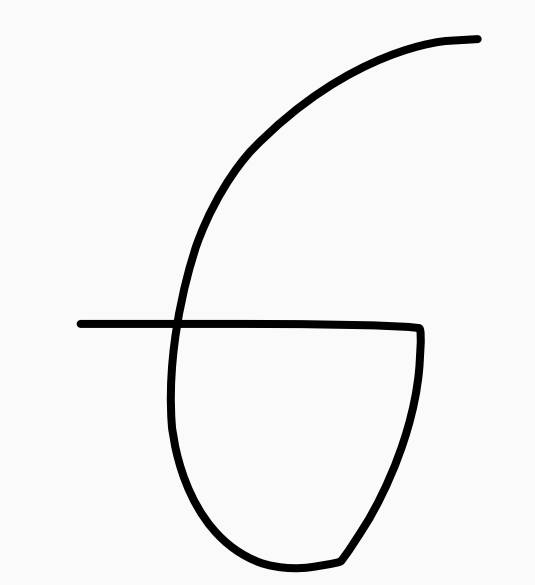


```
Merge-Sort(A, p, r)
  B, ans = Merge-Sort-Variation(A, p, r)
  return ans
```

```
Merge-Sort-Variation(A, p, r)
  if p < r
    q = piso((p+r)/2)
    P1, _ = Merge-Sort-Variation(A, p, q)
    P2, _ = Merge-Sort-Variation(A, q+1, r)
    B, ans3 = Merge(A, P1, P2)
    return B, ans3
  if p == r
    return [{ A[p], 1 }], -1
```

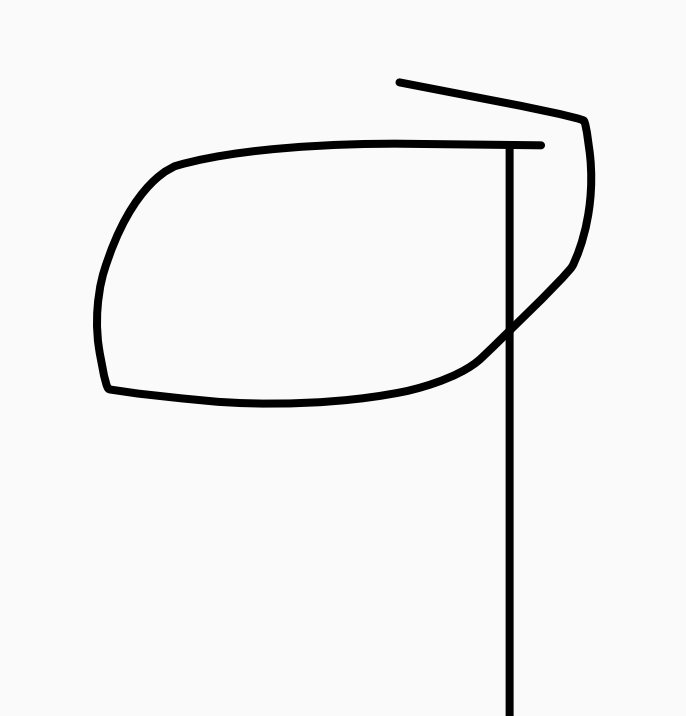
```
Merge(A, L, R)
  n1 = L.size
  n2 = R.size
  L[n1+1] = +inf
  R[n2+1] = +inf
  B = []
  i = j = 1
  while i != n1+1 && j != n2+1
    if L[i].first < R[j].first
      B.push_back(L[i])
      i++
    else if L[i].first > R[j].first
      B.push_back(R[j])
      j++
    else if L[i].first == R[j].first
      B.push_back({ L[i].first, L[i].second+ R[j].second })
      i++, j++
  if B[B.size] > A.size/2
    ans = 2
  return B, ans
```

```
T(n) = 2 T(piso(n/2)) + n
a = 2, b = 2, k = 1
lg2/lg2 = 1 = k
-> caso 2 MT
theta(n^k lgn)=theta(nlgn)
```



```
Pico(A, l, h)
  mid = (l+h)/2
  // pico
  if A[mid-1] < A[mid] > A[mid+1]
    return A[mid]
  // crece
  else if A[mid-1] < A[mid] < A[mid+1]
    return Pico(A, mid+1, h)
  // decrece
