|                                                                        | Notes |
|------------------------------------------------------------------------|-------|
| Linear and binary search                                               |       |
| Michael Nowak                                                          |       |
| Texas A&M University                                                   |       |
|                                                                        |       |
|                                                                        |       |
|                                                                        |       |
|                                                                        |       |
|                                                                        |       |
| Overview                                                               | Notes |
|                                                                        |       |
| MyArray                                                                |       |
| Searching for some value in an array<br>Linear Search<br>Binary Search |       |
|                                                                        |       |
|                                                                        |       |
|                                                                        |       |
|                                                                        |       |
|                                                                        |       |
| Overview                                                               | Notes |
| MyArray                                                                |       |
|                                                                        |       |
| Searching for some value in an array Linear Search Binary Search       |       |
|                                                                        |       |
|                                                                        |       |

### MyArray

```
1  #ifndef MYARRAY_H
2  #define MYARRAY_H
3  struct MyArray {
4    int *arr = nullptr;
5    unsigned int capacity = 0; // no elements can store
6    unsigned int size = 0; // no elements currently held
7  };
8  #endif
```

| U | ) V ( | erv | 'ie | W |
|---|-------|-----|-----|---|
|---|-------|-----|-----|---|

### MyArray

Searching for some value in an array Linear Search Binary Search

### Setting-up

| 1<br>2<br>3 | #include <iostream> #include "MyArray.h" #include "linearSearch.h"</iostream> |
|-------------|-------------------------------------------------------------------------------|
| 1 -         | #include "binarySearch.h"                                                     |
| 5           | ··                                                                            |
| 6           | using namespace std;                                                          |
| 7           |                                                                               |
| 8           | int main()                                                                    |
| 9           | {                                                                             |
| 11          | <pre>MyArray ma1; ma1.arr = new int[7];</pre>                                 |
| 12          | mal.capacity = 7;                                                             |
| 13          | ma1.size = 7;                                                                 |
| 14          | for (unsigned int i = 0; i < ma1.size; ++i)                                   |
| 15          | ma1.arr[i] = i * 10;                                                          |
| 16          |                                                                               |
| 17          | <pre>unsigned int idxofvalue = linearSearch(ma1, 20);</pre>                   |
| 18          |                                                                               |
| 19<br>20    | <pre>unsigned int idxofvalue2 = binarySearch(ma1, 20);</pre>                  |
| 21          | delete [] ma1.arr;                                                            |
| 22          | doloso (j mariali,                                                            |
| 23          | return 0;                                                                     |
| 24          | }                                                                             |

| Notes | N | ot | es |
|-------|---|----|----|
|-------|---|----|----|

### Notes

### Notes

| - |  |  |  |
|---|--|--|--|
|   |  |  |  |

### Setting-up

# Notes \_\_\_\_\_

### Setting-up

```
1  #include <iostream>
2  #include "MyArray.h"
3  #include "binarySearch.h"
4  #include "binarySearch.h"
5
6  using namespace std;
7
8  int main()
9  {
10     MyArray mai;
11     mai.arr = new int[7];
12     mai.capacity = 7;
13     mai.size = 7;
14     for (unsigned int i = 0; i < mai.size; ++i)
15     mai.arr[i] = i * 10;
16
17     unsigned int idxofvalue = linearSearch(mai, 20);
18
19     unsigned int idxofvalue2 = binarySearch(mai, 20);
20     delete [] mai.arr;
22     return 0;
24 }</pre>
```

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

### Setting-up

| 1  | #include <iostream></iostream>                               |
|----|--------------------------------------------------------------|
| 2  | #include "MyArray.h"                                         |
| 3  | #include "linearSearch.h"                                    |
| 4  | #include "binarySearch.h"                                    |
| 5  |                                                              |
| 6  | using namespace std;                                         |
| 7  |                                                              |
| 8  | int main()                                                   |
| 9  | {                                                            |
| 10 | MyArray ma1;                                                 |
| 11 | <pre>ma1.arr = new int[7];</pre>                             |
| 12 | ma1.capacity = 7;                                            |
| 13 | ma1.size = 7;                                                |
| 14 | for (unsigned int i = 0; i < ma1.size; ++i)                  |
| 15 | ma1.arr[i] = i * 10;                                         |
| 16 |                                                              |
| 17 | <pre>unsigned int idxofvalue = linearSearch(ma1, 20);</pre>  |
| 18 |                                                              |
| 19 | <pre>unsigned int idxofvalue2 = binarySearch(ma1, 20);</pre> |
| 20 |                                                              |
| 21 | delete [] ma1.arr;                                           |
| 22 |                                                              |
| 23 | return 0;                                                    |
| 24 | }                                                            |

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

### Setting-up

```
1  #include <iostream>
2  #include "MyArray.h"
3  #include "linearSearch.h"
4  #include "binarySearch.h"
5
6  using namespace std;
7
8  int main()
9  {
10      MyArray ma1;
11      ma1.arr = new int[7];
12      ma1.size = 7;
13      ma1.size = 7;
14      for (unsigned int i = 0; i < ma1.size; ++i)
15      ma1.arr[i] = i * 10;
16
17      unsigned int idxofvalue = linearSearch(ma1, 20);
18
19      unsigned int idxofvalue2 = binarySearch(ma1, 20);
20      delete [] ma1.arr;
22      return 0;
24 }</pre>
```

# Notes \_\_\_\_\_

### Setting-up

```
1  #include <iostream>
2  #include "MyArray.h"
3  #include "binarySearch.h"
4  #include "binarySearch.h"
5
6  using namespace std;
7
8  int main()
9  {
10    MyArray mai;
11    mai.arr = new int[7];
12    mai.capacity = 7;
13    mai.size = 7;
14   for (unsigned int i = 0; i < mai.size; ++i)
15    mai.arr[i] = i * 10;
16
17    unsigned int idxofvalue = linearSearch(mai, 20);
18
19    unsigned int idxofvalue2 = binarySearch(mai, 20);
20    delete [] mai.arr;
22
23    return 0;
24 }</pre>
```

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

### Setting-up

| 1  | #include <iostream></iostream>                               |
|----|--------------------------------------------------------------|
| 2  | #include "MyArray.h"                                         |
| 3  | #include "linearSearch.h"                                    |
| 4  | #include "binarySearch.h"                                    |
| 5  |                                                              |
| 6  | using namespace std;                                         |
| 7  |                                                              |
| 8  | int main()                                                   |
| 9  | {                                                            |
| 10 | MyArray ma1;                                                 |
| 11 | <pre>ma1.arr = new int[7];</pre>                             |
| 12 | ma1.capacity = 7;                                            |
| 13 | ma1.size = 7;                                                |
| 14 | for (unsigned int i = 0; i < ma1.size; ++i)                  |
| 15 | ma1.arr[i] = i * 10;                                         |
| 16 |                                                              |
| 17 | <pre>unsigned int idxofvalue = linearSearch(ma1, 20);</pre>  |
| 18 |                                                              |
| 19 | <pre>unsigned int idxofvalue2 = binarySearch(ma1, 20);</pre> |
| 20 |                                                              |
| 21 | delete [] ma1.arr;                                           |
| 22 |                                                              |
| 23 | return 0;                                                    |
| 24 | }                                                            |

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

| Overview                                                                                                                                                                                                               | Notes |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| MyArray                                                                                                                                                                                                                |       |
| Searching for some value in an array Linear Search Binary Search                                                                                                                                                       |       |
|                                                                                                                                                                                                                        |       |
|                                                                                                                                                                                                                        |       |
| Calling our linearSearch function  1 #include <iostream></iostream>                                                                                                                                                    | Notes |
| 1 #include Clostream> 2 #include "MyArray.h" 3 #include "linearSearch.h" 4 #include "binarySearch.h" 5 6 using namespace std; 7                                                                                        |       |
| <pre>8  int main() 9  { 10     MyArray ma1; 11     ma1.arr = new int[7]; 12     ma1.capacity = 7;</pre>                                                                                                                |       |
| <pre>14</pre>                                                                                                                                                                                                          |       |
| <pre>19    unsigned int idxofvalue2 = binarySearch(ma1, 20); 20 21    delete [] ma1.arr; 22 23    return 0; 24 }</pre>                                                                                                 |       |
|                                                                                                                                                                                                                        |       |
| linearSearch()                                                                                                                                                                                                         | Notes |
| <pre>1  #include "linearSearch.h" 2 3  int linearSearch(MyArray const &amp;mya, int valSearchFor) 4  { 5    for (unsigned int i = 0; i &lt; mya.size; ++i) { 6    if (mya.arr[i] == valSearchFor) 7    return i;</pre> |       |
| 8 } 9 return -1; 10 }                                                                                                                                                                                                  |       |

### linearSearch()

### linearSearch()

```
1  #include "linearSearch.h"
2
3  int linearSearch(MyArray const &mya, int valSearchFor)
4  {
5     for (unsigned int i = 0; i < mya.size; ++i) {
6        if (mya.arr[i] == valSearchFor)
7          return i;
8     }
9     return -1;
10 }</pre>
```

### linearSearch()

```
1 #include "linearSearch.h"
2
3 int linearSearch(MyArray const &mya, int valSearchFor)
4 {
5 for (unsigned int i = 0; i < mya.size; ++i) {
6 if (mya.arr[i] == valSearchFor)
7 return i;
8 }
9 return -1;
10 }</pre>
```

### Notes

| - |  |  |
|---|--|--|
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |

### Notes

| • |  |
|---|--|
|   |  |
|   |  |
|   |  |
|   |  |
|   |  |

### Notes

### linearSearch()

### linearSearch()

### linearSearch()

```
1 #include "linearSearch.h"
2
3 int linearSearch(MyArray const &mya, int valSearchFor)
4 {
5     for (unsigned int i = 0; i < mya.size; ++i) {
6         if (mya.arr[i] == valSearchFor)
7             return i;
8     }
9     return -1;
10 }</pre>
```

Notes

| - |  |  |  |
|---|--|--|--|
| _ |  |  |  |
|   |  |  |  |
| - |  |  |  |

### Notes

| - |  |  |  |
|---|--|--|--|
| - |  |  |  |

### Notes

### linearSearch()

### linearSearch()

```
1  #include "linearSearch.h"
2
3  int linearSearch(MyArray const &mya, int valSearchFor)
4  {
5     for (unsigned int i = 0; i < mya.size; ++i) {
6        if (mya.arr[i] == valSearchFor)
7          return i;
8     }
9     return -1;
10 }</pre>
```

### Returned to main

```
1  #include <iostream>
2  #include "MyArray.h"
3  #include "linearSearch.h"
4  #include "binarySearch.h"
5
6  using namespace std;
7
8  int main()
9  {
10     MyArray ma1;
11     ma1.arr = new int[7];
12     ma1.capacity = 7;
13     ma1.size = 7;
14     for (unsigned int i = 0; i < ma1.size; ++i)
15         mal.arr[i] = i * 10;
16
17     unsigned int idxofvalue = linearSearch(ma1, 20);
18
19     unsigned int idxofvalue2 = binarySearch(ma1, 20);
20     delete [] ma1.arr;
22
23     return 0;
24 }</pre>
```

| Votes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

| 1 | Notes |  |  |  |  |
|---|-------|--|--|--|--|
| _ |       |  |  |  |  |
|   |       |  |  |  |  |
|   |       |  |  |  |  |
|   |       |  |  |  |  |
|   |       |  |  |  |  |
|   |       |  |  |  |  |
| - |       |  |  |  |  |

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

| Overview                                                                                                                              | Notes |
|---------------------------------------------------------------------------------------------------------------------------------------|-------|
| MyArray                                                                                                                               |       |
| Searching for some value in an array Linear Search Binary Search                                                                      |       |
| Sindly Scaren                                                                                                                         |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
| Binary Search                                                                                                                         | Notes |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
| ► Requirement: data is sorted!                                                                                                        |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
|                                                                                                                                       |       |
| Calling our binary search function  1  #include <iostream> 2  #include "MyArray.h" 3  #include "linearSearch.h"</iostream>            | Notes |
| <pre>4 #include "binarySearch.h" 5 6 using namespace std;</pre>                                                                       |       |
| 7 8 int main() 9 {                                                                                                                    |       |
| 10 MyArray ma1;<br>11 mal.arr = new int[7];<br>12 ma1.capacity = 7;                                                                   |       |
| <pre>13     ma1.size = 7; 14     for (unsigned int i = 0; i &lt; ma1.size; ++i) 15     ma1 arr[i] = i * 10.</pre>                     |       |
| 16                                                                                                                                    |       |
| unsigned int idxofvalue2 = binarySearch(ma1, 20);  delete [] ma1.arr;                                                                 |       |
| unsigned int idxofvalue = linearsearch(mai, 20);  unsigned int idxofvalue2 = binarySearch(mai, 20);  delete [] mai.arr;  return 0;  4 |       |

