

---

**1: Big Oh**

---

- (a)  $O(n^2)$
- (b)  $O(\log n)$
- (c)  $O(1/n)$
- (d)  $O(1)$
- (e)  $O(\log n)$

---

**2: Removing Duplicates**

---

**Algorithm:** : First, sort the array A (using merge sort), we will call this sorted array C which is of size  $n$ . Next, iterate through C, in order, while keeping track of the previous element. If the current element is not equal to the previous element (or it's the first element), add it to B. Else, continue to the next element.

**Correctness:** : We can observe that every distinct element in C is included in B. Since A and C contain the same elements, we have correctness.

**Running time:** : The step of sorting A with merge sort take  $O(\log n)$  time and the step of iterating B takes  $O(n)$  time. Thus the overall running time of the algorithm is  $O(\log n)$

---

**3: Square vs Multiply**

---

---

**4: Basic Probability**

---

- (a)
- (b)

---

**5: Array Sums**

---

**Algorithm:** :

**Correctness:** :

**Running time:** :