UNIVERSIDAD DE GRANADA

Departamento de Ciencias de la Computación e Inteligencia Artificial

Algorítmica (2023/24)
2º Grado en Ingeniería Informática
Ejemplo de examen

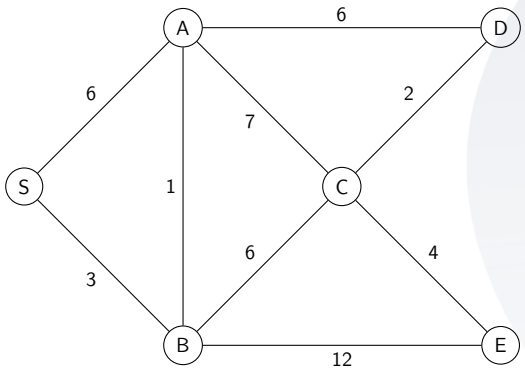
◁ Ejercicio 1 ▷ [2 puntos]

Calcular el orden de eficiencia en notación O(·) de la llamada al procedimiento `proc(v,0,n-1)`, donde `v` es un vector de `n` elementos:

```
int proc(const int *v, const int i, const in j) {
  if (i >= j)
    return v[i];
  else {
    int tam = j - i;
    int pt = i + (tam / 3);
    int st = i + (tam * 2) / 3;
    int rpt = proc(v, i, pt);
    int rtt = proc(v, st+1, j);
    int r = rpt + rtt;
    long s = 0;
    for (int k = i; k <= j; k++)
      s += v[k] + r - k;
    return s;
  }
}
```

◁ Ejercicio 2 ▷ [2 puntos]

Las galerías de una mina que conectan los puntos de extracción de un mineral (A, B, C, D y E, en la figura) con la salida (S, en la figura) se han derrumbado. El coste de excavar de nuevo cada galería es conocido *a priori* (ver el coste de las aristas del grafo en la figura). Se desea poder reabrir aquellas galerías que permitan conectar todos los puntos con la salida, pero con un coste lo menor posible. Diseñe un algoritmo eficiente que resuelva el problema en el caso general y aplíquelo a la instancia del ejemplo de la figura.



◁ Ejercicio 3 ▷ [2 puntos]

Un numero natural N se dice que es cuadrado perfecto si se corresponde con el cuadrado de otro número natural. Por ejemplo, $4 = 2^2$, $25 = 5^5$, $100 = 10^2$, son cuadrados perfectos. Diseñe un algoritmo «divide y vencerás» que, dado un número natural N de entrada, determine si el número es cuadrado perfecto o no.

◁ Ejercicio 4 ▷ [2 puntos]

Dado un tablero t de tamaño $n \times m$ de números naturales, se pretende resolver el problema de obtener el camino de la casilla $(0, 0)$ a la casillas $(n-1, m-1)$ que minimice la suma de los valores $t(i, j)$ de las casillas por las que pasa. En cada casilla (i, j) habrá solo dos posibles movimientos: ir hacia abajo, a la casilla $(i+1, j)$, o ir hacia la derecha, a la casilla $(i, j+1)$ (siempre que no nos salgamos del tablero). Diseñe un algoritmo lo más eficiente posible para resolver este problema mediante programación dinámica. Aplíquelo al siguiente tablero de tamañao 4×4:

2	8	3	4
5	3	4	5
1	2	2	1
3	4	6	5

◁ Ejercicio 5 ▷ [2 puntos]

Diseñe un algoritmo de exploración de grafos que, dado un número natural n, devuelva todas las formas posibles en que un conjunto ascendente de números enteros positivos distintos sume exactamente n. Por ejemplo, si $n = 10$, la salida debería ser:

$1 + 2 + 3 + 4$
$1 + 2 + 7$
$1 + 3 + 6$
$1 + 4 + 5$
$1 + 9$
$2 + 3 + 5$
$2 + 8$
$3 + 7$
$4 + 6$
10