

Erin Alltop

2-2-17

CS261 – Assignment 3 Part 2

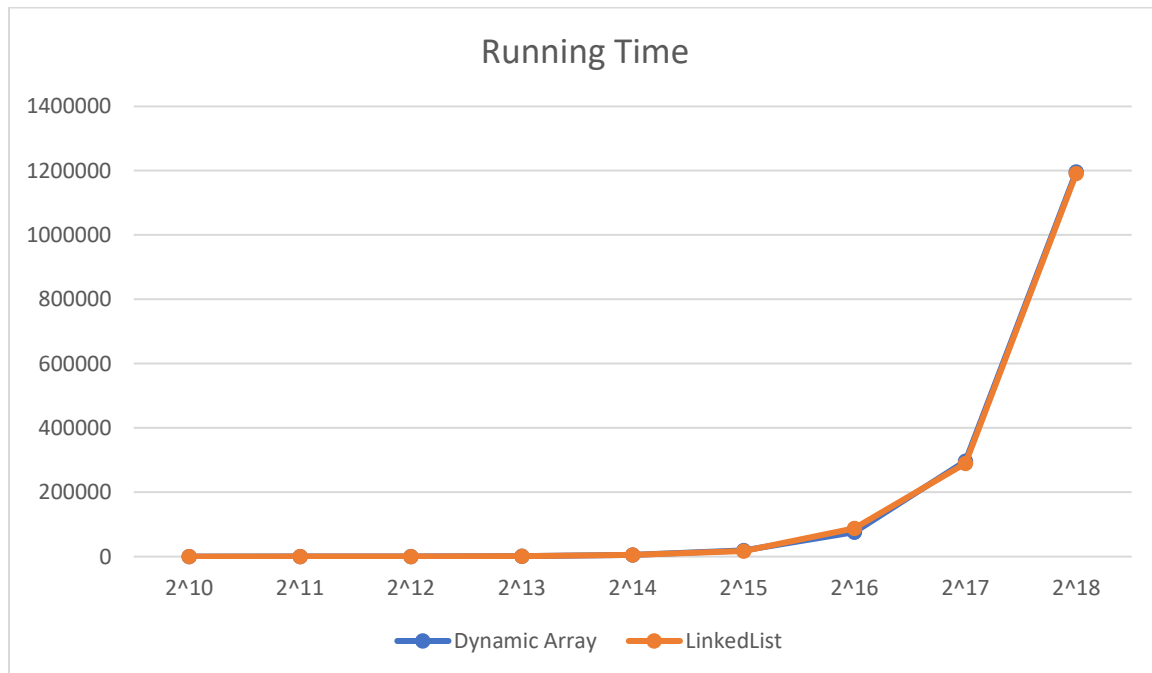
### Dynamic Array Memory Usage and Run Times

Element counts n=	Memory Usage	Run Time in ms
2 <sup>10</sup>	124KB	30ms
2 <sup>11</sup>	124KB	70ms
2 <sup>12</sup>	124KB	320ms
2 <sup>13</sup>	2172KB	1250ms
2 <sup>14</sup>	2172KB	4940ms
2 <sup>15</sup>	2172KB	19310ms
2 <sup>16</sup>	2308KB	76420ms
2 <sup>17</sup>	2356KB	297290ms
2 <sup>18</sup>	2436KB	1.19675e+06ms

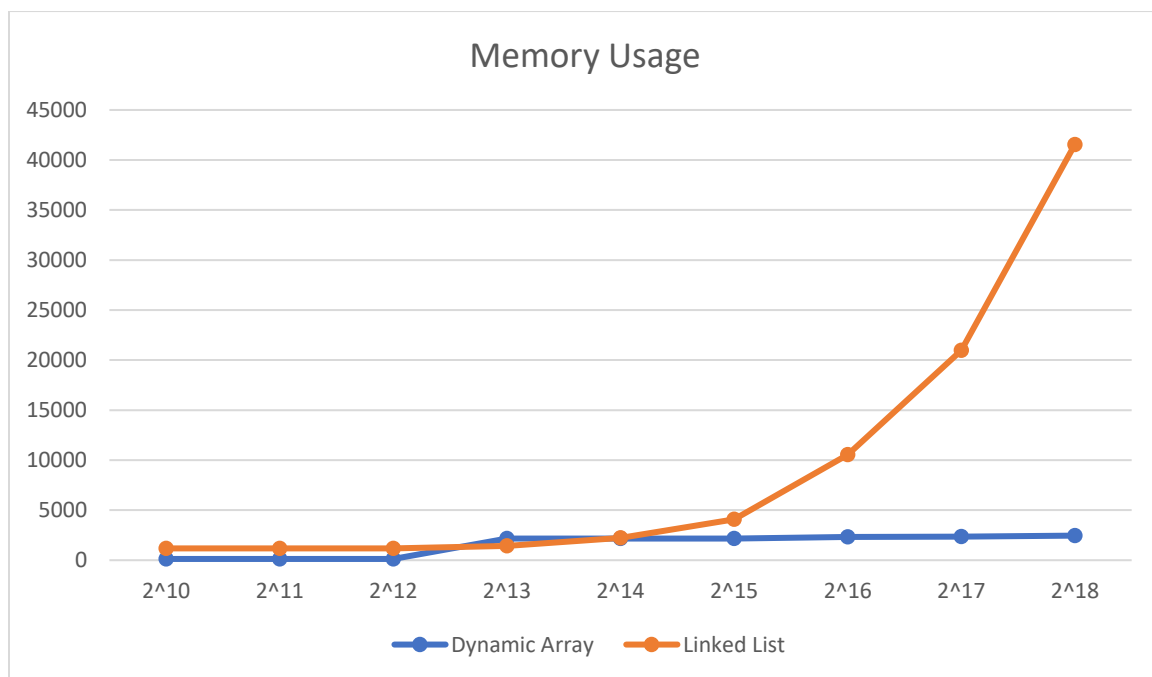
### Linked List Memory Usage and Run Times

Element counts n=	Memory Usage	Run Time in ms
2 <sup>10</sup>	1180KB	40ms
2 <sup>11</sup>	1180KB	140ms
2 <sup>12</sup>	1180KB	380ms
2 <sup>13</sup>	1436KB	1150ms
2 <sup>14</sup>	2228KB	5010ms
2 <sup>15</sup>	4076KB	17780ms
2 <sup>16</sup>	10540KB	88360ms
2 <sup>17</sup>	20964KB	290460ms
2 <sup>18</sup>	41548KB	1.19134e+06ms

## Running Time Comparison of Dynamic Array and Linked List



## Memory Usage Comparison of Dynamic Array and Linked List



1. Which of the implementations uses more memory? Explain why.

The Linked List implementation uses more memory by a good margin, especially at larger values of  $n$ . This is because a linked list must use more memory compared to an array to store pointers and references to pointers for previous and next nodes. A dynamic array does not need to store these pointers and thus uses less memory overall.

2. Which of the implementations is the fastest? Explain why.

Based on my research, I found that both the dynamic array and the linked list had relatively the same run time. In fact, in some instances the dynamic array was faster and in others the linked list was faster. They averaged out to be about the same run time overall.

My educated guess about their run times being similar would be that because both involve iterating through data, they both have similar run times. The dynamic array has to iterate through the array when an element is added or removed, move each element to add or close the gap, and search the array in other functions as well. In a linked list, adding and removing an element can be done more quickly, but searching the linked list is a slower process due to the extra pointer nodes that must be changed.

3. Would you expect anything to change if the loop performed `remove()` instead of `contains()`? If so, why?

Yes, if the loop performed `remove()` instead of `contains()`, I would expect the linked list run time to become much quicker than the dynamic array. In this function the dynamic array would have to loop through the entire array (at worst case) to find the element. The linked list however would be able to perform the action much more quickly, speeding up its overall run time ahead of the dynamic array implementation.