Task: Rice Hub

Proposed by: Christain Kauth

The key insight to solving this problem is the observation that for any K rice fields located at $r_0 \leq r_1 \leq \cdots \leq r_{K-1}$, the transportation cost from all these K fields is minimized by placing the rice hub at a median. For example, when $K = 1$, the hub should be at r_0, and when $K = 2$, placing it between r_0 and r_1 is optimal. In this problem, we will place the rice hub at $r_{\lfloor K/2 \rfloor}$ for simplicity. Following this observation, we denote a solution by a sequence $S \subseteq \langle r_0, \ldots, r_{R-1} \rangle$ and let $|S|$ denote the length of S, which is the solution's value (the number of rice fields whose rice will be transported to the hub). The cost of S is $\text{cost}(S) = \sum_{r_j \in S} |r_j - h(S)|$, where $h(S)$ is the $\lfloor |S|/2 \rfloor$-th element of S.

1 An $O(R^3)$ solution

Armed with this, we can solve the task by a guess-and-verify algorithm. We try all possible lengths of S (ranging between 1 and R). Next observe that in any optimal solution S^*, the rice fields involved must be contiguous; that is, S^* is necessarily $\langle r_s, r_{s+1}, \ldots, r_t \rangle$ for some $0 \leq s \leq t \leq R - 1$. Therefore, there are $R - K + 1$ solutions of length K. For each choice of S, we compute $h(S)$ and the transportation cost in $O(|S|)$ time and check if it is within the budget B. This leads to an $O(R^3)$ algorithm, which suffices to solve subtask 2.

2 An $O(R^2)$ solution

To improve it to $O(R^2)$, we will speed up the computation of $\text{cost}(S)$. Notice that we are only dealing with consecutive rice fields. Thus, for each S, the cost $\text{cost}(S)$ can be computed in $O(1)$ after precomputing certain prefix sums. Specifically, let $T[i]$ be the sum of all coordinates to the left of rice field i, i.e., $T[0] = 0$ and $T[i] = \sum_{j=0}^{i-1} X[j]$. Then, if $S = \langle r_s, \ldots, r_t \rangle$, $\text{cost}(S)$ is given by $(p - s)r_p - (T[p] - T[s]) + (T[t+1] - T[p+1]) - (t - p)r_p$, where $p = \lfloor (s + t)/2 \rfloor$. This $O(R^2)$ algorithm suffices to solve subtask 3.

3 An $O(R \log R)$ solution

Applying a binary search to find the right length in place of a linear search improves the running time to $O(R \log R)$ and suffices to solve all subtasks.

4 An $O(R)$ solution

We replace binary search with a variant of linear search carefully designed to take advantage of the feedback obtained each time we examine a combination of rice fields. In particular, imagine adding in the rice fields one by one. In iteration i, we add r_i and find (1) S_i^*, the best solution that uses only (a subsequence of) the first i rice fields (i.e., $S_i^* \subseteq \langle r_0, \ldots, r_i - 1 \rangle$), and (2) S_i, the best solution that uses only (a subsequence of) the first i rice fields and *contains* $r_i - 1$. This can be computed inductively as follows. As a base case, when $i = 0$, both S_i and S_i^* are just $\langle r_0 \rangle$ and cost 0, which is within the budget $B \geq 0$. For the inductive case, assume that S_i^* and S_i are known. Now consider that S_{i+1} is S_i appended with r_i, denoted by $S_i \cdot r_i$, if the cost $\text{cost}(S_i \cdot r_i)$ is at most B, or otherwise it is the longest prefix of $S_i \cdot r_i$ that costs at most B. Futhermore, S_{i+1}^* is the better of S_i^* and S_{i+1}. To implement this, we represent each S_i by its starting point s and ending point t; thus, each iteration involves incrementing t and possibly s, but s is always at most t. Since $\text{cost}(\langle r_s, \ldots, r_t \rangle)$ takes $O(1)$ to compute, the running time of this algorithm is $O(R)$ and suffices to solve all subtasks.