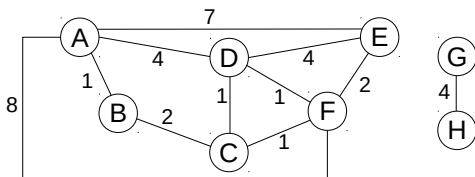


Taller Pre-Parcialito 4 [27/06] - Miércoles

1. Aplicar (y explicar) el algoritmo de Dijkstra para encontrar el camino mínimo desde A hacia el resto de los vértices en el grafo de la figura.



2. Aplicar el algoritmo de Kruskal para obtener el árbol de tendido mínimo del siguiente grafo, mostrando cómo se modifican las estructuras auxiliares en cada iteración. a) Aplicar al mismo grafo el algoritmo de Prim.

	A	B	C	D	E	F
A	0	4	2	5	0	7
B	4	0	6	0	6	0
C	2	6	0	1	4	0
D	5	0	1	0	3	8
E	0	6	4	3	0	8
F	7	0	0	8	8	0

3. Se tiene un arreglo con n enteros positivos, y se desea separar todos sus elementos en dos conjuntos disjuntos, tales que la diferencia entre la suma de los elementos de cada conjunto sea mínima.

Diseñar un algoritmo greedy para formar los dos conjuntos. Indicar el orden de la solución.

4. El problema del recorrido del caballo de ajedrez consiste en indicar una serie de 64 movimientos para que un caballo que se encuentra en una dada posición del tablero pise todas las casillas sin repetir ninguna.

Escribir el pseudocódigo de una función `bool recorrido_del_caballo(A, x, y, i)` que resuelva este problema usando backtracking sabiendo que:

- El parámetro A es una matriz de enteros de 8×8 .
- Los parámetros x e y indican la posición actual del caballo.
- i es el número de paso actual.
- La función devuelve true o false de acuerdo a si se encontró un recorrido o no.
- Se tiene una función `obtener_movimientos(x, y)` que dada una posición devuelve una lista con todos los movimientos posibles, descartando sólo aquellos que se excedan del tablero.

Inicialmente la función se llama con A llena de ceros, algún par de puntos x e y (entre 0 y 8) e i igual a 0. **Si la función devuelve true debe completar la matriz con el orden en el cual se visita cada casilla.**

5. Dado un arreglo de n números reales, se desea encontrar el valor máximo posible para la suma de una ventana contigua. Por ejemplo, para el arreglo $[3 \ -1 \ -5 \ 6 \ 7 \ -9]$, el valor máximo es 13, que se obtiene sumando $6 + 7$. Para resolverlo usando programación dinámica, se define $M[i]$ como el valor máximo de ventana que se puede obtener utilizando los primeros i elementos del arreglo. Para cada valor de i , el valor de $M[i]$ se puede formar extendiendo la ventana que termina en $i - 1$, o bien comenzando una ventana nueva en i . a) Mostrar la relación de recurrencia. b) Mostrar paso por paso la resolución para el arreglo $[-15 \ 29 \ -36 \ 3 \ -22 \ 11 \ 19 \ -5 \ 0 \ 4]$. c) ¿De qué orden es el algoritmo en tiempo y memoria?