## Notes

### binarySearch()

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

| $\overline{}$ |                                                        | _ |
|---------------|--------------------------------------------------------|---|
| 1             | #include "binarySearch.h"                              |   |
| 2             | #include <iostream></iostream>                         |   |
| 3             |                                                        |   |
| 4             | int binarySearch(MyArray const &mya, int valSearchFor) |   |
| 5             | {                                                      |   |
| 6             | /* return (any) position if valSearchFor is in sorted  |   |
| 1             | mya.arr[0size-1] or -1 if valSearchFor is not          |   |
|               | present */                                             |   |
| 7             | unsigned int lowerBound = 0;                           |   |
| 8             | unsigned int upperBound = mya.size - 1;                |   |
| 9             | while (lowerBound <= upperBound) {                     |   |
| 10            | unsigned int midpt = (lowerBound + upperBound) / 2;    |   |
| 11            | <pre>if (mya.arr[midpt] &lt; valSearchFor)</pre>       |   |
| 12            | lowerBound = midpt + 1;                                |   |
| 13            | else if (mya.arr[midpt] == valSearchFor)               |   |
| 14            | return midpt;                                          |   |
| 15            | else /* mya.arr[midpt] > valSearchFor */               |   |
| 16            | upperBound = midpt - 1;                                |   |
| 17            | }                                                      |   |
| 18            | return -1;                                             |   |
| 19            | }                                                      |   |
| 1-2           | •                                                      |   |

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

## Notes

### binarySearch()

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

| 1  | #include "binarySearch.h"                                      |
|----|----------------------------------------------------------------|
| 2  | #include <iostream></iostream>                                 |
| 3  |                                                                |
| 4  | int binarySearch(MyArray const &mya, int valSearchFor)         |
| 5  | {                                                              |
| 6  | /* return (any) position if valSearchFor is in sorted          |
|    | mya.arr[0size-1] or -1 if valSearchFor is not                  |
|    | present */                                                     |
| 7  | unsigned int lowerBound = 0;                                   |
| 8  | unsigned int upperBound = mya.size - 1;                        |
| 9  | <pre>while (lowerBound &lt;= upperBound) {</pre>               |
| 10 | <pre>unsigned int midpt = (lowerBound + upperBound) / 2;</pre> |
| 11 | <pre>if (mya.arr[midpt] &lt; valSearchFor)</pre>               |
| 12 | <pre>lowerBound = midpt + 1;</pre>                             |
| 13 | <pre>else if (mya.arr[midpt] == valSearchFor)</pre>            |
| 14 | return midpt;                                                  |
| 15 | else /* mya.arr[midpt] > valSearchFor */                       |
| 16 | upperBound = midpt - 1;                                        |
| 17 | }                                                              |
| 18 | return -1;                                                     |
| 19 | }                                                              |

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

### Notes

### binarySearch()

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

| 1  | #include "binarySearch.h"                                        |
|----|------------------------------------------------------------------|
|    | #include <iostream></iostream>                                   |
| 3  |                                                                  |
| 4  | int binarySearch(MyArray const &mya, int valSearchFor)           |
| 5  | {                                                                |
| 6  | <pre>/* return (any) position if valSearchFor is in sorted</pre> |
|    | mya.arr[0size-1] or -1 if valSearchFor is not                    |
|    | present */                                                       |
| 7  | unsigned int lowerBound = 0;                                     |
| 8  | unsigned int upperBound = mya.size - 1;                          |
| 9  | <pre>while (lowerBound &lt;= upperBound) {</pre>                 |
| 10 | <pre>unsigned int midpt = (lowerBound + upperBound) / 2;</pre>   |
| 11 | <pre>if (mya.arr[midpt] &lt; valSearchFor)</pre>                 |
| 12 | <pre>lowerBound = midpt + 1;</pre>                               |
| 13 | <pre>else if (mya.arr[midpt] == valSearchFor)</pre>              |
| 14 | return midpt;                                                    |
| 15 | else /* mya.arr[midpt] > valSearchFor */                         |
| 16 | upperBound = midpt - 1;                                          |
| 17 | }                                                                |
| 18 | return -1;                                                       |
| 19 | }                                                                |

