Algorithms Final Project
Longest Increasing Subsequence

by Ted Johansmeyer, Andrew Holmes & Todd Weisse

Finding the longest increasing sequence of consecutive or nonconsecutive numbers within a set of random numbers is known as the longest increasing subsequence problem. To tackle this our function needs three vectors; vector<int> a which contains the initial vector, vector<int> b which is constantly extended by the next smallest possible value and contains the indexes in regards to vector a, and a third vector<int> p which tells us the previous element of the subsequence and ultimately helps us reconstruct the result. We need to use a binary search to search for the next smallest value and thus our **runtime is O(nlogn)**. We also leverage a random number generating function to initially fill our starting vector with 1000 random integers. Every time the function is executed a new random vector is solved and printed in the output.

Work Breakdown

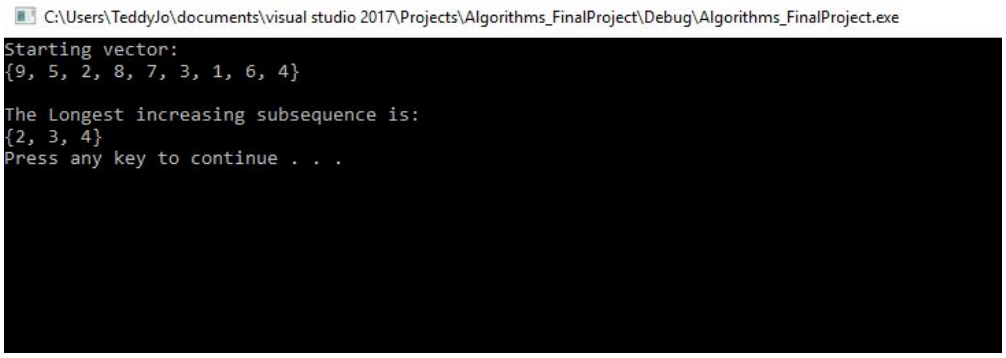
Ted: Designing algorithm, writing algorithm, writing the written portion, determining runtime
Andrew: Designing algorithm, writing algorithm, random number generating function
Todd: Designing algorithm, writing algorithm, verifying test cases

Pseudocode

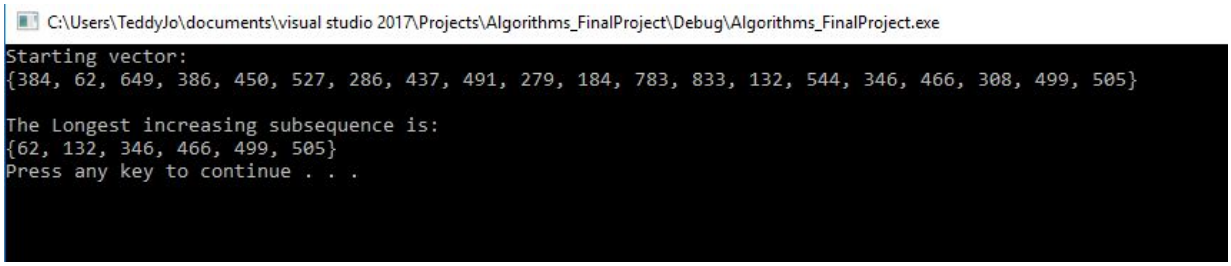
```
1.  longestIncreasingSubsequence(vector<int> &a, vector<int> &b)
2.  if(a is empty) return;          // Special case of empty initial vector
3.  Initialize vector<int>p to size of initial sequence
4.  Push 0 to vector b
5.  for( i = 1; i < size of a; i++)                 // Loop through initial sequence
6.          if(a[b.back()] < a[i])                  // Check the upper bound
7.                  set p[i] to the last element in b and push i onto b; continue
8.          set low to 0 and high to the length of b for the binary search
9.          while(low < high)                       // Start binary search
10.                 mid = (low + high)/2            // Find Mid Point
11.                 if(a[b[mid]] < a[i])
12.                         low = mid + 1
13.                 else
14.                         high = mid              // End Binary Search
15.         if(a[i] < a[b[low]])    // If the element at i is less than the element at b[low]
16.                 p[i] = b[low -1]
17.                 b[low] = i
18. // End of for loop
19. Initialize an int iterator to the size of b & an int index to the last element of b
20. while(it > 0)                                   // Begin reconstructing result to vector b
21.         it--
22.         b[it] = index
23.         index = p[index]
24. // end longestIncreasingSubsequence
```

Sample input & output

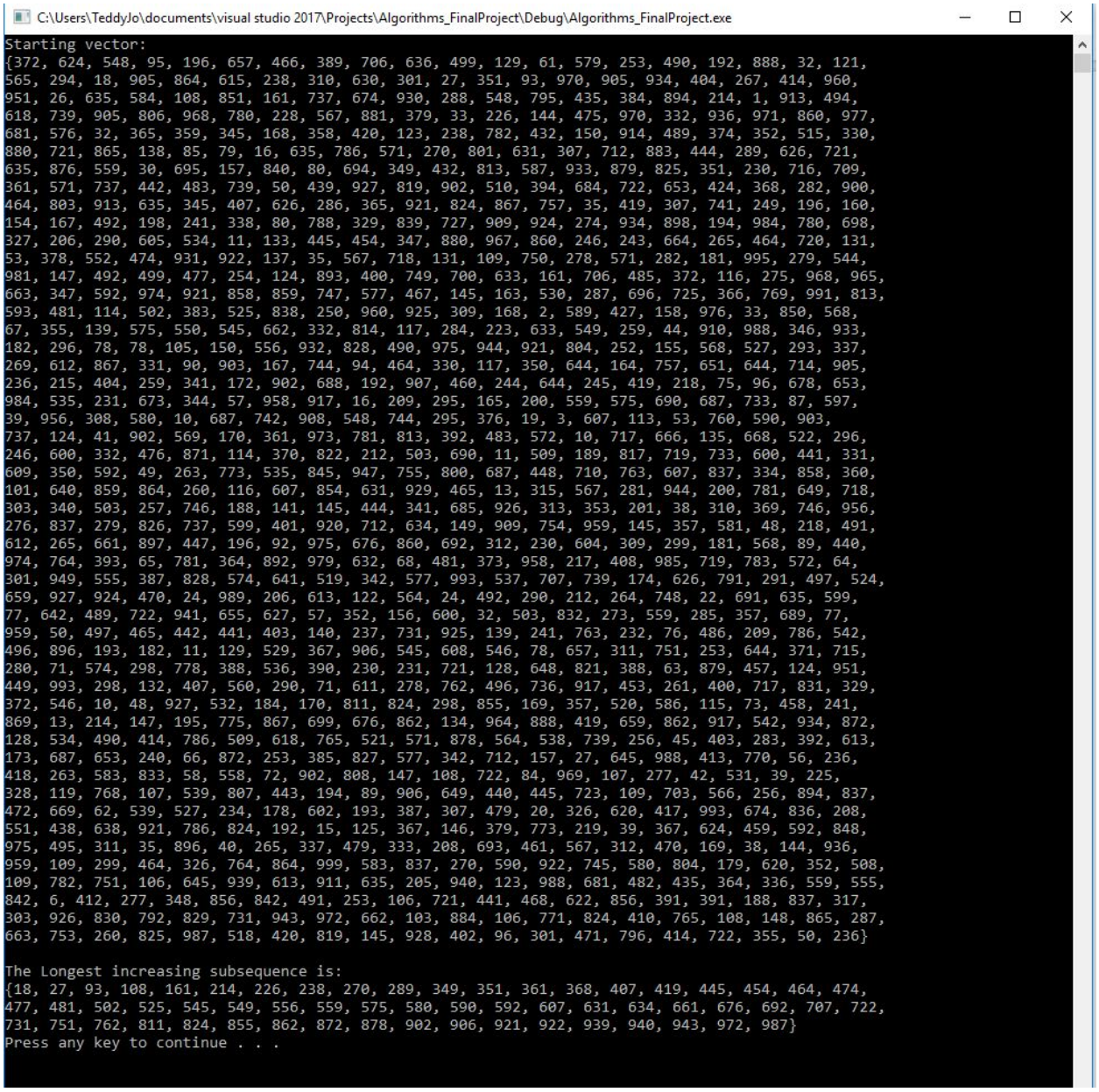
1. Sample input from question



2. n = 20



3. n = 1000


C:\Users\TeddyJo\documents\visual studio 2017\Projects\Algorithms_FinalProject\Debug\Algorithms_FinalProject.exe
Starting vector:
{372, 624, 548, 95, 196, 657, 466, 389, 706, 636, 499, 129, 61, 579, 253, 490, 192, 888, 32, 121,
565, 294, 18, 905, 864, 615, 238, 310, 630, 301, 27, 351, 93, 970, 905, 934, 404, 267, 414, 960,
951, 26, 635, 584, 108, 851, 161, 737, 674, 930, 288, 548, 795, 435, 384, 894, 214, 1, 913, 494,
618, 739, 905, 806, 968, 780, 228, 567, 881, 379, 33, 226, 144, 475, 970, 332, 936, 971, 860, 977,
681, 576, 32, 365, 359, 345, 168, 358, 420, 123, 238, 782, 432, 150, 914, 489, 374, 352, 515, 330,
880, 721, 865, 138, 85, 79, 16, 635, 786, 571, 270, 801, 631, 307, 712, 883, 444, 289, 626, 721,
635, 876, 559, 30, 695, 157, 840, 80, 694, 349, 432, 813, 587, 933, 879, 825, 351, 230, 716, 709,
361, 571, 737, 442, 483, 739, 50, 439, 927, 819, 902, 510, 394, 684, 722, 653, 424, 368, 282, 900,
464, 803, 913, 635, 345, 407, 626, 286, 365, 921, 824, 867, 757, 35, 419, 307, 741, 249, 196, 160,
154, 167, 492, 198, 241, 338, 80, 788, 329, 839, 727, 909, 924, 274, 934, 898, 194, 984, 780, 698,
327, 206, 290, 605, 534, 11, 133, 445, 454, 347, 880, 967, 860, 246, 243, 664, 265, 464, 720, 131,
53, 378, 552, 474, 931, 922, 137, 35, 567, 718, 131, 109, 750, 278, 571, 282, 181, 995, 279, 544,
981, 147, 492, 499, 477, 254, 124, 893, 400, 749, 700, 633, 161, 706, 485, 372, 116, 275, 968, 965,
663, 347, 592, 974, 921, 858, 859, 747, 577, 467, 145, 163, 530, 287, 696, 725, 366, 769, 991, 813,
593, 481, 114, 502, 383, 525, 838, 250, 960, 925, 309, 168, 2, 589, 427, 158, 976, 33, 850, 568,
67, 355, 139, 575, 550, 545, 662, 332, 814, 117, 284, 223, 633, 549, 259, 44, 910, 988, 346, 933,
