HW 2 (due Wednesday, at noon, September 12, 2018)

CS 473: Algorithms, Fall 2018 Version: **1.01**

Submission guidelines and policies as in homework 1.

---

1 (100 PTS.) Weighted discrepancy on the line.

You are given a set $W = \{w_1, \ldots, w_n\}$ of n numbers, where $w_i \in [\![U]\!] = \{1, \ldots, U\}$, for all i, where U is some positive integer number. The purpose of this question is to partition the numbers into two sets that are as balanced as possible. In particular, let $s_i \in \{-1, +1\}$ be an indicator to which of the two sets w_i is assigned to, for all i. An ***assignment*** as such, is a vector $S \in \{-1, +1\}^n$, where the two resulting sets in the partition are $L(S) = \{w_i \mid s_i = -1\}$ and $R(S) = \{w_i \mid s_i = 1\}$.

1.A. (50 PTS.) Given an assignment $S = (s_1, \ldots, s_n)$, the ***balance*** of the partition is

$$B(S) = \left|\sum_{i=1}^{n} s_i w_i\right| = \left|\sum_{x\in L(S)} x - \sum_{y\in R(S)} y\right|$$

Provide an algorithm, as fast as possible, that computes (and outputs) the assignment that realizes

$$\mathcal{B}(w_1, \ldots, w_n) = \min_{S\in\{-1,+1\}^n} B(S).$$

The running time would depend on both n and U. What is the running time of your algorithm?

How much space does your algorithm need? The smaller, the better.

Can you improve the space, if you only need to compute the value of the optimal solution (not the assignment itself).

Pseudo-code for the algorithm is highly recommended for this part.

1.B. (50 PTS.) Given a parameter k, let $\mathcal{S}(k, n)$ be the set of all vectors in $\{-1, +1\}^n$ that have at most k coordinates with -1 in them. Provide an algorithm, as fast as possible, that computes the assignment that realizes $\min_{S\in\mathcal{S}(k,n)} B(S)$.

1.C. (Not for submission, but you can think about it.)

Given integers $i \leq j$, and an assignment S, let $B(S, i, j) = \left|\sum_{\ell=i}^{j} s_i w_i\right|$ be the balance of the subsequence $w_i, \ldots, w_j$ for the given assignment. The ***discrepancy*** of the input, given an assignment S, is $D(S) = \max_{i\leq j} B(S, i, j)$. The discrepancy of the input $\mathcal{D}(w_1, \ldots, w_n) = \min_{S\in\{-1,+1\}^n} D(S)$. Provide an algorithm, as fast as possible, that computes the assignment that realizes $\mathcal{D}(w_1, \ldots, w_n)$.

2 (100 PTS.) Good path.

(To solve this problem, you might want to revisit topological ordering, and how to compute it in linear time.)

Consider a DAG G with n vertices and m edges that represents courses available. Each vertex v of G corresponds to a course, with value α_v (which might be negative, if it is a useless course). A vertex v is ***useful*** if $\alpha_v > 0$.

2.A. (20 PTS.) Show an algorithm that in linear time computes all the vertices that can reach a sink of G via a path that goes through at least one useful vertex.

2.B. (20 PTS.) Describe an algorithm that, in linear time, computes all the vertices that can reach a sink of G via a path that goes through at least β useful vertices, where β is a prespecified parameter.

2.C. (30 PTS.) Show an algorithm, as fast as possible, that computes for all the vertices v in G the most beneficial path from v to any sink of G. The ***benefit*** of a path is the total sum of the values of vertices along the path.

2.D. (30 PTS.) Using the above, describe how to compute, in linear time, a path that visits all the vertices of G if such a path exists.

3 (100 PTS.) Frequent flier miles.

You are give an directed graph G with n vertices and m edges, positive prices on the edges of G, a parameter $k \leq n$, and two vertices s and t. (That is, for every edge $(u, v) \in \mathsf{E}(\mathsf{G})$, there is an associated price $p\big((u, v)\big) > 0$.) Describe an algorithm, as fast as possible, that computes the shortest path from s to t, where the path is allowed to not pay for (at most) k edges used by the path. What is the running time of your algorithm? (You can use the Dijkstra algorithm as a subroutine if you want/need to.)