

Runtime and pseudocode

Note: Assume lists are indexed starting at 1 and all loop endpoints are inclusive. Arithmetic, assignments, and accessing a list entry each take constant time.

1. Consider the following algorithm, which takes a list A of $n \geq 1$ numbers and produces a new list B .

```
Algorithm Mystery(A, n):  
    Create a list B of length n  
    for i = 1 to n:  
        total = 0  
        for j = 1 to i:  
            total = total + A[j]  
        B[i] = total  
    return B
```

- (a) Trace the algorithm on $A = [3, 1, 4, 2]$. Record the value of `total` at the end of each outer-loop iteration and give the returned list B .
- (b) Describe in one sentence what $B[i]$ represents for a general input list A .
- (c) How many times does the statement `total = total + A[j]` execute? Write your answer as a sum, then simplify it to a formula in n .
- (d) Give the tightest big- O bound you can for the running time. Explain your answer using your operation count.
- (e) For large n , approximately how would doubling the input length affect the running time?