

ADDA

1. Introducción

Problema de Pociones	
Técnica	Programación Dinámica
Tamaño	despensa.size() - i
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 • ¿¿Necesaria?? List<Integer>multiplicidades: Guarda todas las multiplicidades de las pociones
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: 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	ProbPociones(0, 0, 0)
Instanciación	ProbPociones(?i, ?i, ?i)
Problema Generalizado	$ppg(p) = \begin{cases} (p[1],) & \text{sin is even} \\ () & i > d.length() \text{ or } nAct \geq 0 \\ cA_{a \in A_{co, i, d}}(c(jejeje)) & \text{ecoc} \end{cases}$ co=cAcu; d=despensa; p=(i, nAct, costeAcumulado) Sp=(?i, ?i)
?i	?i
sA	?i
Solución reconstruida	?i