

Pro. 15-6

(爲了方便解說，將 $n * n$ 棋盤的每一個格子都編號，從最下面一行從左到右依序標爲 $0..n-1$ ，第二行左到右爲 $n..2n-1$ ，以此類推。)

令 $F(x)$ 是 checker 從最下面那行移動到格子 x ，能賺到最多錢的數目。利用下面的 DP recurrence：

$$F(x) = \max \begin{cases} F(x-n) + p(x-n, x) & , \text{if } x \geq n \\ F(x-n-1) + p(x-n-1, x) & , \text{if } x \geq n \text{ and } (x \bmod n) \neq 0 \\ F(x-n+1) + p(x-n+1, x) & , \text{if } x \geq n \text{ and } (x \bmod n) \neq n-1 \\ 0 & , \text{if } x < n \end{cases}$$

答案就是最上面一行的最多錢的格子： $\max\{F(n^2-n), \dots, F(n^2-1)\}$

每個格子 x ，能夠從三個方向來：右下格($x-n-1$)、下面那格($x-n$)、左下格($x-n+1$)這三格其中之一走過來，所以每次都去找那三格的最好走法來走到我這一格。當往回問到最下面那一行時，就不能再往下問了，所以當 $x < n$ 時，會停下來。再多紀錄一個 table，拿來作 backtrack，紀錄我每次都是從下面三格的那一格過來的。

Bottom-up 可以依序 $1 \sim n^2$ 填表。

Time complexity: 每一格只需要 $O(1)$ 的時間就能知道，總共有 n^2 格，所以總共需要 $O(n^2)$ 。