CS170–Fall 2022 — Homework 6 Solutions

Deming Chen cdm@pku.edu.cn

Last Modified: January 8, 2023

Collaborators: NONE

2 Egg Drop

(a) $f(1, k) = 1$, for $k \geqslant 1$. We just need to drop an egg from the first floor.

$f(0, k) = 0$, because we already know an egg will always break.

$f(n, 1) = n$, because we have no more egg, and thus have to drop the egg from the 1st floor, 2nd floor, $\cdots$, n-th floor in turn.

$f(n, 0) = +\infty$, because we will never know what l is.

(b) Suppose the first drop is from the jth floor. If the egg breaks, we know that $l \leqslant j-1$. Afterward the minimum number of drops is $f(j-1, k-1)$. If the egg does not break, we know that $l \geqslant j$, so minimum number remaining drops is $f(n-j, k)$. However, both cases may happen. So the worst case is $\max\{f(j-1, k-1), f(n-j, k)\}$. Therefore, a recurrence relation for $f(n, k)$ is

$$f(n, k) = 1 + \min_{1 \leqslant j \leqslant n} \max\{f(j-1, k-1), f(n-j, k)\}. \tag{1}$$

3 Paper Cutting

(a) For convenience, we use the coordinates the left top corner and the right bottom corner to denote a rectangle in the original large rectangle $\{(x_1, y_1), (x_2, y_2)\}$, where $(x_1 \leqslant x_2, y_1 \leqslant y_2)$. The subproblem is to find the minimum number of cuts needed to separate the stains out in a inner rectangle $f(x_1, y_1, x_2, y_2)$. What we want is $f(1, 1, m, n)$

(b) Base case: if a rectangle has no strain, or is full of strains, then $f(x_1, y_1, x_2, y_2) = 0$.

Recurrence relation: the first cut can be horizontal or vertical:

$$f(x_1, y_1, x_2, y_2) = 1 + \begin{cases} \min\limits_{x_1 \leqslant j \leqslant x_2 - 1} [f(x_1, y_1, j, y_2) + f(j+1, y_1, x_2, y_2)], \text{ if } y_1 = y_2 \\ \min\limits_{y_1 \leqslant j \leqslant y_2 - 1} [f(x_1, y_1, x_2, j) + f(x_1, j+1, x_2, y_2)], \text{ if } x_1 = x_2 \\ \min\{f_H, f_V\}, \text{if } x_1 < x_2 \text{ and } y_1 < y_2, \end{cases} \tag{2}$$

where

$$f_H = \min_{x_1 \leqslant j \leqslant x_2 - 1} [f(x_1, y_1, j, y_2) + f(j+1, y_1, x_2, y_2)],$$

and

$$f_V = \min_{y_1 \leqslant j \leqslant y_2 - 1} [f(x_1, y_1, x_2, j) + f(x_1, j+1, x_2, y_2)].$$

(c) The cost of runtime is $\mathcal{O}(m^3 n^3)$. x_1 and x_2 have $(m+1)m/2$ choices. y_1 and y_2 have $(n+1)n/2$ choices. So, there are $\mathcal{O}(m^2 n^2)$ sub-problems. For one sub-problem, we need $\mathcal{O}(mn)$ time to check whether it is a base case. If not, calculus the recurrence relation takes time $\mathcal{O}(m+n)$. Therefore, time complexity of solving the above mentioned recurrence is $\mathcal{O}(m^3 n^3)$.