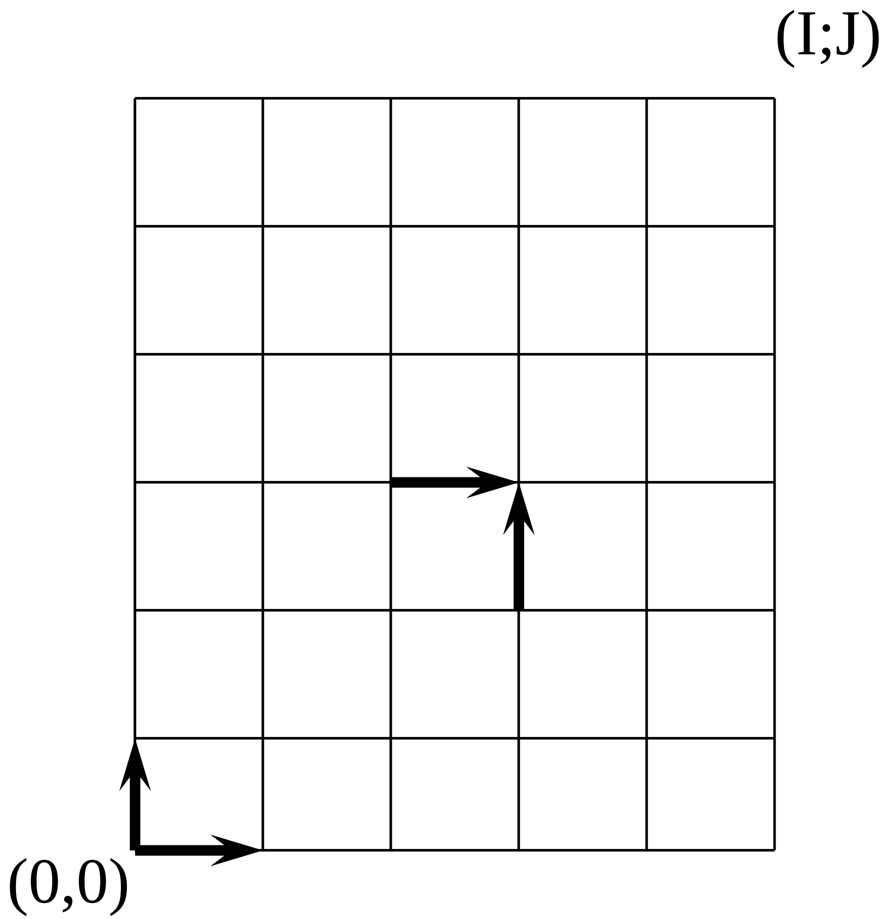
Problemas que NO son de optimización



$$ncam(i,j) = \begin{cases} 1 & i = 0 \vee j = 0 \\ ncam(i-1,j) + ncam(i,j-1) & i > 0 \vee j > 0 \end{cases}$$

```
función ncam (I,J:N):N;

var T:matriz[0..I,0..J] de N;

para i=0 hasta I hacer T[i,0]=1;
para j=0 hasta J hacer T[0,j]=1;

para i=1 hasta I hacer
  para j=1 hasta J hacer
      T[i,j]=T[i-1,j]+T[i,j-1];
  fpara
fpara
ncaminos=T[I,J];
fin;
```

$$
fib(n) = \begin{cases} 1 & n = 0 \vee n = 1 \\ fib(n-1) + fib(n-2) & n > 1 \end{cases}
$$

función fib(n:**N**)**N**;
var f1,f2,f3:**N**;

si n<=1 entonces fib=1
sino
f1=1; f2=1;
para i=2 hasta n hacer
   f3=f1+f2;
   f1=f2; f2;=f3;
fpara
fib=f3;
fin;

$$f(n,m)=\begin{cases}1 & n=m \vee m=0\\ f(n-1,m-1)+f(n-1,m) & n>m\end{cases}$$

n \ m	0	1	2	3	4	5
0	-					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1

```
función Combinaciones(int N,M)
 var T: matriz[0..N,0..M];

para i=1 hasta N hacer T[i,0]=1
                       T[i,i]=1
fpara;

para i=2 hasta N hacer
   para j=1 hasta i hacer
         T[i,j]= T[i-1,j-1]+T[i-1,j]
    fpara
fpara
return T[N,M]
```

Coste temporal $O(N^2)$

Coste espacial $O(N^2)$

Mejora del coste temporal

función Combinaciones2(int N,M)
 var T: **matriz**[0..N,0..M];

para i=1 **hasta** N **hacer** T[i,0]=1 **fpara**;

para i=2 **hasta** N **hacer**
  para j=1 **hasta** min(i,M) **hacer**
      T[i,j]= T[i-1,j-1]+T[i-1,j]
   fpara
fpara
return T[N,M]

m / n	0	1	2	3	4	5
0	-					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1

Coste temporal O(N.M)
Coste espacial O(N^2)

