
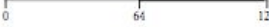
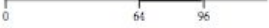
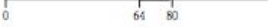
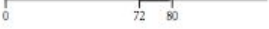





Binary Search and Ternary Search

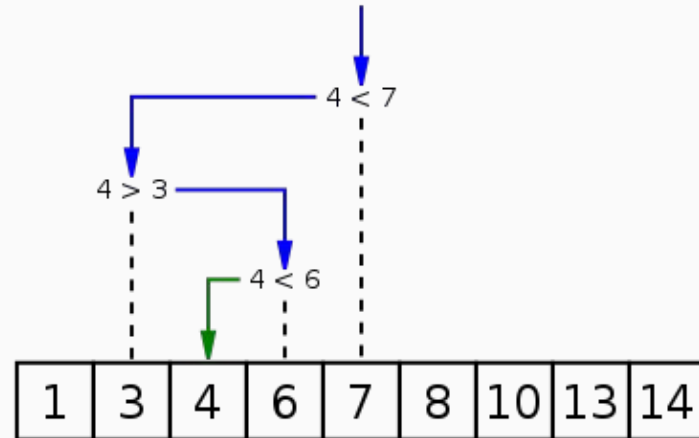
Peter, Alex, Harrison

<http://bit.ly/1MwE0bL>

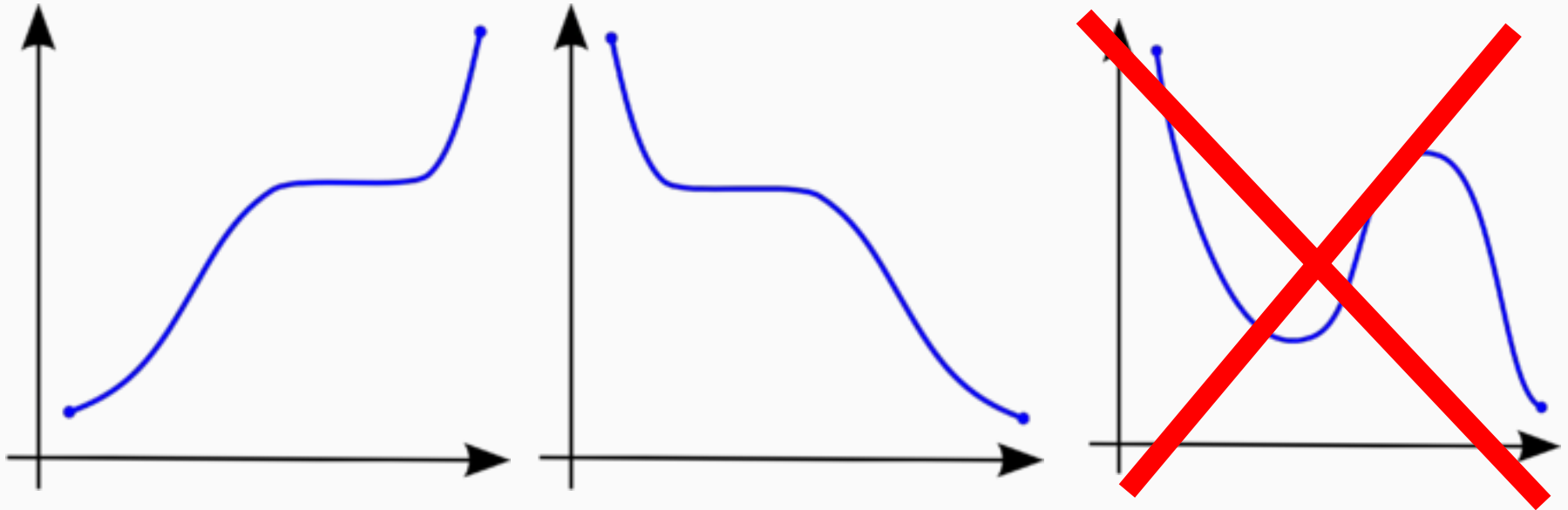
Binary Search

| interval | size | Q | A |
|---|------|--------|-----|
|  | 128 | < 64 ? | no |
|  | 64 | < 96 ? | yes |
|  | 32 | < 80 ? | yes |
|  | 16 | < 72 ? | no |
|  | 8 | < 76 ? | no |
|  | 4 | < 78 ? | yes |
|  | 2 | < 77 ? | no |
|  | 1 | = 77 | |

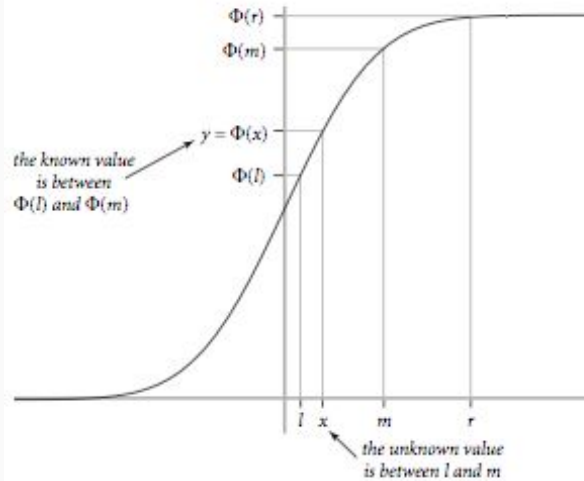
Finding a hidden number with binary search