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

```
1  #include "binarySearch.h"
2  #include <iostream>
3
4  int binarySearch(MyArray const &mya, int valSearchFor)
5  {
6     /* return (any) position if valSearchFor is in sorted mya.arr[0..size-1] or -1 if valSearchFor is not present */
7     unsigned int lowerBound = 0;
8     unsigned int upperBound = mya.size - 1;
9     while (lowerBound <= upperBound) {
10         unsigned int midpt = (lowerBound + upperBound) / 2;
11         if (mya.arr[midpt] < valSearchFor)
12         lowerBound = midpt + 1;
13         else if (mya.arr[midpt] == valSearchFor)
14         return midpt;
15         else /* mya.arr[midpt] > valSearchFor */
16         upperBound = midpt - 1;
17     }
18     return -1;
19 }
```

### Notes

### binarySearch()

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

| 1  | #include "binarySearch.h"                                        |
|----|------------------------------------------------------------------|
| 2  | #include <iostream></iostream>                                   |
| 3  |                                                                  |
| 4  | int binarySearch(MyArray const &mya, int valSearchFor)           |
| 5  | {                                                                |
| 6  | <pre>/* return (any) position if valSearchFor is in sorted</pre> |
|    | mya.arr[0size-1] or -1 if valSearchFor is not                    |
|    | present */                                                       |
| 7  | unsigned int lowerBound = 0;                                     |
| 8  | unsigned int upperBound = mya.size - 1;                          |
| 9  | <pre>while (lowerBound &lt;= upperBound) {</pre>                 |
| 10 | <pre>unsigned int midpt = (lowerBound + upperBound) / 2;</pre>   |
| 11 | <pre>if (mya.arr[midpt] &lt; valSearchFor)</pre>                 |
| 12 | <pre>lowerBound = midpt + 1;</pre>                               |
| 13 | <pre>else if (mya.arr[midpt] == valSearchFor)</pre>              |
| 14 | return midpt;                                                    |
| 15 | else /* mya.arr[midpt] > valSearchFor */                         |
| 16 | upperBound = midpt - 1;                                          |
| 17 | }                                                                |
| 18 | return -1;                                                       |
| 19 | }                                                                |
|    |                                                                  |

| Notes |  |  |
|-------|--|--|
|       |  |  |
|       |  |  |
|       |  |  |
|       |  |  |
|       |  |  |
|       |  |  |
|       |  |  |

```
1  #include "binarySearch.h"
2  #include <iostream>
3
4  int binarySearch(MyArray const &mya, int valSearchFor)
5  {
6     /* return (any) position if valSearchFor is in sorted mya.arr[0..size-1] or -1 if valSearchFor is not present */
7     unsigned int lowerBound = 0;
8     unsigned int upperBound = mya.size - 1;
9     while (lowerBound <= upperBound) {
10         unsigned int midpt = (lowerBound + upperBound) / 2;
11     if (mya.arr[midpt] < valSearchFor)
12     lowerBound = midpt + 1;
13     else if (mya.arr[midpt] == valSearchFor)
14         return midpt;
15     else /* mya.arr[midpt] > valSearchFor */
16         upperBound = midpt - 1;
17     }
18     return -1;
19 }
```

Notes

### binarySearch()

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

| 1  | #include "binarySearch.h"                              |
|----|--------------------------------------------------------|
| 2  | #include <iostream></iostream>                         |
| 3  |                                                        |
| 4  | int binarySearch(MyArray const &mya, int valSearchFor) |
| 5  | {                                                      |
| 6  | /* return (any) position if valSearchFor is in sorted  |
|    | mya.arr[0size-1] or -1 if valSearchFor is not          |
|    | present */                                             |
| 7  | unsigned int lowerBound = 0;                           |
| 8  | unsigned int upperBound = mya.size - 1;                |
| 9  | <pre>while (lowerBound &lt;= upperBound) {</pre>       |
| 10 | unsigned int midpt = (lowerBound + upperBound) / 2;    |
| 11 | <pre>if (mya.arr[midpt] &lt; valSearchFor)</pre>       |
| 12 | lowerBound = midpt + 1;                                |
| 13 | <pre>else if (mya.arr[midpt] == valSearchFor)</pre>    |
| 14 | return midpt;                                          |
| 15 | <pre>else /* mya.arr[midpt] &gt; valSearchFor */</pre> |
| 16 | upperBound = midpt - 1;                                |
| 17 | }                                                      |
| 18 | return -1;                                             |
| 19 | 1                                                      |
| 1  |                                                        |

