1. Algoritmo de particionamento:

```
==============================================================
Algoritmo: particionamento
Entrada: vetor A, índices i, p, f
Saída: índice do vetor (número natural)
--------------------------------------------------------------
01. aux := A[i], A[i] := A[p], A[p] := aux
02. a := i, b := f
    // Invariantes do laço da linha 3:
    // para todo i <= k < a, A[k] <= A[i]  <-- P(a)
    // para todo b < k <= f, A[k] >  A[i]  <-- P(b)
03. ENQUANTO a <= b
      // Invariantes do laço da linha 4:
      // a <= b+1, P(a)
04. | ENQUANTO a <= b e A[a] <= A[i]
05. | | ++a
      // Invariantes do laço da linha 6:
      // a <= b+1, P(b)
06. | ENQUANTO a <= b e A[b] > A[i]
07. | | --b
      // Se a <= b, então A[a] > A[i] e A[b] <= A[i], daí a troca abaixo.
08. | SE a <= b // ok se <
09. | | aux := A[a], A[a] := A[b], A[b] := aux
10. | | ++a, --b
    // para todo i <= k < a, A[k] <= A[i]
    // para todo b < k <= f, A[k] >  A[i]
    // a >= i, b <= f, a = b+1
11. aux := A[i], A[i] := A[b], A[b] := aux
12. RETORNE b.
==============================================================
```

2. Algoritmo de ordenação por particionamento ("quicksort"):

```
==============================================================
Algoritmo: ord_partic
Entrada: vetor A, índices i, f
Saída: nenhuma
--------------------------------------------------------------
1. SE i < f
2. | p := particionamento(A, i, piso((i+f)/2), f)
3. | ord_partic(A,i,p-1)
4. | ord_partic(A,p+1,f)
==============================================================
```

3. Complexidade dos algoritmos:

a) O algoritmo de particionamento executa em $\theta(n)$, sendo $n = f-i+1$.

b) O algoritmo de ordenação executa em tempo, no pior caso, limitado pela função a seguir:

$T(n) = a$, se $n < 2$, para algum $a > 0$.

$T(n) = \theta(n) + \text{máx}_{\{0 <= q < n\}} (T(q) + T(n-q-1))$.

4. EXERCÍCIO: mostre que $T(n) = O(n^2)$.
   Observação: você provavelmente precisará usar o fato de que $q^2 + (n-q-1)^2$ assume valor máximo quando $q=0$ ou $q=n-1$. Primeiro, faça a demonstração utilizando este fato; em seguida, tente demonstrá-lo.

5. No melhor caso, o custo do algoritmo é:

$T(n) = \theta(n) + 2*T(\text{piso}(n/2))$.

6. EXERCÍCIO: Usando o teorema mestre, mostre que o tempo de execução do algoritmo no melhor caso é $\theta(n * \lg n)$.