  else if A[mid-1] > A[mid] > A[mid+1]
    return Pico(A, l, mid-1)
```

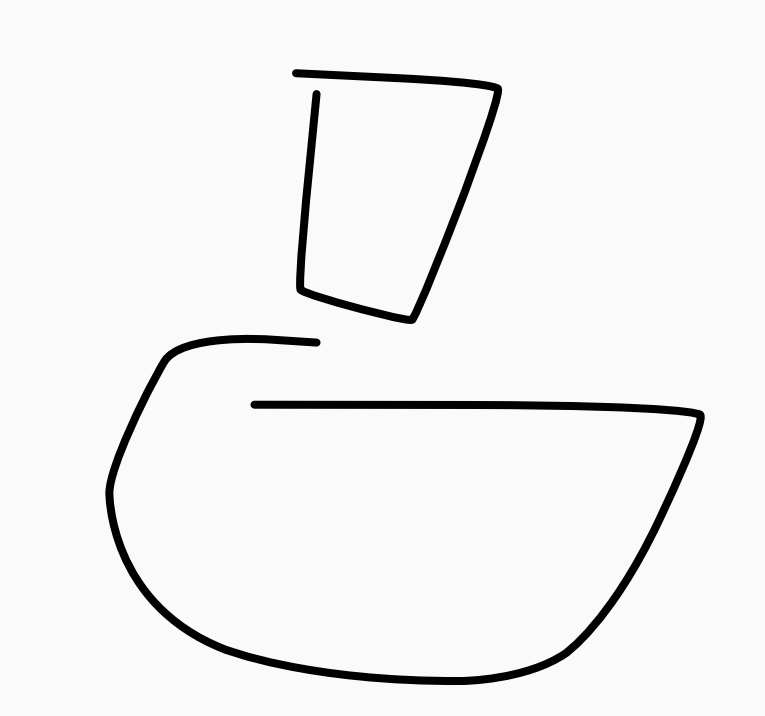
```
T(n) = T(n/2) + c
a=1, b=2, k=0
lg1/lg2=0/1=0=k
-> caso 2
theta(n^0 lgn)=theta(lgn)
```



```
// p y r son los índices de A1..k
Merge-Sort(A1..k, p, r)
  if p < r
    mid = piso((p+r)/2)
    L = Merge-Sort(C, p, mid)
    R = Merge-Sort(C, mid+1, r)
    return Merge(C, L, R)
  if p == r
    return Ap
```

```
Merge(C, L, R)
  n1 = L.size
  n2 = R.size
  L[n1 + 1] = +inf
  R[n2 + 1] = +inf
  i = j = 1
  B = [ ]
  while i!= n1+1 && j!= n2+1
    if L[i] <= R[j]
      B.push_back(L[i]) , i++
    else
      B.push_back(R[j]) , j++
  return B
```

```
T(n) = 2 T(piso(n/2)) + n
a = 2, b = 2, k = 1
lg2/lg2 = 1 = k
-> caso 2 MT
theta(n^k lgn)=theta(nlgn)
```



```
// indexado desde 1
K-Rotacion(A, B, l, h)
  mid = piso((l+h)/2)
  if A[mid] == B[mid]
    return 0
  else if A[1] == B[mid]
    return mid-1
  else if A[mid] < B[mid]
    return K-Rotacion(A, B, mid+1, h)
  else if A[mid] > B[mid]
    return K-Rotacion(A, b, l, mid-1)
```

```
T(n) = T(n/2) + c
a=1, b=2, k=0
lg1/lg2=0/1=0=k
-> caso 2
theta(n^0 lgn)=theta(lgn)
```