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

### Notes \_\_\_\_\_\_

### binarySearch()

# Notes

| 1  | #include "binarySearch.h"                                                                |  |  |  |  |  |
|----|------------------------------------------------------------------------------------------|--|--|--|--|--|
| 2  | #include <iostream></iostream>                                                           |  |  |  |  |  |
| 3  |                                                                                          |  |  |  |  |  |
| 4  | int binarySearch(MyArray const &mya, int valSearchFor)                                   |  |  |  |  |  |
| 5  | {                                                                                        |  |  |  |  |  |
| 6  | /* return (any) position if valSearchFor is in sorted                                    |  |  |  |  |  |
|    | mya.arr[0size-1] or -1 if valSearchFor is not                                            |  |  |  |  |  |
| 7  | present */                                                                               |  |  |  |  |  |
| 8  | unsigned int lowerBound = 0;                                                             |  |  |  |  |  |
| 9  | <pre>unsigned int upperBound = mya.size - 1; while (lowerBound &lt;= upperBound) {</pre> |  |  |  |  |  |
| 10 | unsigned int midpt = (lowerBound + upperBound) / 2;                                      |  |  |  |  |  |
| 11 | if (mya.arr[midpt] < valSearchFor)                                                       |  |  |  |  |  |
| 12 | lowerBound = midpt + 1;                                                                  |  |  |  |  |  |
| 13 | else if (mya.arr[midpt] == valSearchFor)                                                 |  |  |  |  |  |
| 14 | return midpt;                                                                            |  |  |  |  |  |
| 15 | else /* mya.arr[midpt] > valSearchFor */                                                 |  |  |  |  |  |
| 16 | upperBound = midpt - 1;                                                                  |  |  |  |  |  |
| 17 | }                                                                                        |  |  |  |  |  |
| 18 | return -1;                                                                               |  |  |  |  |  |
| 19 | }                                                                                        |  |  |  |  |  |
|    |                                                                                          |  |  |  |  |  |

| Notes |  |  |
|-------|--|--|
|       |  |  |
|       |  |  |
|       |  |  |
|       |  |  |
|       |  |  |
|       |  |  |

### Notes

### Returned to main

```
1  #include <iostream>
2  #include "MyArray.h"
3  #include "binarySearch.h"
4  #include "binarySearch.h"
5
6  using namespace std;
7
8  int main()
9  {
10    MyArray ma1;
11    ma1.arr = new int[7];
12    ma1.capacity = 7;
13    ma1.size = 7;
14    for (unsigned int i = 0; i < ma1.size; ++i)
15         ma1.arr[i] = i * 10;
16
17    unsigned int idxofvalue = linearSearch(ma1, 20);
18
19    unsigned int idxofvalue2 = binarySearch(ma1, 20);
20    delete [] ma1.arr;
22
23    return 0;
24 }</pre>
```

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

### Returned to main

| 1  | #include <iostream></iostream>                               |
|----|--------------------------------------------------------------|
| 2  | #include "MyArray.h"                                         |
| 3  | #include "linearSearch.h"                                    |
| 4  | #include "binarySearch.h"                                    |
| 5  |                                                              |
| 6  | using namespace std;                                         |
| 7  |                                                              |
| 8  | int main()                                                   |
| 9  | {                                                            |
| 10 | MyArray ma1;                                                 |
| 11 | ma1.arr = new int[7];                                        |
| 12 | mal.capacity = 7;                                            |
| 13 | ma1.size = 7;                                                |
| 14 | for (unsigned int i = 0; i < ma1.size; ++i)                  |
| 15 | ma1.arr[i] = i * 10;                                         |
| 16 | ,                                                            |
| 17 | <pre>unsigned int idxofvalue = linearSearch(ma1, 20);</pre>  |
| 18 |                                                              |
| 19 | <pre>unsigned int idxofvalue2 = binarySearch(ma1, 20);</pre> |
| 20 |                                                              |
| 21 | delete [] ma1.arr;                                           |
| 22 |                                                              |
| 23 | return 0;                                                    |
| 24 | }                                                            |
| 24 | J                                                            |

| Notes |  |  |  |
|-------|--|--|--|
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |
|       |  |  |  |

### Returned to main

```
unsigned int idxofvalue = linearSearch(ma1, 20);
    unsigned int idxofvalue2 = binarySearch(ma1, 20);
```

| Notes |  |
|-------|--|
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
| Notes |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
| Notes |  |
| Notes |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |
|       |  |