Mejora del coste espacial

función Combinaciones2(int N,M)
 var T: **matriz**[0..N,0..M];

para i=1 **hasta** N **hacer** T[i,0]=1 **fpara**;

para i=2 **hasta** N **hacer**
  para j=1 **hasta** min(i,M) **hacer**
      V2[j]= V1[j-1]+V1[j]
  fpara
  V1=V2
fpara
return V2[M]

n \ m	0	1	2	3	4	5
0	-					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1

Coste temporal O(N.M)
Coste espacial O(M)

Ajuste de líneas en un texto

Sea W un texto compuesto por la secuencia de palabras $w_1, w_2, \ldots, w_n$. Queremos que aparezca en K líneas de M caracteres cada líneay ajustado a izquierda y derecha. Para ello podemos introducir espacios en blanco entre las palabras. Una medida de lo “buena” que es una solución podría ser:

$$\sum_{k=1}^{K} \Phi(i_k, j_k)$$

donde $\Phi(i_k, j_k)$ es un valor asociado a la línea k, siendo i_k y j_k las palabras inicial y final de la línea k. Una posible definición de ϕ es

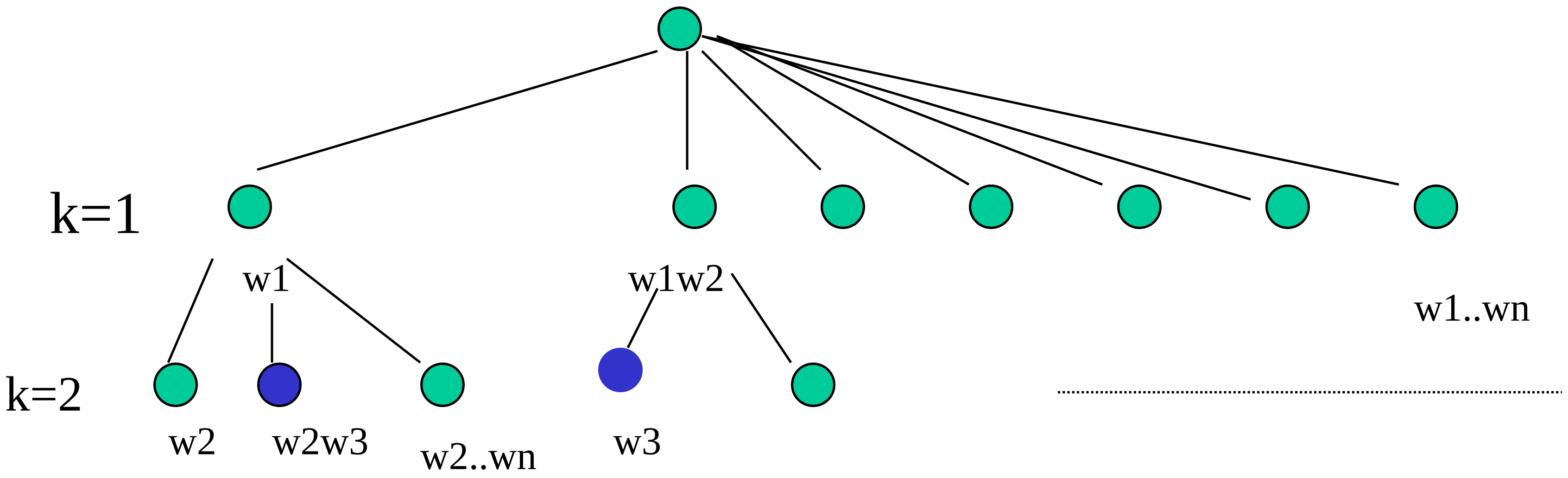
$$\Phi(i_k, j_k) = (M - (\sum_{n=i_k}^{j_k} l_n + (j_k - i_k))^2$$

siendo l_i la longitud de la palabra i.

