University of Nevada, Las Vegas Computer Science 477/677 Spring 2019

Assignment 5: Due Wednesday May 1, 2019

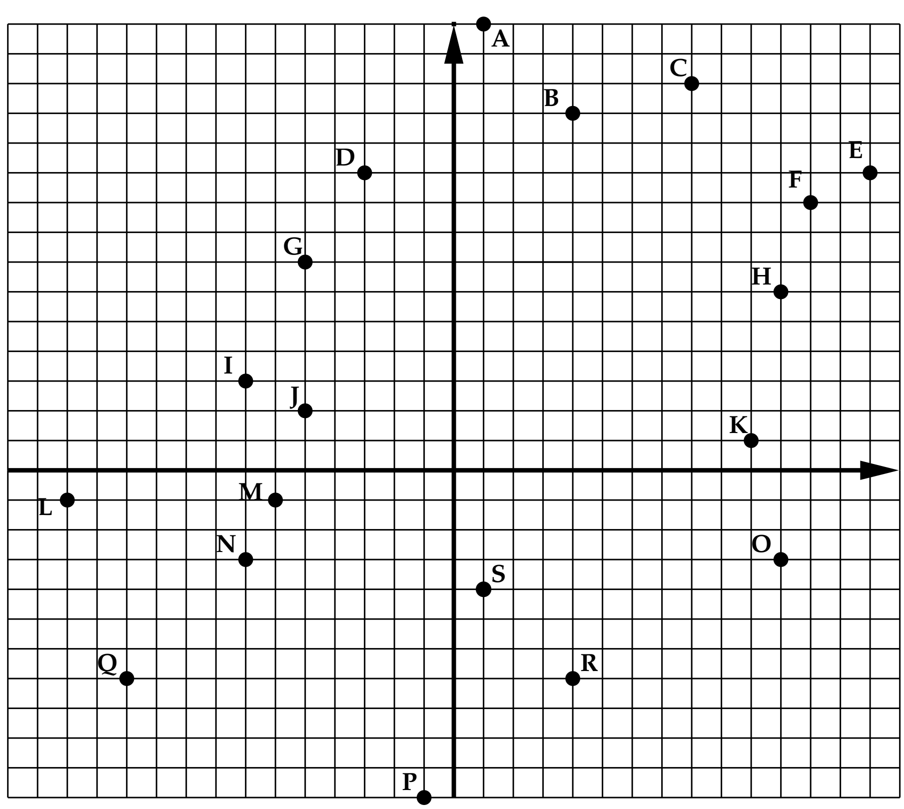
Name:____

You are permitted to work in groups, get help from others, read books, and use the internet. But the handwriting on this document must be your own. Print out the document, staple, and fill in the answers. You may attach extra sheets. Turn in the pages to the graduate assistant at the beginning of class, May 1.

1. Find the longest common subsequence of the two sequences a, c, b, d, c, a, b, d and a, b, d, c, b by filling in the dynamic programming table.

		a	c	b	d	c	a	b	d
	0								
a									
b									
d									
c									
b									

2. Compute the convex hull of the indicated points in the graph below, using Graham Scan.



Range Query Structures

Suppose we are given an array $A[0] \ldots A[n-1]$ of size n. A *range query* is a question of the type, "What is the sum of all the values of A on the interval $[i, j]$? We will write Query$[A, i, j] = \sum_{k=i}^{j} A[k]$ Of course, you can simply compute that sum in $O(n)$ time. But if there will be many queries, you can save time by precomputing some of the values. For example, Query$[A, 2, 19]$ can be evaluated much faster if you have already computed and saved the values of Query$[A, 2, 6]$ and Query$[A, 7, 19]$: simply add those values.

We do not allow subtraction in our calculations, since the entries of A could be members of an arbitrary semigroup, that is, "+" could be overloaded to mean any associative operation, which may not have inverses, such as maximum, minimum, or matrix multiplication.

$n = 4$ and A has the values 9,7,3,6. There are 10 possible queries, and we can store them all, as in (a) below. In that case any query can be answered with a single fetch. On the other had, if we are willing to do two fetches, we need store only six values. For example Query$[0, 2]$ =Query$[0, 1]$+Query$[2, 2] = 16 + 3 = 19$.

	0	**1**	**2**	**3**
0	9	16	19	25
1		7	10	16
2			3	9
3				6

(a)

	0	**1**	**2**	**3**
0	9	16		
1		7		
2			3	9
3				6

(b)

Now suppose that $n = 16$, and the values of A are

$$9, 7, 3, 6, 9, 11, 3, 10, 12, 5, 7, 13, 3, 4, 1, 3$$

If we are willing to do only one fetch per query, we must store the values of all 120 possible queries. If we are willing to do any number of fetches, we need only store the values of Query$[i, i]$, that is, simply the values of the array, as shown below.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	9															
1		7														
2			3													
3				6												
4					9											
5						11										
6							3									
7								10								
8									12							
9										5						
10											7					
11												13				
12													3			
13														4		
14															1	
15																3

3. If we are willing to do up to two fetches per query, we can get away with storing only 48 values. Fill in 48 entries in in the figure below with those values. Leave the remaining places blank.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																

In order to find the value of any query with at most two fetches for an array of size n, we need to store. $\Omega(n \log n)$ values

4. If we are willing to do up to three fetches per query, we can get away with storing only 34 values, as shown below. How would you compute Query[3, 14]?

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	9				34				70				98			
1		7	10	16												
2			3	9												
3				6												
4					9				45				73			
5						11	14	24								
6							3	13								
7								10								
8									12				40			
9										5	12	25				
10											7	20				
11												13				
12													3			
13														4	5	8
14															1	4
15																3

In order to find the value of any query with at most three fetches for an array of size n, we need to store $\Omega(n \log \log n)$ values. For up to four fetches, we need to store $\Omega(n \log^* n)$ values.

In order to find the value of any query with at most $\alpha(n)$ fetches, we need to store $\Omega(n\alpha(n))$ values, where α is the inverse of the Ackermann function.

Scratch: