

Fremsta $\geq x$ í vaxandi fylki --- Lykkjulausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í vaxandi röð.
// Eftir: a[i..j-1] er óbreytt, i <= k <= j,
//        a[i..k-1] < x <= a[k..j-1].
//
//        Myndrænt:
//        | <x | >=x |
//        i    k    j
//
public static int search( double[] a, int i, int j, double x )
{
    int p=?2?, q=?3?;
    while( ?4? )
    {
        // ?1?
        //
        int m = ?5?;
        if( a[m] ?6? x )
            p = ?7?;
        else
            q = ?8?;
    }
    return ?9?;
}
```

Fremsta $\geq x$ í vaxandi fylki --- Endurkvæm lausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í vaxandi röð.
// Eftir: a[i..j-1] er óbreytt, i <= k <= j,
//        a[i..k-1] < x <= a[k..j-1].
//
//        Myndrænt:
//        | <x | >=x |
//        i    k    j
//
public static int search( double[] a, int i, int j, double x )
{
    if( ?1? ) return ?2?;
    int m = ?3?;
    if( a[m] ?4? x )
        return search(a,i,?5?,x);
    else
        return search(a,?6?,?7?,x);
}
```

Fremsta >x í vaxandi fylki --- Lykkjulausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir:  a[i..j-1] er svæði í a sem er í vaxandi röð.
// Eftir:  a[i..j-1] er óbreytt, i <= k <= j,
//         a[i..k-1] <= x < a[k..j-1].
//
//         Myndrænt:
//         | <=x | >x |
//         i     k     j
//
public static int search( double[] a, int i, int j, double x )
{
    int p=?2?, q=?3?;
    while( ?4? )
    {
        // ?1?
        //
        int m = ?5?;
        if( a[m] ?6? x )
            p = ?7?;
        else
            q = ?8?;
    }
    return ?9?;
}
```

Fremsta >x í vaxandi fylki --- Endurkvæm lausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir:  a[i..j-1] er svæði í a sem er í vaxandi röð.
// Eftir:  a[i..j-1] er óbreytt, i <= k <= j,
//         a[i..k-1] <= x < a[k..j-1].
//
//         Myndrænt:
//         | <=x | >x |
//         i     k     j
//
public static int search( double[] a, int i, int j, double x )
{
    if( ?1? ) return ?2?;
    int m = ?3?;
    if( a[m] ?4? x )
        return search(a,i,?5?,x);
    else
        return search(a,?6?,?7?,x);
}
```

Aftasta $\leq x$ í vaxandi fylki --- Lykkjulausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í vaxandi röð.
// Eftir: a[i..j-1] er óbreytt, i-1 ≤ k ≤ j-1,
//        a[i..k] ≤ x < a[k+1..j-1].
//
//        Myndrænt:
//        | ≤x | >x |
//        i   k   j
//
public static int search( double[] a, int i, int j, double x )
{
    int p=??, q=??;
    while( ?? )
    {
        // ??
        //
        int m = ??;
        if( a[m] ?? x )
            p = ??;
        else
            q = ??;
    }
    return ??;
}
```

Aftasta $\leq x$ í vaxandi fylki --- Endurkvæm lausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í vaxandi röð.
// Eftir: a[i..j-1] er óbreytt, i-1 ≤ k ≤ j-1,
//        a[i..k] ≤ x < a[k+1..j-1].
//
//        Myndrænt:
//        | ≤x | >x |
//        i   k   j
//
public static int search( double[] a, int i, int j, double x )
{
    if( ?? ) return ??;
    int m = ??;
    if( a[m] ?? x )
        return search(a,i,??,x);
    else
        return search(a,??,??,x);
}
```

Aftasta <x í vaxandi fylki --- Lykkjulausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í vaxandi röð.
// Eftir: a[i..j-1] er óbreytt, i-1 <= k <= j-1,
//        a[i..k] < x <= a[k+1..j-1].
//
//        Myndrænt:
//        | <x | >=x |
//        i   k       j
//
public static int search( double[] a, int i, int j, double x )
{
    int p=?2?, q=?3?;
    while( ?4? )
    {
        // ?1?
        //
        int m = ?5?;
        if( a[m] ?6? x )
            p = ?7?;
        else
            q = ?8?;
    }
    return ?9?;
}
```

Aftasta <x í vaxandi fylki --- Endurkvæm lausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í vaxandi röð.
// Eftir: a[i..j-1] er óbreytt,
//        i-1 <= k <= j-1,
//        a[i..k] < x <= a[k+1..j-1].
//
//        Myndrænt:
//        | <x | >=x |
//        i   k       j
//
public static int search( double[] a, int i, int j, double x )
{
    if( ?1? ) return ?2?;
    int m = ?3?;
    if( a[m] ?4? x )
        return search(a,i,?5?,x);
    else
        return search(a,?6?,?7?,x);
}
```

