

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	$s = (as, ct, dt)$; 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 >= nIni" y "ct" tenga el menor valor posible
Solución Parcial	Sp = (a, ct)
Alternativas	$A_{cAcu}, i, despensa = \{a:k...0\};$ $k = \min(can, nAct/can);$ $can = despensa.get(i).getCant();$ Falta filtro de si es nigromante o no
Instanciación	$pp(despensa, nAct, tipo) = ppg(despensa, nIni, tipo, 0, nIni, 0)$
Problema Generalizado	$ppg(X, p) = \begin{cases} (null, 0) & i > d.length() \text{ or } nAct <= 0 \\ cA_{a \in A_{co, i, d}}(c(X, a, ppg(sp))) & \text{ecoc} \end{cases}$ <p> $co=cAcu; d=despensa;$ $p=(i, nAct, cAcu)$ $X=(despensa, nIni, tipo)$ $sp=(i++, d.get(i).getDano() * a + nAct, d.get(i).getCoste() * a)$ </p>
sA	$(X, a)=(a, cAcu+pociones.get(i).getCoste()*a)$
Solución reconstruida	? $\hat{}$