, 4747, 4749, 4751, 4753, 4757, 4759, 4763, 4767, 4769, 4771, 4773, 4777, 4779, 4781, 4783, 4787, 4791, 4793, 4797, 4801, 4803, 4807, 4809, 4811, 4813, 4817, 4819, 4821, 4823, 4827, 4829, 4831, 4833, 4837, 4839, 4843, 4847, 4849, 4851, 4853, 4857, 4859, 4863, 4867, 4869, 4871, 4873, 4877, 4879, 4881, 4883, 4887, 4891, 4893, 4897, 4901, 4903, 4907, 4909, 4911, 4913, 4917, 4919, 4921, 4923, 4927, 4929, 4931, 4933, 4937, 4939, 4943, 4947, 4949, 4951, 4953, 4957, 4959, 4963, 4967, 4969, 4971, 4973, 4977, 4979, 4981, 4983, 4987, 4991, 4993, 4997, 5001, 5003, 5007, 5009, 5011, 5013, 5017, 5019, 5021, 5023, 5027, 5029, 5031, 5033, 5037, 5039, 5043, 5047, 5049, 5051, 5053, 5057, 5059, 5063, 5067, 5069, 5071, 5073, 5077, 5079, 5081, 5083, 5087, 5091, 5093, 5097, 5101, 5103, 5107, 5109, 5111, 5113, 5117, 5119, 5121, 5123, 5127, 5129, 5131, 5133, 5137, 5139, 5143, 5147, 5149, 5151, 5153, 5157, 5159, 5163, 5167, 5169, 5171, 5173, 5177, 5179, 5181, 5183, 5187, 5191, 5193, 5197, 5201, 5203, 5207, 5209, 5211, 5213, 5217, 5219, 5221, 5223, 5227, 5229, 5231, 5233, 5237, 5239, 5243, 5247, 5249, 5251, 5253, 5257, 5259, 5263, 5267, 5269, 5271, 5273, 5277, 5279, 5281, 5283, 5287, 5291, 5293, 5297, 5301, 5303, 5307, 5309, 5311, 5313, 5317, 5319, 5321, 5323, 5327, 5329, 5331, 5333, 5337, 5339, 5343, 5347, 5349, 5351, 5353, 5357, 5359, 5363, 5367, 5369, 5371, 5373, 5377, 5379, 5381, 5383, 5387, 5391, 5393, 5397, 5401, 5403, 5407, 5409, 5411, 5413, 5417, 5419, 5421, 5423, 5427, 5429, 5431, 5433, 5437, 5439, 5443, 5447, 5449, 5451, 5453, 5457, 5459, 5463, 5467, 5469, 5471, 5473, 5477, 5479, 5481, 5483, 5487, 5491, 5493, 5497, 5501, 5503, 5507, 5509, 5511, 5513, 5517, 5519, 5521, 5523, 5527, 5529, 5531, 5533, 5537, 5539, 5543, 5547, 5549, 5551, 5553, 5557, 5559, 5563, 5567, 5569, 5571, 5573, 5577, 5579, 5581, 5583, 5587, 5591, 5593, 5597, 5601, 5603, 5607, 5609, 5611, 5613, 5617, 5619, 5621, 5623, 5627, 5629, 5631, 5633, 5637, 5639, 5643, 5647, 5649, 5651, 5653, 5657, 5659, 5663, 5667, 5669, 5671, 5673, 5677, 5679, 5681, 5683, 5687, 5691, 5693, 5697, 5701, 5703, 5707, 5709, 5711, 5713, 5717, 5719, 5721, 5723, 5727, 5729, 5731, 5733, 5737, 5739, 5743, 5747, 5749, 5751, 5753, 5757, 5759, 5763, 5767, 5769, 5771, 5773, 5777, 5779, 5781, 5783, 5787, 5791, 5793, 5797, 5801, 5803, 5807, 5809, 5811, 5813, 5817, 5819, 5821, 5823, 5827, 5829, 5831, 5833, 5837, 5839, 5843, 5847, 5849, 5851, 5853, 5857, 5859, 5863, 5867, 5869, 5871, 5873, 5877, 5879, 5881, 5883, 5887, 5891, 5893, 5897, 5901, 5903, 5907, 5909, 5911, 5913, 5917, 5919, 5921, 5923, 5927, 5929, 5931, 5933, 5937, 5939, 5943, 5947, 5949, 5951, 5953, 5957, 5959, 5963, 5967, 5969, 5971, 5973, 5977, 5979, 5981, 5983, 5987, 5991, 5993, 5997, 6001, 6003, 6007, 6009, 6011, 6013, 601
```

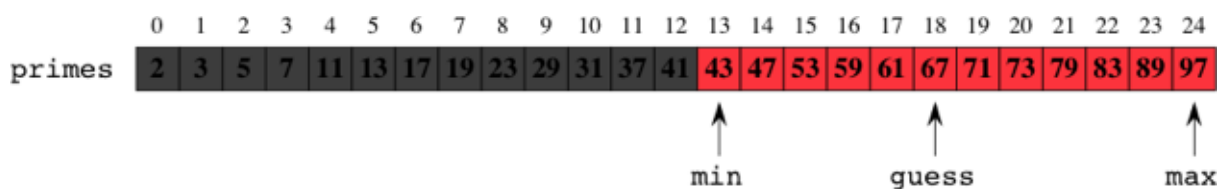
búsqueda binaria sería entonces en el índice 12 (que es $(0 + 24) / 2$). ¿`primes[12]` es igual a 67? No, `primes[12]` es 41.



