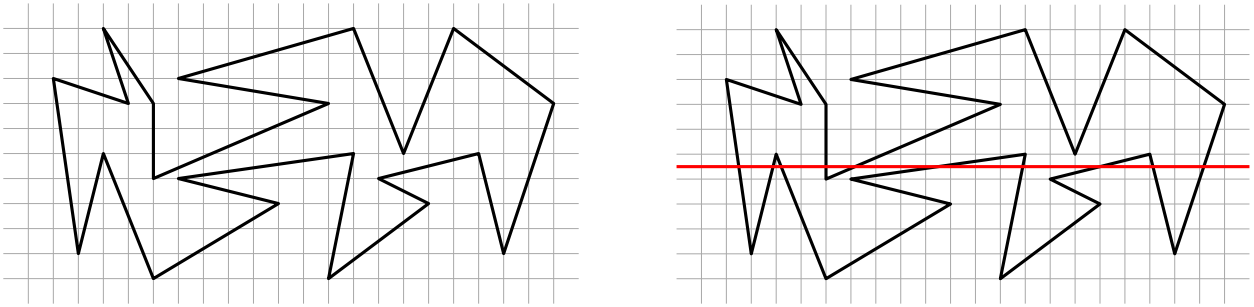
Algorithms (CSCI-261-03, CSCI-264), Fall 2023/24
Homework 2, due Friday, Sept 29, 2023, 11:59pm

Problem 1

Given is a polygon with n vertices $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ listed in clockwise order. You want to find a horizontal line that crosses the polygon the largest number of times. The line cannot pass through any of the vertices. Design an $O(n \log n)$ algorithm which finds the largest number of crossings a horizontal line (which does not pass through any of the n vertices) can have with the polygon.

Example: Suppose the polygon on the left is on the input. The red horizontal line, shown on the right, crosses the polygon 10 times. This is the largest possible number of crossings and hence the output is 10.



Problem 2

Consider the divide-and-conquer algorithm shown below; the algorithm accesses a global array A of integers.

```
WHATDOIDO(integer left, integer right):
    if left==right:
        if A[left]<0 return (0, 0, 0, A[left])
        else return (A[left], A[left], A[left], A[left])
    if left<right:
        m = (left+right)/2 (rounded down)
        (lmaxsum, llmaxsum, lrmaxsum, lsum) = WHATDOIDO(left, m)
        (rmaxsum, rlmaxsum, rrmaxsum, rsum) = WHATDOIDO(m+1, right)
        maxsum = max{lmaxsum, rmaxsum, lrmaxsum+rlmaxsum}
        leftalignedmaxsum = max{llmaxsum, lsum+rlmaxsum}
        rightalignedmaxsum = max{rrmaxsum, lrmaxsum+rsum}
        sum = lsum+rsum
        return (maxsum, leftalignedmaxsum, rightalignedmaxsum, sum)
```

Before running the algorithm, we ask the user to enter n integers that we store in the array A. Then we run `WHATDOIDO(1,n)` and out of the four returned values, we output the first.

a) State the recurrence for $T(n)$ that captures the running time of the algorithm as closely as possible.

b) Use the "unrolling the recurrence" or the mathematical induction to find a tight bound on $T(n)$.

c) What does the algorithm do?

- For an input in A and integers `left` and `right`, succinctly describe the meaning of the return variables `maxsum`, `leftalignedmaxsum`, `rightalignedmaxsum`, and `sum`.
- Recall that we run `WHATDOIDO(1,n)` and output the first of the four returned values. Succinctly describe the meaning of this output — what does it correspond to in terms of the input?

Problem 3

In the problem of counting inversions, you are given a permutation $a_1, a_2, \ldots, a_n$ of numbers $1, 2, \ldots, n$ and the goal is to count the number of pairs i, j, where $i < j$ and $a_i > a_j$. In this homework problem, you are given a sequence of n numbers $b_1, b_2, \ldots, b_n$ and your task is to compute the "weighted count" of inversions defined as follows: An inversion is a pair of indices i, j where $i < j$ and $b_i > b_j$. An i, j inversion has weight $b_i + b_j$ and the weighted count for the input sequence is the sum of the weights of all its inversions.

For example, for $n = 5$ and input sequence $7, 3, 8, 1, 5$, we have the following inversions weights: $7 + 3 = 10$, $7 + 1 = 8$, $7 + 5 = 12$, $3 + 1 = 4$, $8 + 1 = 9$, and $8 + 5 = 13$. The overall weighted count is: $10 + 8 + 12 + 4 + 9 + 13 = 56$.

Design an $O(n \log n)$ algorithm which computes the weighted count of inversions for a given input sequence.

Problem 4

For each of the following recurrences, use the Master theorem to express $T(n)$ as a Theta of a simple function. State what the corresponding values of a, b, and $f(n)$ are and how you determined which case of the theorem applies. Do not worry about the base case or rounding.

1. $T(n) = 3T(n/9) + \sqrt{n}$
2. $T(n) = 4T(n/2) + n^3$
3. $T(n) = 5T(n/4) + n \log n$

Problem 5 (CSCI-264 only)

In the Convex Hull problem, we are given n points with coordinates $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ and the task is to compute the smallest convex polygon which includes all points. The coordinates are arbitrary real numbers and the polygon should be output as a sequence of its vertices, in clockwise order (starting at any of the vertices). Prove that every Convex Hull algorithm has $\Omega(n \log n)$ running time.