Vuelta atrás + voraces - Diciembre 2015.

Solución para tupla de tamaño variable.

Secuencia de decisiones: $\langle d_1, d_2, \ldots d_s \rangle$ donde s es el número total de movimientos realizados, $1 \leq s \leq N$. d_i indica el índice de la celda a la que se accede en el movimiento i-ésimo.

Reestricciones explícitas: $(\forall i)(d_i \in \{1, 2, \ldots N\} : 1 \leq i \leq s)$

Reestricciones implícitas: Una tupla no es factible si alguno de sus elementos es mayor o igual que el siguiente, por lo que ha de cumplir que:

$$(\forall i)(d_i > d_{i-1} : 2 \leq i \leq s)$$

Además, una tupla también será factible cuando todos sus capitales acumulados sean positivos:

$$(\forall k)\left(\underbrace{C}_{\text{capital inicial}} + \sum_{i=1}^{k} V[d_i]\right) > 0 \quad \text{con} \quad 1 \leq k \leq s$$

Función objetivo: maximizar $C + \sum_{i=1}^{k} V[d_i]$ y el óptimo se alcanza con el capital total acumulado es máximo.

Una tupla es solución si $ds = N$.

```
Procedimiento Backtracking-optima (D: datos-problema; k: entero; e/s x, x-mejor:
    tupla; e/s u-mejor: valor)
        preparar-recorrido-nivel-k;
        mientras ∃-hermano-nivel-k hacer:
            siguiente-hermano-nivel-k;
            opcion
                solucion(D,k,x) ∧ correcto(D,k,x): si valor(D,k,x) > u-mejor entonces
                                                       x-mejor = x;
                                                       u-mejor = valor(D,k,x);
                                                   fsi
                ~solucion(D,k,x) ∧ correcto(D,k,x):
                                   Backtracking-optima(D,k+1,x,x-mejor,u-mejor);
            fopción
        fmientras
    fprocedimiento

Funcion correcto(V[1..n]: vect ent.; C: entero; k: entero; x: tupla) retorna
        (b: booleano)
        var i: entero, correcto: booleano fvar
        correcto = verdadero;
        para i = 2 hasta k hacer
            si (x[i] <= x[i-1]) entonces correcto = falso fsi
        fpara
        para i = 1 hasta k hacer
            si valor(V,C,k,x) <= 0 entonces correcto = falso fsi
        fpara
        retorna correcto
    ffunción
```

```
Funcion valor ( V[1..n] : vector de enteros ; C : entero ; x : tupla ; k : entero) retorna
    (s : entero)
    var total : entero fvar
    total = C;
    para i = 1 hasta k hacer
            total += V[x[i]];
    fpara
    retorna total
ffuncion

Funcion tratar ( x : tupla ; N : entero)
    para i = 1 hasta N hacer
        imprimir x[i]
    fpara
ffunción

Soe ucion (x, k) : x[k] = N
Preparar - recorrido - nivel - k : x[k] = 0 ;
Siguiente - hermano - nivel - k : x[k] += 1;
∃ - hermano - nivel - k : x[k] < N
```