182, 296, 78, 78, 105, 150, 556, 932, 828, 490, 975, 944, 921, 804, 252, 155, 568, 527, 293, 337,
269, 612, 867, 331, 90, 903, 167, 744, 94, 464, 330, 117, 350, 644, 164, 757, 651, 644, 714, 905,
236, 215, 404, 259, 341, 172, 902, 688, 192, 907, 460, 244, 644, 245, 419, 218, 75, 96, 678, 653,
984, 535, 231, 673, 344, 57, 958, 917, 16, 209, 295, 165, 200, 559, 575, 690, 687, 733, 87, 597,
39, 956, 308, 580, 10, 687, 742, 908, 548, 744, 295, 376, 19, 3, 607, 113, 53, 760, 590, 903,
737, 124, 41, 902, 569, 170, 361, 973, 781, 813, 392, 483, 572, 10, 717, 666, 135, 668, 522, 296,
246, 600, 332, 476, 871, 114, 370, 822, 212, 503, 690, 11, 509, 189, 817, 719, 733, 600, 441, 331,
609, 350, 592, 49, 263, 773, 535, 845, 947, 755, 800, 687, 448, 710, 763, 607, 837, 334, 858, 360,
101, 640, 859, 864, 260, 116, 607, 854, 631, 929, 465, 13, 315, 567, 281, 944, 200, 781, 649, 718,
303, 340, 503, 257, 746, 188, 141, 145, 444, 341, 685, 926, 313, 353, 201, 38, 310, 369, 746, 956,
276, 837, 279, 826, 737, 599, 401, 920, 712, 634, 149, 909, 754, 959, 145, 357, 581, 48, 218, 491,
612, 265, 661, 897, 447, 196, 92, 975, 676, 860, 692, 312, 230, 604, 309, 299, 181, 568, 89, 440,
974, 764, 393, 65, 781, 364, 892, 979, 632, 68, 481, 373, 958, 217, 408, 985, 719, 783, 572, 64,
301, 949, 555, 387, 828, 574, 641, 519, 342, 577, 993, 537, 707, 739, 174, 626, 791, 291, 497, 524,
659, 927, 924, 470, 24, 989, 206, 613, 122, 564, 24, 492, 290, 212, 264, 748, 22, 691, 635, 599,
77, 642, 489, 722, 941, 655, 627, 57, 352, 156, 600, 32, 503, 832, 273, 559, 285, 357, 689, 77,
959, 50, 497, 465, 442, 441, 403, 140, 237, 731, 925, 139, 241, 763, 232, 76, 486, 209, 786, 542,
496, 896, 193, 182, 11, 129, 529, 367, 906, 545, 608, 546, 78, 657, 311, 751, 253, 644, 371, 715,
280, 71, 574, 298, 778, 388, 536, 390, 230, 231, 721, 128, 648, 821, 388, 63, 879, 457, 124, 951,
