Nombre: ....
Apellidos: ....
DNI: ....

Estructuras de Datos y Algoritmos — Examen final ordinario 2018/19
Grado en Desarrollo de Videojuegos. 2º V
Facultad de Informática, UCM

Instrucciones:
Esta primera parte del examen dura **50 minutos** y tiene una puntuación total de **3pt**.
No se puede encender el ordenador ni utilizar calculadora.

1. (0.5 pt) Calcula el coste asintótico de las siguientes dos funciones, indicando y explicando los principales pasos realizados.

```
int f1(int n) {
  int r = 1;
  while (n > 1) {
    r = r * n;
    n = n - 1;
  }
  return r;
}

int f2(int m) {
  int r = 0;
  for (int i = 0; i < m; ++i) {
    r = r + f1(i);
  }
  return r;
}
```

2. (0.5 pt) Un diccionario implementado como una tabla de dispersión abierta está definido como
unordered_map<K, V, Hash, Pred>. Explica con detalle cada uno de los cuatro parámetros de tipo que acepta.

3. (0.25 pt) En la técnica algorítmica de «vuelta atrás», ¿qué son los marcadores y para qué sirven? Pon un pequeño ejemplo.

4. (0.25 pt) Indica los costes de los métodos `empty`, `search` y `remove` de los árboles binarios de búsqueda (BST).

5. (0.25 pt) Obtén la recurrencia de coste $T(n)$ de la siguiente función recursiva.

```
int f3(unsigned int n) {
  if (n == 0)
    return 0;
  else
    return f3(n-1) + f3(n-1) + n;
}
```

6. (1.25 pt) A partir de la siguiente recurrencia $T(n)$ que representa el coste de un algoritmo recursivo, utiliza el método de las expansiones para calcular en qué orden de complejidad está incluida $T(n)$. Se deben realizar **todos los pasos** e indicar claramente el resultado obtenido en cada uno de ellos.

$$T(n) = \begin{cases} 8 & \text{si } n = 0 \\ 3T(n-1) + 5 & \text{si } n > 0 \end{cases}$$

Recordatorio

$\sum_{i=a}^{n}[f(i) + g(i)] = \sum_{i=a}^{n} f(i) + \sum_{i=a}^{n} g(i)$	$\sum_{i=a}^{n} k \cdot s_i = k \cdot \sum_{i=a}^{n} s_i$	$\sum_{i=a}^{n} i = \frac{(a+n)(n-a+1)}{2}$	$\sum_{i=0}^{n-1} k^i = \frac{1-k^n}{1-k}$