Ejemplos de soluciones

w1	w1w2	w1w2w3
w2w3	w3	w4
w4w5	w4w5	w5

Arbol de soluciones



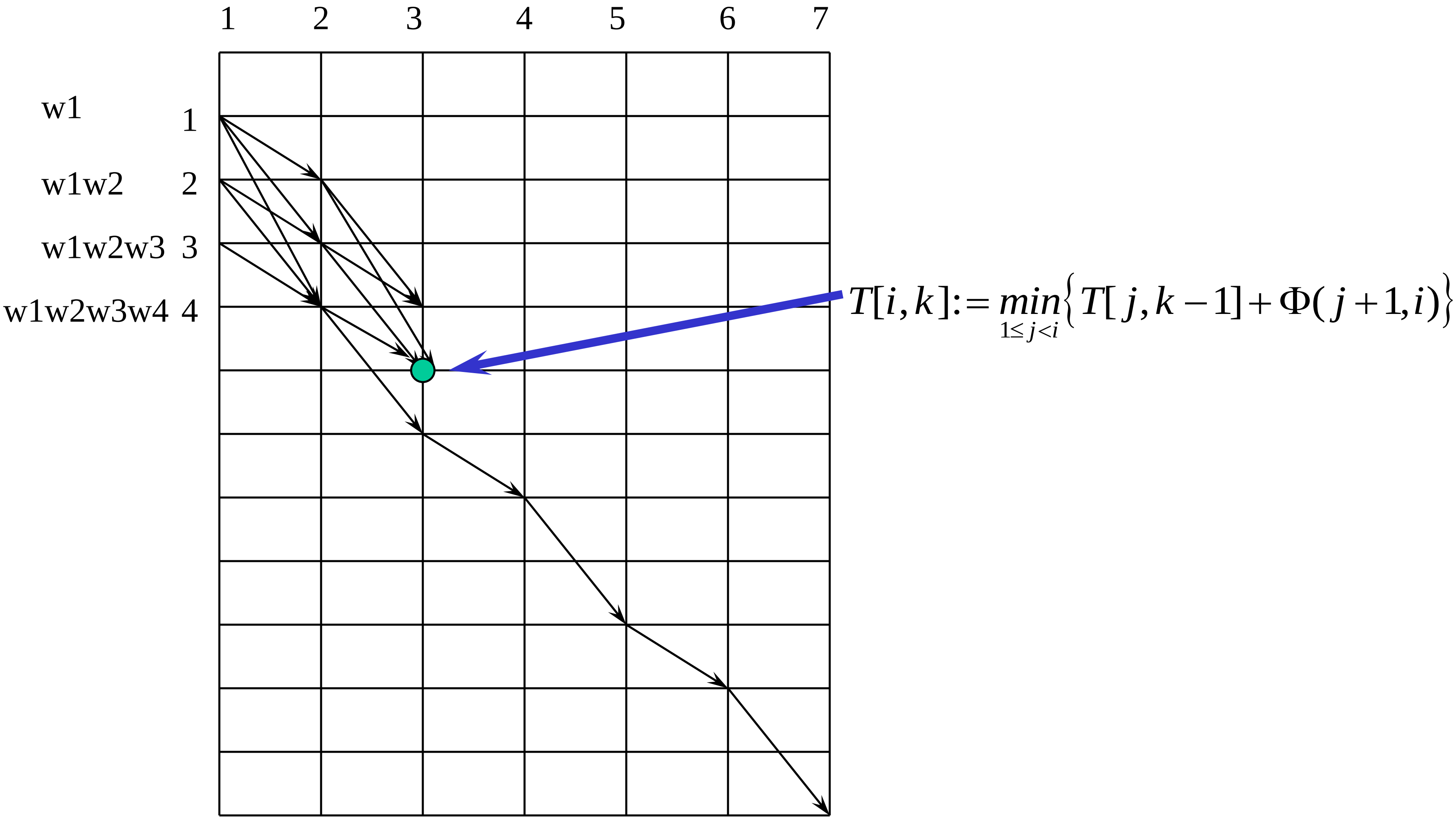
¿Cómo haber alineado hasta la palabra j con k líneas?

Ejemplo:

w1	w2	w3w4w5
	w2w3	w4w5
	w2w3w4	**w5**
w1w2	w3	w4w5
	w3w4	**w5**
w1w2w3	**w4**	**w5**

$$f(i,k) = \begin{cases} \Phi(1,i) & k = 1 \\ \min\limits_{\forall j, 1 \le j < i} \left\{ f(j,k-1) + \Phi(j+1,i) \right\} & k > 1 \end{cases}$$

Nota: Se considera que Φ (i,j) es infinito si no caben las palabras en la línea


1 2 3 4 5 6 7
w1 1
w1w2 2
w1w2w3 3
w1w2w3w4 4
$T[i,k] := \min_{1 \le j < i} \left\{ T[j,k-1] + \Phi(j+1,i) \right\}$

función ajustar(l:lista de **N**; n:**N**; K:**N**):**N**;
var T: **matriz**[1..n,1..K] de **N**;

para i=1 **hasta** n **hacer** T[i,1]=Φ (1,i) **fpara**;
para k=2 **hasta** K **hacer**
  para i=k **hasta** n **hacer**

$$T[i,k] = \min_{1 \le j < i} \{ T[j,k-1] + \Phi(j+1,i) \}$$

  fpara
fpara
ajustar=T[n,K];
fin.

