1. (20%) Let A, B be two arrays of length n where $A[1] < A[2] < \cdots < A[n]$ and $B[1] > B[2] > \cdots > B[n]$.

   (a) Give an $O(1)$-time algorithm that can find the index k so that $A[k] - B[k]$ is minimized.

   (b) Prove that any algorithm that can find the index k so that $A[k] + B[k]$ is minimized requires $\Omega(n)$ time.

2. (30%) Give asymptotic upper bounds for the following $T(n)$, and justify your answers. You may assume that $T(1) = 1$.

   (a) $T(n) = T(\lceil n/3 \rceil) + T(\lceil n/4 \rceil) + O(n)$.

   (b) $T(n) = T(\lceil n/3 \rceil + 5) + T(\lceil n/4 \rceil + 7) + O(n)$.

   (c) $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + O(\log n)$.

3. (10%) Let Q be the convex polygon illustrated in Figure 1. Answer the following questions by calculating cross products as we did in the lecture.

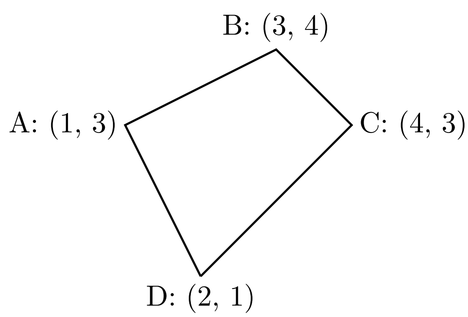

Figure 1: The convex polygon Q.

   (a) Let $p = (4, 2)$. Is $p \in Q$? By $p \in Q$, we mean that p is on the boundary of Q or in its interior. If not, what is the convex hull of $Q \cup \{p\}$?

   (b) Let $p = (3, 2)$. Is $p \in Q$? If not, what is the convex hull of $Q \cup \{p\}$?

4. (15%) Given an n by n matrix $A \in \mathbb{R}^{n\times n}$. Finding the *monotonic*[1] path from $A[1][1]$ to $A[n][n]$ so that the sum of values on the cells visited by the path is minimized, i.e. the shortest monotonic path.

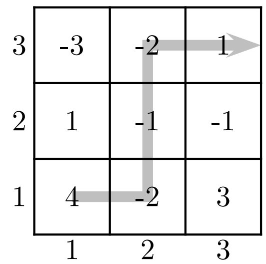

Figure 2: An illustration of the monotonic path.

   (a) Devise a *polynomial-time* algorithm that can output the length of the shortest monotonic path. By polynomial-time algorithms, we mean those algorithms whose running time is $O(n^c)$ for some constant $c \geq 0$. Give the pseudocode of your algorithm.

   (b) Devise a polynomial-time algorithm that can output the cells visited by the shortest monotonic path.

   (c) Explain why your algorithms are correct and analyze their running time.

5. (10%) Give an array A of n real numbers. Devise an $O(n \log n)$-time algorithm that can output the longest *bitonic* subsequence S of A. We say a subsequence S is bitonic if there exists an index k so that $S[i] < S[j]$ for every $i < j$, $j \leq k$ and $S[i] > S[j]$ for every $i < j$, $i \geq k$.

   (a) Give the pseudocode of your algorithm.

   (b) Explain why your algorithm is correct and analyze its running time.

6. (15%) Give a bag and n stones where the i-th stone has weight w_i and value v_i. We would like to place some of the n stones into the bag so that the total value of the selected stones is maximized and the total weight of the selected stones does not exceed m, a given parameter.

[1] By monotonic path, we mean a path that goes only upward and rightward.

(a) Devise an $O(mn)$-time algorithm that can answer whether there exists a subset of the n stones whose total weight is k, for any $k \leq m$. Give the pseudocode of your algorithm.

(b) Devise an $O(mn)$-time algorithm that can calculate the maximum value of the stones that you can pack into the bag. Give the pseudocode of your algorithm.

(c) Explain why your algorithms are correct.