* 1. The complexity of pushright(x) is O(n)

The complexity of popleft() is O(1)

The complexity of element\_at(i) is O(i) which in general is O(n)

* 1. This does not satisfy the fundamental inequality of amortized cost. The aggregate true cost of n calls to element\_at(n) is kn^2 which is O(n^2) for some constant k. However the proposed aggregate amortized cost for the same actions would be O(n). Therefore there always exists some n for which the true aggregate cost is greater than the proposed aggregate amortized cost, and so the proposal must be incorrect.



* 1. The true cost of element\_at(i) is O(1). The true cost of popleft() is O(1). For pushright(x), the cost of copying the array into a larger one is kn where n is the initial size of the array and k is a constant. After n calls to pushright, the total cost is n (for the inserting) + k(1+2+4+8+….+2^(floor(log\_2(n-1))) which is strictly less than k(2n-3)+n which is O(n). Therefore the amortized cost of a single call to pushright(x) is O(1). Therefore all of the function calls have amortized cost O(1).