

P1

Claim:

Given the pictured graph with $2n$ vertices, the following dynamic program runs in polynomial time and outputs the number of independent sets of the graph.

Algorithm:

Input: The number of pairs in the pictured graph.

Output: The number of independent sets in the graph.

IndSet(n):

 Let V be an empty list

$V[0] = 1$

$V[1] = 3$

 For $2 \leq i \leq n$ **do**

$V[i] = 2V[i - 1] + V[i - 2]$

EndFor

 Return $V[n]$

Claim:

The algorithm runs in polynomial time.

Proof:

That the algorithm terminates is obvious given that each element in the array is accessed once and only once. The number of steps to reach each point is cached as an array element as it's found, so during loop iteration, previously found step costs are accessed in constant time. Therefore the dominant runtime factor is the loop. Thus, the algorithm terminates in polynomial time. \square

Claim:

The algorithm correctly outputs the number of independent sets.

Proof:

Let $\text{OPT}(n, f)$ denote number of independent sets in the pictured graph, where n is the number of columns in the graph and f is a flag indicating whether the vertex in the independent set in the n th column is in the top row T , in the bottom row B , or in neither row N . Let $\text{OPT}(n)$, with only one parameter, denote the sum of $\text{OPT}(n, f)$ for each value of f .

Case 1: If $n=0$ then there is only the empty set.

$$\begin{aligned}\text{OPT}(0) &= \text{OPT}(0, T) + \text{OPT}(0, B) + \text{OPT}(0, N) \\ &= 0 + 0 + 1 \\ &= 1\end{aligned}$$

Case 2: If $n=1$ then there are three possible independent sets.

$$\begin{aligned}\text{OPT}(1) &= \text{OPT}(1, T) + \text{OPT}(1, B) + \text{OPT}(1, N) \\ &= 1 + 1 + 1 \\ &= 3\end{aligned}$$

Case 3: If $n>1$ then the total number of independent sets relies on previous solutions.

First, consider the values of the two parameter $\text{OPT}(n, f)$ at n .

If there is a vertex in the independent set in the top row of column n , then, to remain an independent set, any vertex in column $n-1$ must be in the bottom row or not be in the set at all.

$$\text{OPT}(n, T) = \text{OPT}(n-1, B) + \text{OPT}(n-1, N)$$

Similar to $\text{OPT}(n, T)$, if there is a vertex in the independent set in the bottom row of column n , then, to remain an independent set, any vertex in column $n-1$ must be in the top row or not be in the set at all.

$$\text{OPT}(n, B) = \text{OPT}(n-1, T) + \text{OPT}(n-1, N)$$

Note that $\text{OPT}(n, B)$ and $\text{OPT}(n, T)$ have the same number of solutions since one is a mirror reflection of the other.

If there is no vertex in the independent set in column n , then $\text{OPT}(n, N)$ is the sum of previous columns' solutions.

$$\begin{aligned}\text{OPT}(n, N) &= \text{OPT}(n-1, T) + \text{OPT}(n-1, B) + \text{OPT}(n-1, N) \\ &= \text{OPT}(n-1)\end{aligned}$$

Now, combining these two parameter solutions we find the one parameter solution for $\text{OPT}(n)$.

$$\begin{aligned}
\text{OPT}(n) &= \text{OPT}(n, T) + \text{OPT}(n, B) + \text{OPT}(n, N) \\
&= \text{OPT}(n-1, B) + \text{OPT}(n-1, N) + \text{OPT}(n-1, T) + \text{OPT}(n-1, N) + \text{OPT}(n-1) \\
&= \text{OPT}(n-1) + \text{OPT}(n-1) + \text{OPT}(n-1, N) \\
&= 2 * \text{OPT}(n-1) + \text{OPT}(n-2, N) + \text{OPT}(n-2, T) + \text{OPT}(n-2, N) \\
&= 2 * \text{OPT}(n-1) + \text{OPT}(n-2)
\end{aligned}$$

Thus, our recurrence relation for $\text{OPT}(n)$:

$$\text{OPT}(n) = \begin{cases} 1 & n=0 \\ 3 & n=1 \\ 2 * \text{OPT}(n-1) + \text{OPT}(n-2) & n>1 \end{cases}$$

□