Odd-Even Transposition Sort

```
#include<stdio.h>
#define N 6
__global__ void OE_sort(int *a,int n)
{
int tid=threadIdx.x;
 int i,t;
 for(i=1;i<=n/2;i++)
 {
    if(tid < n-1 && tid % 2 == 1)
    {
       t = a[tid+1];
       a[tid+1] = max(a[tid],t);
       a[tid] = min(a[tid],t);
    }
  __syncthreads();

 if(tid % 2 == 0)
 {
       t = a[tid+1];
       a[tid+1] = max(a[tid],t);
       a[tid] = min(a[tid],t);
 }
 __syncthreads();
 }
}
int main()
{
int a[]={70,80,60,40,50,30};
int i;

printf("\n Array elements before sorting:");
for(i=0;i<N;i++)
    printf("\t %d",a[i]);

int *d_a;
cudaMalloc((void**)&d_a,N*sizeof(int));

cudaMemcpy(d_a,&a,N*sizeof(int),cudaMemcpyHostToDevice);

OE_sort<<<1,N>>>(d_a,N);

cudaMemcpy(&a,d_a,N*sizeof(int),cudaMemcpyDeviceToHost);

printf("\n Array elements after sorting:");
for(i=0;i<N;i++)
    printf("\t %d",a[i]);
```

```
cudaFree(d_a);
return 0;
}
```

Output

```
Array elements before sorting:  70  80  60  40  50  30
Array elements after sorting:   30  40  50  60  70  80
```