

Práctica 2. Programación dinámica

Federico Carrillo Chaves
federico.carrilloch@alum.uca.es
Teléfono: +34615158732
NIF: 32095180Z

8 de noviembre de 2020

1. Formalice a continuación y describa la función que asigna un determinado valor a cada uno de los tipos de defensas.

$$f(\text{radio}, \text{rango}, \text{salud}) = (\text{radio} + \text{rango}) - 1 + \text{rango}^2 + 3 \frac{\text{radio}}{\text{salud}}$$

Escriba aquí su respuesta al ejercicio 1.

2. Describa la estructura o estructuras necesarias para representar la tabla de subproblemas resueltos.

Escriba aquí su respuesta al ejercicio 2.

3. En base a los dos ejercicios anteriores, diseñe un algoritmo que determine el máximo beneficio posible a obtener dada una combinación de defensas y *ases* disponibles. Muestre a continuación el código relevante.

```
// sustituya este código por su respuesta
void selectDefenses(...) {

    unsigned int cost = 0;
    std::list<Defense*>::iterator it = defenses.begin();
    while(it != defenses.end()) {
        if(cost + (*it)->cost <= ases) {
            selectedIDs.push_back((*it)->id);
            cost += (*it)->cost;
        }
        ++it;
    }
}
```

4. Diseñe un algoritmo que recupere la combinación óptima de defensas a partir del contenido de la tabla de subproblemas resueltos. Muestre a continuación el código relevante.

```
// sustituya este código por su respuesta
void selectDefenses(...) {

    unsigned int cost = 0;
    std::list<Defense*>::iterator it = defenses.begin();
    while(it != defenses.end()) {
        if(cost + (*it)->cost <= ases) {
            selectedIDs.push_back((*it)->id);
            cost += (*it)->cost;
        }
        ++it;
    }
}
```

Todo el material incluido en esta memoria y en los ficheros asociados es de mi autoría o ha sido facilitado por los profesores de la asignatura. Haciendo entrega de este documento confirmo que he leído la normativa de la asignatura, incluido el punto que respecta al uso de material no original.