PIVO = AZUL

0 1 2 3 4 5

U M D O I S

dir

Esq

S M D O I U

int particiona(int \*V, int inicio(0), int fim(5)){

int esq, dir, pivo, aux;

esq (0)= inicio;

dir (5)= fim;

pivo (U)= V[inicio];

while(esq(0) < dir(5)){

while(V[esq] (U)(M)(D)(O)(I)(S)<= pivo(U)) {

esq++; (1)(2)(3)(4)(5)(6)

}

while(V[dir] (S) > pivo(U)){ 🡪FALSE

dir--; ()

}

if(esq(6) < dir(5)){ 🡪FALSE

aux ()= V[esq] ();

V[esq] () = V[dir] ();

V[dir] ()= aux();

} }

V[inicio] (0)= V[dir] (S);

V[dir] (5)= pivo(U);

return dir();

}

PIVO = AZUL

0 1 2 3 4 5

S M D O I U

dir

Esq

I M D O S U

int particiona(int \*V, int inicio(0), int fim(4)){

int esq, dir, pivo, aux;

esq (0)= inicio;

dir (4)= fim;

pivo (S)= V[inicio];

while(esq(0) < dir(4)){

while(V[esq] (S)(M)(D)(O)(I)<= pivo(S)) {

esq++; (1)(2)(3)(4)(5)

}

while(V[dir] (I) > pivo(S)){ 🡪FALSE

dir--; ()

}

if(esq(5) < dir(4)){ 🡪FALSE

aux ()= V[esq] ();

V[esq] () = V[dir] ();

V[dir] ()= aux();

} }

V[inicio] (0)= V[dir] (I);

V[dir] (4)= pivo(S);

return dir(4);

}

PIVO = AZUL

0 1 2 3 4 5

I M D O S U

Esq dir

I D M O S U

int particiona(int \*V, int inicio(0), int fim(3)){

int esq, dir, pivo, aux;

esq (0)= inicio;

dir (3)= fim;

pivo (I)= V[inicio];

while(esq(0) < dir(3)){

while(V[esq] (I)(M)<= pivo(I)) {🡪False

esq++; (1)

}

while(V[dir] (O)(D) > pivo(I)){ 🡪false

dir--; (2)

}

if(esq(1) < dir(2)){ 🡪verdadeiro

aux (M)= V[esq] (M);

V[esq] (1) = V[dir] (D);

V[dir] (2)= aux(M);

} }

V[inicio] ()= V[dir] ();

V[dir] ()= pivo();

return dir();

}

PIVO = AZUL

0 1 2 3 4 5

I M D O S U

I D M O S U

Dir Esq

D I M O S U

int particiona(int \*V, int inicio(0), int fim(3)){

int esq, dir, pivo, aux;

esq (0)= inicio;

dir (3)= fim;

pivo (I)= V[inicio];

while(esq(1) < dir(2)){

while(V[esq] (D)(M)<= pivo(I)) {🡪false

esq++; (2)

}

while(V[dir] (M)(D) > pivo(I)){ 🡪verdadeiro // false

dir--; (1)

}

if(esq(2) < dir(1)){ 🡪false

aux ()= V[esq] ();

V[esq] () = V[dir] ();

V[dir] ()= aux();

}

}

V[inicio] (0)= V[dir] (D);

V[dir] (1)= pivo(I);

return dir(1);

}

PIVO = AZUL

0 1 2 3 4 5

D I M O S U

Esq

Dir

int particiona(int \*V, int inicio(0), int fim(0)){

int esq, dir, pivo, aux;

esq (0)= inicio;

dir (0)= fim;

pivo (D)= V[inicio];

while(esq(0) < dir(0)){ 🡪false

while(V[esq] ()()<= pivo()) {

esq++; ()

}

while(V[dir] ()() > pivo()){

dir--; ()

}

if(esq() < dir()){

aux ()= V[esq] ();

V[esq] () = V[dir] ();

V[dir] ()= aux();

}

}

V[inicio] (0)= V[dir] (0);

V[dir] (0)= pivo(D);

return dir(0);

}

0 1 2 3 4 5

D I M O S U