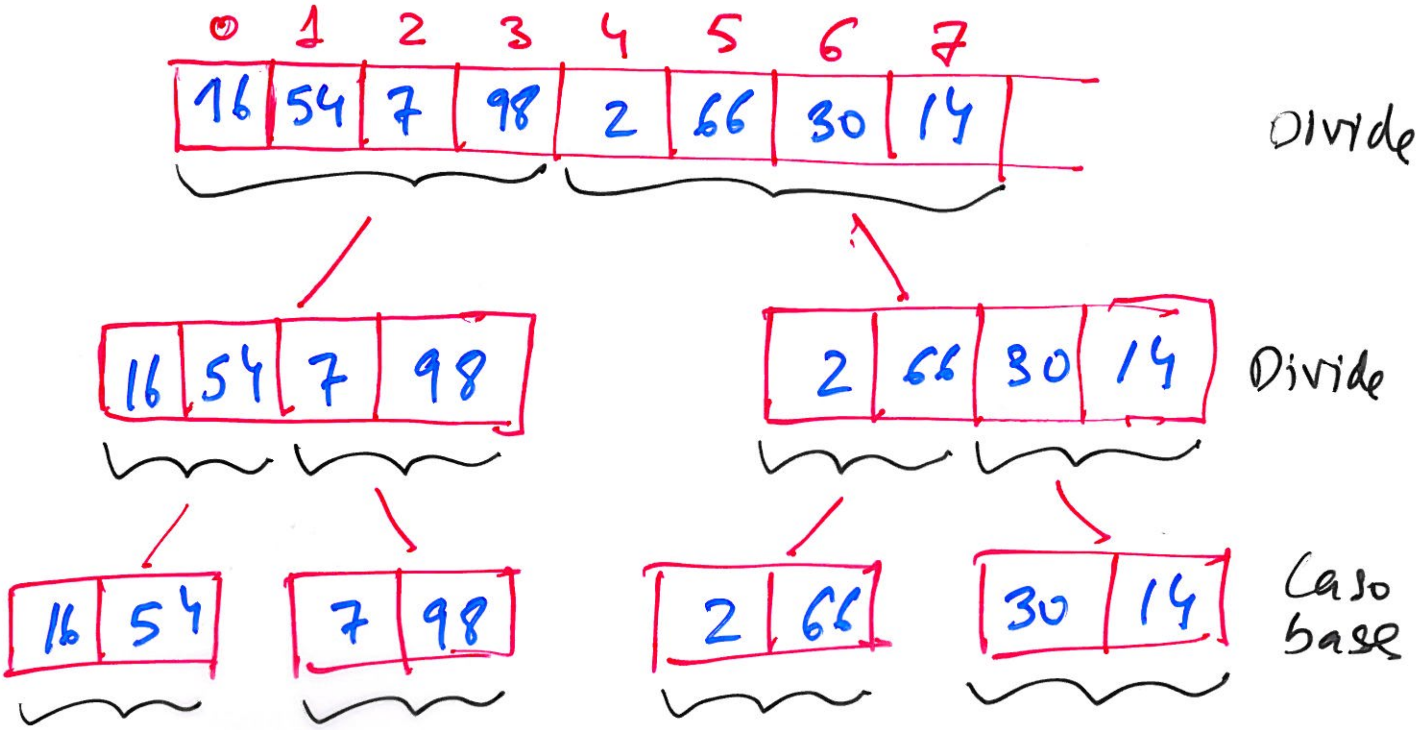
Dado un vector de enteros, determinar mediante DyV los elementos máximo y mínimo. El vector no está ordenado



Max=54 | Max=98 | Max=66 | Max=30
Min=16 | Min=7 | Min=2 | Min=14

Max=Max(54, 98)=98
Min= Min (16,7) =7

Max=Max(66, 30)=66
Min=Min(2, 14)=2

Max= Max(98, 66)=98
Min = Min (7, 2)=2

Solución

Max	Min
98	2

Idea:

Caso base: Si solo tenemos 2 números a comparar el máximo será el mayor y el mínimo el menor. Si solo tenemos 1 número, será tanto el max como el min

Dividir:

Se divide el vector en 2 subvectores de "igual" tam

Combinar:

Maximo: mayor(máximosubvector1, máximosubvector2)
Minimo: menor(mínimosubvector1, mínimosubvector2)

```
void calculamaxmin(~~Vector~~ vector, int max, min,
                                     int inf, sup);
  { int ~~medio~~, max1, max2, min1, min2;
  if (sup-inf) ≤ 1
          CASO BASE;
  else
      { medio = (inf + sup)/2;
      calculamaxmin(vector, max1, min1, inf, medio)
     calculamaxmin(vector, max2, min2, medio+1, sup);
      max = maximo(max1, max2);
      min = minimo(min1, min2);
     }
  }
```

