(3 points)

1. In L_1 matching, instead of counting mismatches we look at $o[i] = \sum_{j=1}^{m} |t[i+j-1]-p[j]|$. Design a deterministic $\tilde{\mathcal{O}}(n/\varepsilon^2)$ time algorithm that approximates this value, for all $i = 1, 2, \ldots, n-m+1$, with multiplicative error $1+\varepsilon$.

2. Design an $\mathcal{O}(nk)$ time algorithm that given a pattern $p[1..m]$, a text $t[1..n]$ and a parameter k, finds a substring $t[i..j]$ such that the edit distance between $t[i..j]$ and $p[1..m]$ is at most k.