Condiciones de contorno: No se ha tenido en consideración que la última línea no genera distorsión.

Con recuperación del camino (segmentación)

```
función ajustar2(l:lista de N; n:N; K:N);
var T,B: matriz[1..n,1..K] de N;

para i=1 hasta n hacer T[i,1]=Φ(1,i);
                       B[i,1]= -1;
fpara;
para k=2 hasta K hacer
  para i=k hasta N hacer
```

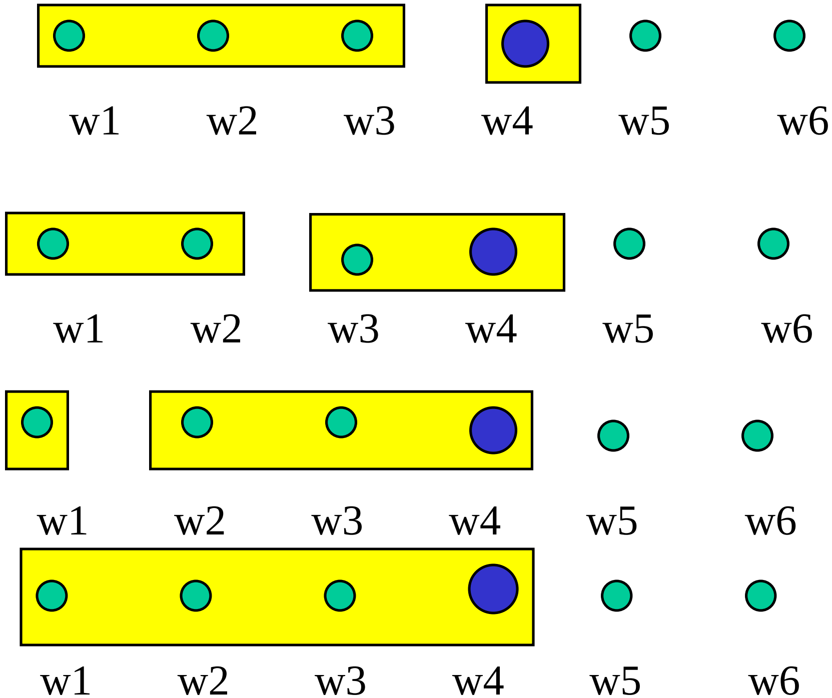
$$T[i,k] = \min_{k \le j < i} \{T[j,k-1] + \Phi(j+1,i)\}$$

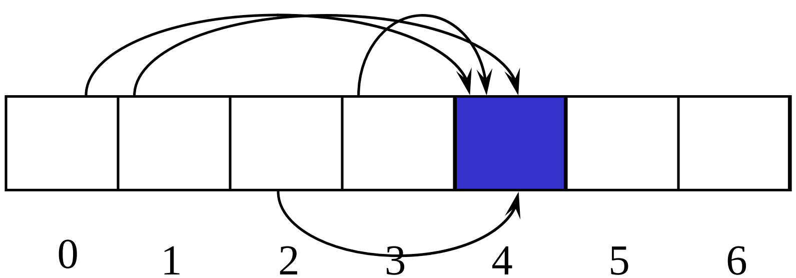
$$B[i,k] = \arg\min_{k \le j < i} \{T[j,k-1] + \Phi(j+1,i)\}$$

```
  fpara
fpara
i=N; k=K; lista= [n];
mientras k>1 hacer
          lista.insertar(B[i,k]);
          i=B[i,k]; k--;
fmientras
return(T[n,K], lista)
fin.
```

Problema: Alinear un texto de n palabras sin fijar el número de líneas.

Es un problema de segmentación.





$$V[i] = \min_{0 \le j < i} \{ V[j] + \Phi(j+1, i) \}$$

Problemas propuestos:

Problema 1: Dado un conjunto de k tipos de monedas, de los que se dispone una cantidad ilimitada de monedas de cada tipo, encontrar el mínimo conjunto de monedas con que se puede devolver una cantidad N.

Problema 2: Se quieren abonar N campos y para ello se dispone de M sacos de abono. El beneficio que produce abonar cada campo i, $1<=i<=N$ con m sacos de abono $0<=m<=M$ viene dado por la función $b(m,i)$. Se pide desarrollar un algoritmo de PD que devuelva mejor asignación de sacos a los campos, es decir aquella asignación que maximiza la suma de los beneficios obtenidos en cada campo.