Dado un vector de enteros de tamaño N, diseñar un algoritmo DyV para determinar la longitud de la secuencia estrictamente creciente más larga

3	2	7	4	8	4	6	7	2	5

Dado un vector que contiene secuencias de números enteros, diseñar un algoritmo DyV que determine la longitud de la secuencia más larga de una determinada clave K

6	6	4	4	4	4	4	6	6	6	6	6	3	3	3	

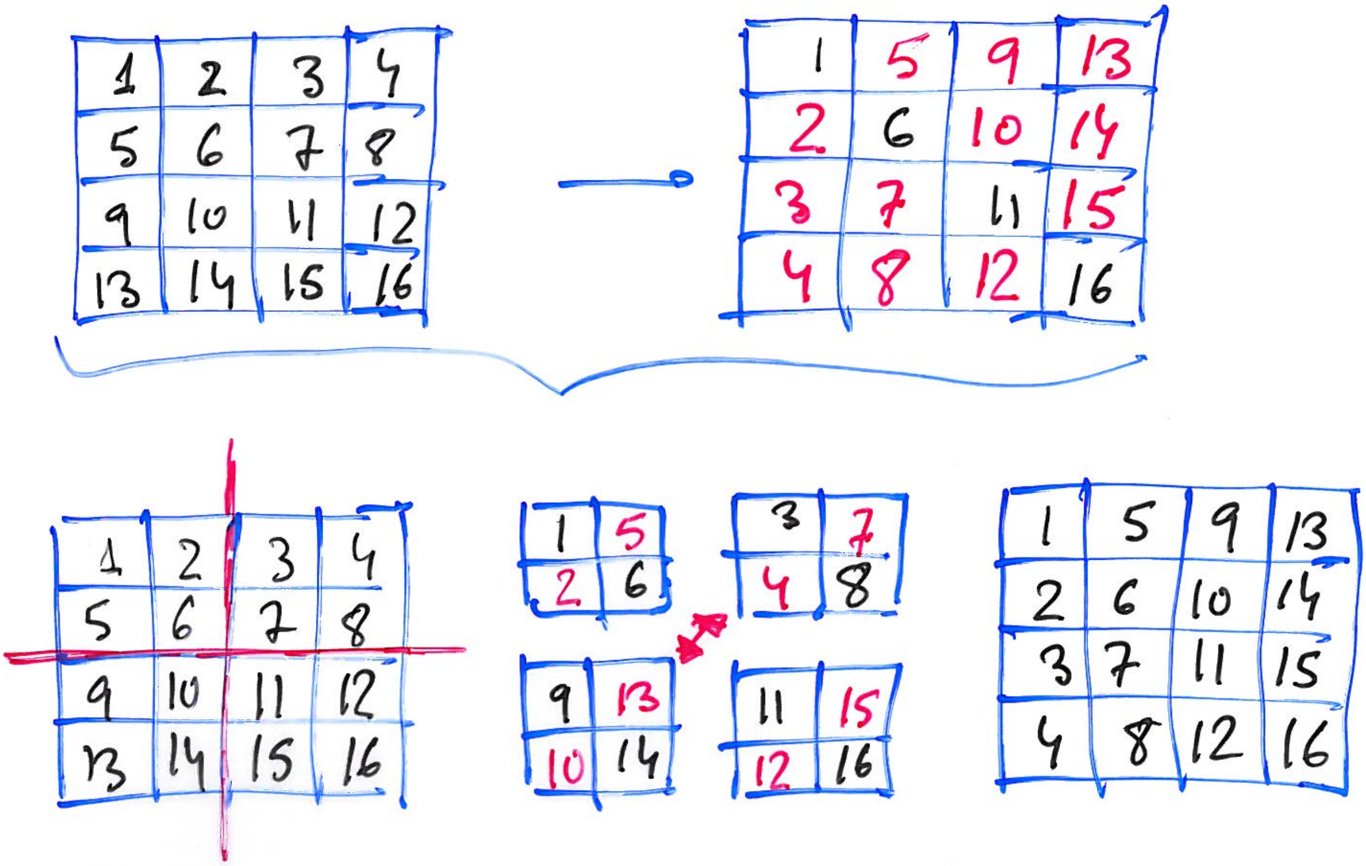
5

Dado un vector de enteros de tamaño N, diseñar un algoritmo DyV que determine la posición dentro del vector, si existe, de 2 elementos consecutivos que tengan el mismo signo y que su suma (en valor absoluto) sea máxima.

