

CS225

EX 1. I use C++ code to represent the algorithm.

```
void extract(T key)
{
    T temp;
    for (int i = numitem; i > 0; --i)
    {
        temp = this this->popfront();
        if (temp == key)
            continue;
        this->pushback(temp);
    }
}
```

in all cases, the time complexity is  $O(n)$  obviously.

EX2. Deques can turn insertion sort into an optimal sorting algorithm.

Since we can push the new value at the front and end of the sequence, the worst case will still be  $O(n^2)$ , but the single insertion will be  $\frac{n}{2}$  instead of  $n$ .

Ex3

$k$       ~~swapping~~  
↓      ↓  
[13, 21, 7, 8, 5, 1, 3, 2, 6, 11, 10, 9, 4, 17, 18]

j

$k$       i      j  
↓      ↓      ↓  
[9, 4, 17, 18, 5, 1, 3, 2, 6, 11, 10, 13, 21, 7, 8]

① | exchange( $k, k+j, i$ )  
 $j := j - i$

$k$       j      i  
↓      ↓      ↓  
[2, 6, 11, 10, 5, 1, 3, 9, 4, 17, 18, 13, 21, 7, 8]

① | exchange( $k, k+i, i$ )  
 $j := j - i$

i      j      k  
↓      ↓      ↓  
[5, 1, 3, 10, 2, 6, 11, 9, 4, 17, 18, 13, 21, 7, 8]

② | exchange  
( $k, k+i, i$ )  
 $i := i - j$   
 $k = k + j$

i      j      k  
↓      ↓      ↓  
[5, 1, 3, 11, 2, 6, 10, 9, 4, 17, 18, 13, 21, 7, 8]

①

i      j      k  
↓      ↓      ↓  
[5, 1, 3, 6, 2, 11, 10, 9, 4, 17, 18, 13, 21, 7, 8]

① | exchange( $k, k+i, i$ )  
end

out: [5, 1, 3, 2, 6, 11, 10, 9, 4, 17, 18, 13, 21, 7, 8]

~~Ex. 4. iii~~

Ex 4 (ii) we need to ~~use its~~ send all its children to the root and ~~not~~ mark its parent.

Pivot.

Ex 3 (iii) [13, 21, 7, 8, 5, 1, 3, 2, 6, 11, 10, 9, 4, 12, 18] ↓  
find the 7th smallest. [1] [13, 21, 7, 8, 5, 3, 6, 11, 10, 9, 4, 12, 18]  
[8, 5, 4, 3, 1, 2, 18] ↑  
pivot.

find the 5th smallest

(8)

[ ] [13, 21, 7, 8, 5, 6, 11, 10, 9, 4, 12, 18]

find the 4th smallest

(7) ↑  
pivot

find the 8smallest

[4, 5] [13, 21, 7, 8, 5, 6, 11, 10, 9, 4, 12, 18]  
↑  
pivot.

[8. find the smalles] [7, 8, 9, 10] [.....]  
↑  
pivot

find the smalles

[7]

[9]

↑  
pivot

[ ] [ ]

return 7;