

1) (5 pts) ANL (Algorithm Analysis)

Given an array, `vals`, of size `n`, one can determine the sum of the elements in the array from index `i` through index `j` ($i \leq j$), inclusive, simply by running a for loop through the elements:

```
int sum = 0;
for (int z=i; z<=j; z++)
    sum += vals[z];
```

This type of sum is known as a contiguous subsequence sum.

Note: There are more efficient ways to do this if many sums of this format need to be determined, but for the purposes of this problem, assume that this is how such a sum is determined.

(a) (3 pts) What is the worst case run time of answering `q` questions about contiguous subsequence sums on an array of size `n`? Express your answer in Big-Oh notation, in terms of both `n` and `q`. Give a brief justification for your answer.

The worst case query is when $i = 0$ and $j = n-1$. Answering this one query takes $O(n)$ time, since we loop through each element of the array. If we were to answer `q` of these queries, it would take $O(nq)$ time.

Grading: 1 pt for the answer, 2 pts for the reason. A specific worst case scenario doesn't need to be given, but simply referencing that a large query may take time proportional to the array size is good enough.

(b) (2 pts) What is the best case run time of answering `q` questions about contiguous subsequence sums on an array of size `n`? Express your answer in Big-Oh notation, in terms of both `n` and `q`. Give a brief justification for your answer.

The best case query is where $i = j$. Running such a query takes $O(1)$ time. Running `q` of these best case queries takes $O(q)$ time. Note that in this case, the answer does not depend on `n` at all.

Grading: 1 pt for the answer, 1 pt for the reason