449, 993, 298, 132, 407, 560, 290, 71, 611, 278, 762, 496, 736, 917, 453, 261, 400, 717, 831, 329,
372, 546, 10, 48, 927, 532, 184, 170, 811, 824, 298, 855, 169, 357, 520, 586, 115, 73, 458, 241,
869, 13, 214, 147, 195, 775, 867, 699, 676, 862, 134, 964, 888, 419, 659, 862, 917, 542, 934, 872,
128, 534, 490, 414, 786, 509, 618, 765, 521, 571, 878, 564, 538, 739, 256, 45, 403, 283, 392, 613,
173, 687, 653, 240, 66, 872, 253, 385, 827, 577, 342, 712, 157, 27, 645, 988, 413, 770, 56, 236,
418, 263, 583, 833, 58, 558, 72, 902, 808, 147, 108, 722, 84, 969, 107, 277, 42, 531, 39, 225,
328, 119, 768, 107, 539, 807, 443, 194, 89, 906, 649, 440, 445, 723, 109, 703, 566, 256, 894, 837,
472, 669, 62, 539, 527, 234, 178, 602, 193, 387, 307, 479, 20, 326, 620, 417, 993, 674, 836, 208,
551, 438, 638, 921, 786, 824, 192, 15, 125, 367, 146, 379, 773, 219, 39, 367, 624, 459, 592, 848,
975, 495, 311, 35, 896, 40, 265, 337, 479, 333, 208, 693, 461, 567, 312, 470, 169, 38, 144, 936,
959, 109, 299, 464, 326, 764, 864, 999, 583, 837, 270, 590, 922, 745, 580, 804, 179, 620, 352, 508,
109, 782, 751, 106, 645, 939, 613, 911, 635, 205, 940, 123, 988, 681, 482, 435, 364, 336, 559, 555,
842, 6, 412, 277, 348, 856, 842, 491, 253, 106, 721, 441, 468, 622, 856, 391, 391, 188, 837, 317,
303, 926, 830, 792, 829, 731, 943, 972, 662, 103, 884, 106, 771, 824, 410, 765, 108, 148, 865, 287,
663, 753, 260, 825, 987, 518, 420, 819, 145, 928, 402, 96, 301, 471, 796, 414, 722, 355, 50, 236}
The Longest increasing subsequence is:
{18, 27, 93, 108, 161, 214, 226, 238, 270, 289, 349, 351, 361, 368, 407, 419, 445, 454, 464, 474,
477, 481, 502, 525, 545, 549, 556, 559, 575, 580, 590, 592, 607, 631, 634, 661, 676, 692, 707, 722,
731, 751, 762, 811, 824, 855, 862, 872, 878, 902, 906, 921, 922, 939, 940, 943, 972, 987}
Press any key to continue . . .