¿El índice que estamos buscando es mayor o menor que 12? Como los valores en el arreglo están en orden ascendente, y $41 < 67$, el valor 67 debe estar a la derecha del índice 12. En otras palabras, el índice que estamos tratando de adivinar debe ser mayor que 12. Actualizamos el valor de `min` a $12 + 1$, o 13, y dejamos `max` sin cambio en 24.



¿Cuál es el siguiente índice que intentamos? El promedio de 13 y 24 es 18.5, el cual redondeamos hacia abajo a 18, puesto que un índice de un arreglo debe ser un entero. Encontramos que `primes[18]` es 67.



El algoritmo de la búsqueda binaria se detiene en este punto, ya que ha encontrado la respuesta. Solo tomó dos intentos, en lugar de 19 intentos que le hubiera tomado a la búsqueda lineal.

Esto mismo en código se veería tal que:

```
int size = 25;
int primes[] = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37,
               41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97};

int binarySearch(int x) {
    int min = 0;
    int max = size - 1;
    while (min <= max) {
        int guess = min + (max - min) / 2;
        if (primes[guess] == x) {
            return guess;
        } else if (primes[guess] < x) {
            min = guess + 1;
        } else {
            max = guess - 1;
        }
    }
    return -1;
}
```

Tiempo de Ejecución de la Búsqueda Binaria

La idea clave es que cuando una búsqueda binaria hace un intento incorrecto, la porción del arreglo que contiene los intentos razonables se reduce por lo menos a la mitad. Si la porción razonable tenía 32 elementos, entonces un intento incorrecto la reduce para que tenga a lo más 16. La búsqueda binaria reduce el tamaño de la porción razonable a la mitad después de cada intento incorrecto.

Si empezamos con un arreglo de longitud 8, los intentos incorrectos reducen el tamaño de la porción razonable a 4, luego a 2 y luego a 1. Una vez que la porción razonable contiene solo un elemento, no ocurren más intentos; el intento para la porción de 1 elemento es correcta o incorrecta, y ya terminamos.

Así que con un arreglo de longitud 8, la búsqueda binaria necesita a lo más cuatro intentos.

Cada vez que duplicamos el tamaño del arreglo, necesitamos a lo más un intento más. Supón que necesitamos a lo más m intentos para un arreglo de longitud n . Entonces, para un arreglo de longitud $2n$, el primer intento corta la porción razonable del arreglo a un tamaño n , y a lo más en m intentos terminamos, dándonos un total de a lo más $m+1$ intentos. Echemos un vistazo al caso general de un arreglo de longitud n . Podemos expresar el número de intentos, en el peor caso, como "el número de veces que dividimos repetidamente a la mitad, empezando en n , hasta obtener el valor de 1, más uno". Pero esa es una manera inconveniente de escribirlo.

Afortunadamente, hay una función matemática que significa lo mismo que el número de veces que dividimos repetidamente a la mitad, empezando en n , hasta obtener el valor de 1: el **logaritmo base 2 de n** . Eso suele escribirse como $\log_2(n)$, pero quizá también lo hayas visto escrito como $\lg(n)$ en escritos de ciencias de las computación.

