

ADDA

1. Introducción

Problema de Pociones	
Técnica	Programación Dinámica
Tamaño	<code>despensa.size() - i</code>
Propiedades Compartidas	<ul style="list-style-type: none"> • List<Pocion>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 $[0, \text{despensa.size}]$: posición actual de la lista • Integer nAct: nivel actual • Integer cAcu: coste acumulado
Solución	s = (as, ct, dt) ; siendo: <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" \geq nIni"y "ct" tenga el menor valor posible
Solución Parcial	Sp = (a, ct)
Alternativas	$A_{cAcu, i, despensa} = \{a:k...0\}$; $k = \min(\text{can}, nAct/\text{can})$; $\text{can} = \text{despensa.get}(i).\text{getCant}()$; Falta filtro de si es nigromante o no
Instanciación	$pp(\text{despensa}, nAct, \text{tipo}) = ppg(\text{despensa}, nIni, \text{tipo}, 0, nIni, 0)$
Problema Generalizado	$ppg(X, p) = \begin{cases} (null, 0) & i > d.length() \text{ or } nAct \leq 0 \\ cA_{a \in A_{co, i, d}}(c(X, a, ppg(sp))) & \text{ecoc} \end{cases}$ $co = cAcu$; $d = despensa$; $p = (i, nAct, cAcu)$ $X = (despensa, nIni, \text{tipo})$ $sp = (\text{variaciones de } p)$
sA	$(X, a) = (a, cAcu + pociones.get(i).getCoste() * a)$
Solución reconstruida	?¿