

## Algoritmalar - Ödev 7

## 1) Selection sort

```

for (int i=0; i<n-1; i++)
    int min=i;
    for (int j=i+1; j<n; j++)
        if (array[j] < array[min])
            min=j;
    int temp = array[min];
    array[min] = array[i];
    array[i] = temp;

```

$$\begin{aligned}
 &1 + n + n-1 \\
 &n-1 \\
 &n-1 + n(n+1)/2 - 1 + n(n+1)/2 - n
 \end{aligned}$$

$$\begin{aligned}
 &n(n+1)/2 - n \\
 &n(n+1)/2 - n
 \end{aligned}$$

$$n-1$$

$$n-1$$

$$n-1$$

$$T(n) = n^2 + 7n - 6 //$$

## 2) Bubble sort

```

for (int i=0; i<n-1; i++)
    for (int j=0; j<n-i-1; j++)
        if (array[j] > array[j+1])
            int temp = array[j];
            array[j] = array[j+1];
            array[j+1] = temp;

```

$$1 + n + n-1$$

$$n-1 + n(n+1)/2 - 1 + (n-1)n/2$$

$$(n-1)n/2$$

$$(n-1)n/2$$

$$(n-1)n/2$$

$$(n-1)n/2$$

$$T(n) = 3n^2 + n - 2 //$$

## 3) Insertion sort

```

for (int i=1; i<n; i++)
    int temp = array[i];
    int j = i-1;
    while (temp < array[j] && j >= 0)
        array[j+1] = array[j];
        j--;
    array[j+1] = temp;

```

$$1 + n + n-1$$

$$n-1$$

$$n-1$$

$$2(n(n+1)/2 - 1)$$

$$(n-1)n/2$$

$$(n-1)n/2$$

$$n-1$$

$$T(n) = 2n^2 + 3n - 3 //$$

4)

- 1 i ile eşit integer
- 2 i n'den büyükse 11. adıma git
- 3 j ile eşit integer
- 4 i (n-1)'den büyükse 10. adıma git
- 5 nmax ile eşit integer
- 6 h ile eşit integer
- 7 h i'den büyük veya eşitse 9. adıma git
- 8 nmax'a array[j+h] ekle ve h i'den küçükse 7. adıma dön
- 9 eğer nmax max'dan büyükse max'ı nmax'a eşitle
- 10 j (n-i)'den küçük veya eşitse 4. adıma dön
- 11 i n'den küçük veya eşitse 2. adıma dön

for int i=1; i <= n; i++	1 + n+1 + n
for int j=0; j <= n-i; j++	n + (n+1)(n+2)/2 - 1 + n(n+1)/2
nmax = 0	n(n+1)/2
for int h=0; h < i; h++	n(n+1)/2 + (n+1)(n+2)(n+3)/6 - n-1 + n(n+1)(n+2)/6
nmax += array[j+h]	n(n+1)(n+2)/6
if nmax > max	n(n+1)(n+2)/6
max = nmax	n(n+1)(n+2)/6

$$T(n) = \frac{5}{6}n^3 + 5n^2 + \frac{49}{6}n + 2 //$$

Kadane algoritması

int max = array[0]	1
int tmax = array[0]	1
for (int i=0; i < n; i++)	1 + n+1 + n
tmax += array[i]	n
if tmax < array[i]	n
tmax = array[i]	n
if tmax > max	n
max = tmax	n

$$T(n) = 7n + 4 //$$

$$\text{Kadane } (7n + 4) < \text{Çözüm } \left(\frac{5}{6}n^3 + 5n^2 + \frac{49}{6}n + 2\right)$$

Kadane algoritmasının çalışma süresinin derecesi 2 birim daha düşük.

5)

$$a) \frac{n \log(n^2 + 1) + n^2 \log n}{2n \log n}$$

$$O(n^2 \log n)$$

$$b) \frac{(n \log n + 1)^2}{2n^2 \log n} + \frac{(\log n + 1)(n^2 + 1)}{n^2 \log n}$$

$$O(n^2 \log n)$$

$$c) \frac{n^{2^n} + n^{n^2}}{n^{2^n}}$$

$$O(n^n)$$