Fremsta $\leq x$ í minnkandi fylki --- Lykkjulausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í minnkandi röð.
// Eftir: a[i..j-1] er óbreytt, i <= k <= j,
//        a[i..k-1] > x >= a[k..j-1].
//
//        Myndrænt:
//        | >x | <=x |
//        i   k   j
//
public static int search( double[] a, int i, int j, double x )
{
    int p=?2?, q=?3?;
    while( ?4? )
    {
        // ?1?
        //
        int m = ?5?;
        if( a[m] ?6? x )
            p = ?7?;
        else
            q = ?8?;
    }
    return ?9?;
}
```

Fremsta $\leq x$ í minnkandi fylki --- Endurkvæm lausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í minnkandi röð.
// Eftir: a[i..j-1] er óbreytt,
//        i <= k <= j,
//        a[i..k-1] > x >= a[k..j-1].
//
//        Myndrænt:
//        | >x | <=x |
//        i   k   j
//
public static int search( double[] a, int i, int j, double x )
{
    if( ?1? ) return ?2?;
    int m = ?3?;
    if( a[m] ?4? x )
        return search(a,i,?5?,x);
    else
        return search(a,?6?,?7?,x);
}
```

Fremsta $<x$ í minnkandi fylki --- Lykkjulausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í minnkandi röð.
// Eftir: a[i..j-1] er óbreytt,  $i \leq k \leq j$ ,
//         $a[i..k-1] \geq x > a[k..j-1]$ .
//
//        Myndrænt:
//        |  $\geq x$  |  $< x$  |
//        i      k      j
//
public static int search( double[] a, int i, int j, double x )
{
    int p=?2?, q=?3?;
    while( ?4? )
    {
        // ?1?
        //
        int m = ?5?;
        if( a[m] ?6? x )
            p = ?7?;
        else
            q = ?8?;
    }
    return ?9?;
}
```

Fremsta $<x$ í minnkandi fylki --- Endurkvæm lausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í minnkandi röð.
// Eftir: a[i..j-1] er óbreytt,
//         $i \leq k \leq j$ ,
//         $a[i..k-1] \geq x > a[k..j-1]$ .
//
//        Myndrænt:
//        |  $\geq x$  |  $< x$  |
//        i      k      j
//
public static int search( double[] a, int i, int j, double x )
{
    if( ?1? ) return ?2?;
    int m = ?3?;
    if( a[m] ?4? x )
        return search(a,i,?5?,x);
    else
        return search(a,?6?,?7?,x);
}
```

Aftasta $\geq x$ í minnkandi fylki --- Lykkjulausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í minnkandi röð.
// Eftir: a[i..j-1] er óbreytt, i-1 <= k <= j-1,
//        a[i..k] >= x > a[k+1..j-1].
//
//        Myndrænt:
//        | >=x | <x |
//        i    k    j
//
public static int search( double[] a, int i, int j, double x )
{
    int p=?2?, q=?3?;
    while( ?4? )
    {
        // ?1?
        //
        int m = ?5?;
        if( a[m] ?6? x )
            p = ?7?;
        else
            q = ?8?;
    }
    return ?9?;
}
```

Aftasta $\geq x$ í minnkandi fylki --- Endurkvæm lausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í minnkandi röð.
// Eftir: a[i..j-1] er óbreytt,
//        i-1 <= k <= j-1,
//        a[i..k] >= x > a[k+1..j-1].
//
//        Myndrænt:
//        | >=x | <x |
//        i    k    j
//
public static int search( double[] a, int i, int j, double x )
{
    if( ?1? ) return ?2?;
    int m = ?3?;
    if( a[m] ?4? x )
        return search(a,i,?5?,x);
    else
        return search(a,?6?,?7?,x);
}
```