0	1	2	3	4	5	6	7	8	9	10
1	-2	20	-20	2	3	-3	-10	2	3	4

Devolvería 6 → elementos -3, -10, suma 13

Trasposición de una matriz usando DyV



Idea:

se intercambian los valores que NO están en la diagonal principal

Dividir

se divide la matriz en 4 submatrices de igual tamaño, trasponiendo cada una de ellas

Combinar:

intercambiar las dos submatrices que NO están en la diagonal principal

Caminos más cortos desde un nodo a los demás. (Algoritmo de Dijkstra)

Suponemos un grafo dirigido y ponderado con pesos no negativos, cuyos nodos están numerados: `0, ..., n-1`. Su matriz de costes es `C`, de tal modo que `C[i, j]` indica el coste de ir desde el nodo `i` al nodo `j`. Si no existe ese lado, el coste es ∞.

- `S`: Conjunto de vértices `{v}` para los que conocemos el camino mínimo de `0` a `v`.
- `D`: Vector que almacena en la posición `v`, el coste de viajar desde `0` a `v`, pasando sólo por nodos de `S`.
- `P`: Vector de nodos, tal que `P[v]` es el nodo anterior a `v` en el camino desde `0` a `v`.

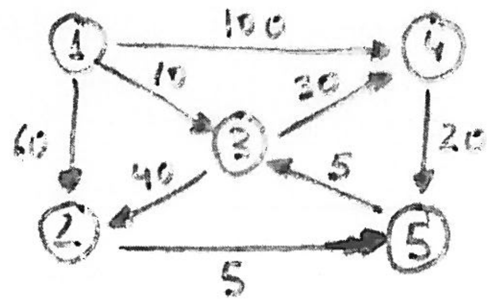
```
S = {0};
para (i = 1; i < n; i++)
  D[i] = C[0, i];
  P[i] = 0;
para (i = 0; i < n-1; i++)
  Escoger un vértice w ∈ V - S tal que
    D[w] es un mínimo;
  insertar(S, w);
  para (cada vértice v ∈ V - S)
    si (D[w] + C[w, v] < D[v])
      D[v] = D[w] + c[w, v];
      P[v] = w;
```

➢ Eficiencia: Matriz de Adyacencia: $O(n^2)$
Lista de Adyacencia: $O(a \log n)$

➢ El algoritmo es un ejemplo de Técnica "voraz" (Greedy).

➢ Ejemplo de aplicación: planificación de vuelos.

EJEMPLO



ENCONTRAR CAMINOS MINIMOS DEL VERTICE 1 A LOS DEMAS

INICIALMENTE

$S = \{1\}$

$D[2] = 60;$ $\boxed{D[3] = 10;}$ $D[4] = 100;$ $D[5] = \infty$

$P[i] = 1 \quad \forall i$

ITERACION 1

$V - S = \{2, 3, 4, 5\} \quad w = 3 \Rightarrow S = \{1, 3\} \Rightarrow V - S = \{2, 4, 5\}$

$D[2] = \min(D[2], D[3] + C[3,2]) = \min(60, 50) = 50 \Rightarrow P[2] = 3$

$D[4] = \min(D[4], D[3] + C[3,4]) = \min(100, 40) = 40 \Rightarrow P[4] = 3$

$D[5] = \min(D[5], D[3] + C[3,5]) = \min(\infty, \infty) = \infty$

$D[2] = 50;$ $\boxed{D[4] = 40;}$ $D[5] = \infty;$ $P[2] = 3;$ $P[4] = 3;$ $P[5] =$

ITERACION 2

$V - S = \{2, 4, 5\} \quad w = 4 \Rightarrow S = \{1, 3, 4\} \Rightarrow V - S = \{2, 5\}$

$D[2] = \min(D[2], D[4] + C[4,2]) = \min(50, \infty) = 50;$ $P[2] = 3$

$D[5] = \min(D[5], D[4] + C[4,5]) = \min(\infty, 60) = 60;$ $P[5] = 4$

$\boxed{D[2] = 50;}$ $D[5] = 60;$ $P[2] = 3;$ $P[5] = 4;$

ITERACION 3

$V - S = \{2, 5\} \quad w = 2 \Rightarrow S = \{1, 3, 4, 2\} \Rightarrow V - S = \{5\}$

$D[5] = \min(D[5], D[2] + C[2,5]) = \min(60, 55) = 55;$ $P[5] = 2.$

$\boxed{D[5] = 55;}$ $P[5] = 2$

FINALMENTE

$w = 5 \Rightarrow S = \{1, 3, 4, 2, 5\} \longrightarrow$ FIN

CAMINO DE 1 A 5

$P[5] = 2 \rightarrow P[2] = 3 \rightarrow P[3] = 1 \dashrightarrow$ $\boxed{(1, 3, 2, 5)}$ COSTO 55