

ADDA

1. Introducción

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