Aftasta >x í minnkandi fylki --- Lykkjulausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í minnkandi röð.
// Eftir: a[i..j-1] er óbreytt, i-1 <= k <= j-1,
//        a[i..k] > x >= a[k+1..j-1].
//
//        Myndrænt:
//        | >x | <=x |
//        i   k       j
//
public static int search( double[] a, int i, int j, double x )
{
    int p=?2?, q=?3?;
    while( ?4? )
    {
        // ?1?
        //
        int m = ?5?;
        if( a[m] ?6? x )
            p = ?7?;
        else
            q = ?8?;
    }
    return ?9?;
}
```

Aftasta >x í minnkandi fylki --- Endurkvæm lausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í minnkandi röð.
// Eftir: a[i..j-1] er óbreytt,
//        i-1 <= k <= j-1,
//        a[i..k] > x >= a[k+1..j-1].
//
//        Myndrænt:
//        | >x | <=x |
//        i   k       j
//
public static int search( double[] a, int i, int j, double x )
{
    if( ?1? ) return ?2?;
    int m = ?3?;
    if( a[m] ?4? x )
        return search(a,i,?5?,x);
    else
        return search(a,?6?,?7?,x);
}
```


Eitthvert x í vaxandi fylki --- Lykkjulausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í vaxandi röð.
// Eftir: a[i..j-1] er óbreytt,
//       Ef x er til í a[i..j-1] þá er  $i \leq k < j$  og  $a[k] == x$ ,
//       annars er  $k < 0$  og  $i \leq -k-1 \leq j$  og
//        $a[i..-k-2] < x < a[-k-1..j-1]$ .
//
//       Myndrænt í fyrra tilviki:
//       | <=x |x| >=x |
//       i     k     j
//
//       Myndrænt í seinna tilviki:
//       | <x | >x |
//       ^     ^     ^
//       i     -k-1    j
//
public static int search( double[] a, int i, int j, double x )
{
    int p=?2?, q=?3?;
    while( ?4? )
    {
        // ?1?
        //
        int m = ?5?;
        if( a[m]==x ) return ?6?;
        if( a[m] ?7? x )
            p = ?8?;
        else
            q = ?9?;
    }
    return ?10?;
}
```

Eitthvert x í vaxandi fylki --- Endurkvæm lausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í vaxandi röð.
// Eftir: a[i..j-1] er óbreytt,
//       Ef x er til í a[i..j-1] þá er i<=k<j og a[k]==x,
//       annars er k<0 og i <= -k-1 <= j og
//       a[i..-k-2] < x < a[-k-1..j-1].
//
//       Myndrænt í fyrra tilviki:
//       | <=x |x| >=x |
//       i     k     j
//
//       Myndrænt í seinna tilviki:
//       | <x | >x |
//       ^     ^     ^
//       i    -k-1    j
//
public static int search( double[] a, int i, int j, double x )
{
    if( ?1? ) return ?2?;
    int m = ?3?;
    if( a[m]==x ) return ?4?;
    if( a[m] ?5? x )
        return search(a,i,?6?,x);
    else
        return search(a,?7?,?8?,x);
}
```

Eitthvert x í minnkandi fylki --- Lykkjulausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í minnkandi röð.
// Eftir: a[i..j-1] er óbreytt,
//       Ef x er til í a[i..j-1] þá er i<=k<j og a[k]==x,
//       annars er k<0 og i <= -k-1 <= j og
//       a[i..-k-2] > x > a[-k-1..j-1].
//
//       Myndrænt í fyrra tilviki:
//       | >=x |x| <=x |
//       i     k     j
//
//       Myndrænt í seinna tilviki:
//       |   >x   |   <x   |
//       ^       ^       ^
//       i     -k-1     j
//
public static int search( double[] a, int i, int j, double x )
{
    int p=?2?, q=?3?;
    while( ?4? )
    {
        // ?1?
        //
        int m = ?5?;
        if( a[m]==x ) return ?6?;
        if( a[m] ?7? x )
            p = ?8?;
        else
            q = ?9?;
    }
    return ?10?;
}
```

Eitthvert x í minnkandi fylki ---

Endurkvæm lausn

```
// Notkun: int k = search(a,i,j,x);
// Fyrir: a[i..j-1] er svæði í a sem er í minnkandi röð.
// Eftir: a[i..j-1] er óbreytt,
//       Ef x er til í a[i..j-1] þá er  $i \leq k < j$  og  $a[k] == x$ ,
//       annars er  $k < 0$  og  $i \leq -k-1 \leq j$  og
//        $a[i..-k-2] > x > a[-k-1..j-1]$ .
//
//       Myndrænt í fyrra tilviki:
//       | >=x |x| <=x |
//       i     k     j
//
//       Myndrænt í seinna tilviki:
//       |   >x   |   <x   |
//       ^       ^       ^
//       i     -k-1     j
//
public static int search( double[] a, int i, int j, double x )
{
    if( ?1? ) return ?2?;
    int m = ?3?;
    if( a[m]==x ) return ?4?;
    if( a[m] ?5? x )
        return search(a,i,?6?,x);
    else
        return search(a,?7?,?8?,x);
}
```