

ADDA



## 1. Introducción

Problema de Pociones	
Técnica	Programación Dinámica
Tamaño	<code>despensa.size() - i</code>
Propiedades Compartidas	<ul style="list-style-type: none"> <li>List&lt;Pocion&gt; <b>despensa</b>: Guarda todas las pociones</li> <li>Integer <b>nIni</b>: nivel inicial oponente</li> <li>TipoPersonaje <b>tp</b>: dirá si es nigromante o no</li> </ul>
Propiedades Individuales	<ul style="list-style-type: none"> <li>Integer <b>i</b> en <math>[0, \text{despensa.size}]</math>: posición actual de la lista</li> <li>Integer <b>nAct</b>: nivel actual</li> <li>Integer <b>cAcu</b>: coste acumulado</li> </ul>
Solución	<b>s = (as, ct, dt)</b> ; siendo: <ul style="list-style-type: none"> <li>as: MultiSet de alternativas seleccionadas en el paso</li> <li>ct: coste total</li> <li>dt: daño total</li> </ul>
Objetivo	Encontrar "s" tal que "dt >= nIni" y "ct" tenga el menor valor posible
Solución Parcial	<b>Sp = (a, ct)</b>
Alternativas	Si "g": $A_{cAcu, i, despensa} = \{0\}$ ; Si "!g": $A_{cAcu, i, despensa} = \{a:k...0\}$ ; $g = \text{despensa.get}(i).\text{getTipo} == \text{"Otro"} \ \&\& \ tp == \text{"Nigromante"}$ $k = \min(\text{can}, nAct/\text{can})$ ; $\text{can} = \text{despensa.get}(i).\text{getCant}()$ ;
Instanciación	$pp(\text{despensa}, nAct, \text{tipo}) = ppg(\text{despensa}, nIni, \text{tipo}, 0, nIni, 0)$
Problema Generalizado	$ppg(X, p) = \begin{cases} (null, 0) & i > d.length() \text{ or } nAct \leq 0 \\ cA_{a \in A_{co, i, d}}(c(X, a, ppg(sp))) & \text{ecoc} \end{cases}$ $co = cAcu$ ; $d = \text{despensa}$ ; $p = (i, nAct, cAcu)$ $X = (\text{despensa}, nIni, \text{tipo})$ $sp = (i++, d.get(i).getDano() * a + nAct, d.get(i).getCoste() * a)$
Combina (c)	$c(p, a, ('a, n1), ('a, n2)) = (a, n1 + n2 + fi * fa * c - 1)$
selecciona Alternativa (sA)	$sA(X, a) = (a, cAcu + pociones.get(i).getCoste() * a)$
Solución reconstruida	<div>3</div> $rc(K, m) = \begin{cases} \perp & m_a(K) = \perp \\ \emptyset + (d.get(0), m_a(K)) & \text{caso base} \\ rc(K_1^{m_a(K)}, m) + (d.get(j), m_a(K)) & \text{caso recursivo} \end{cases}$