Binary Search



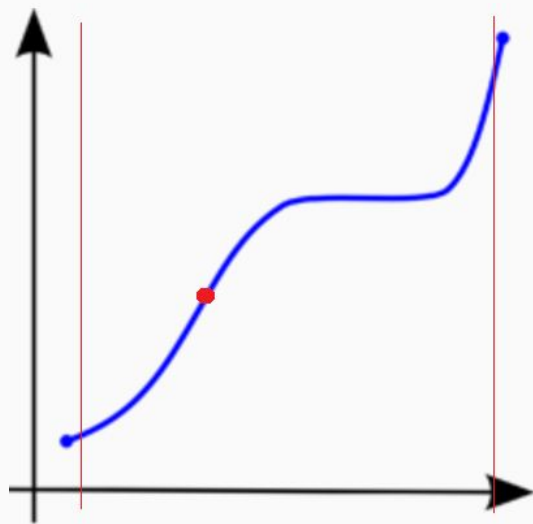
Binary Search



Binary search (bisection) to invert an increasing function (one step)

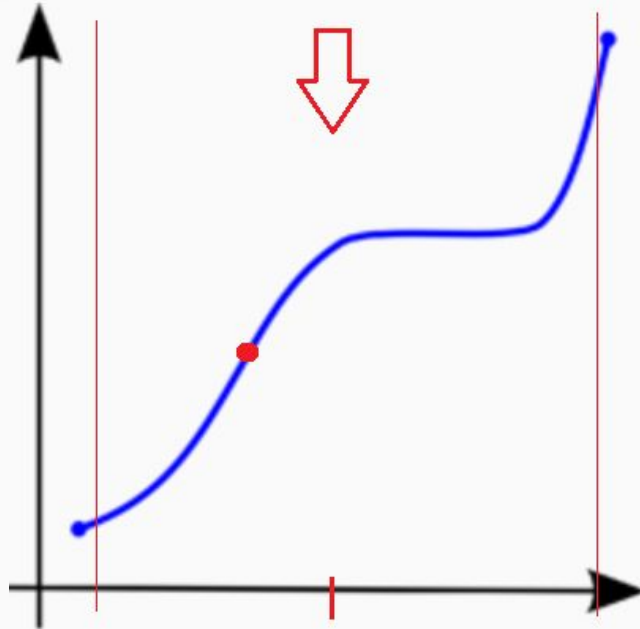
Example

Step 1: Pick left and right. Make sure this is both large enough and valid.



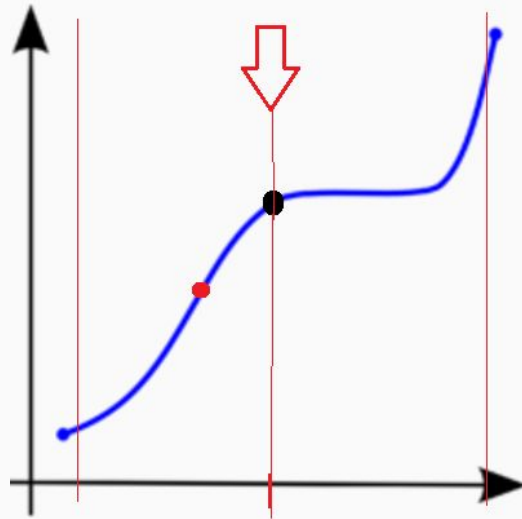
Example

Step 2: Pick middle



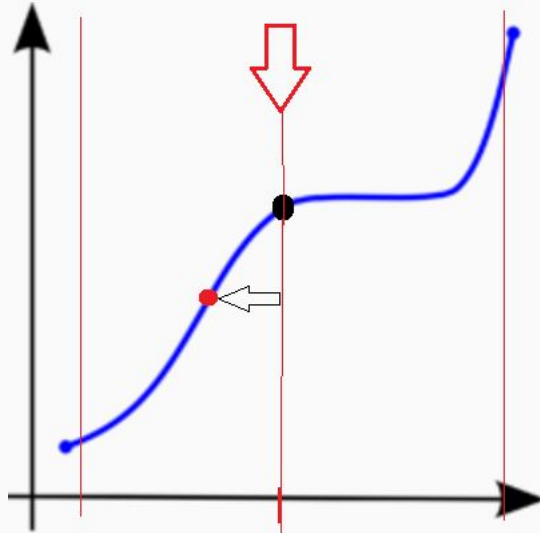
Example

Step 3: Evaluate function at middle



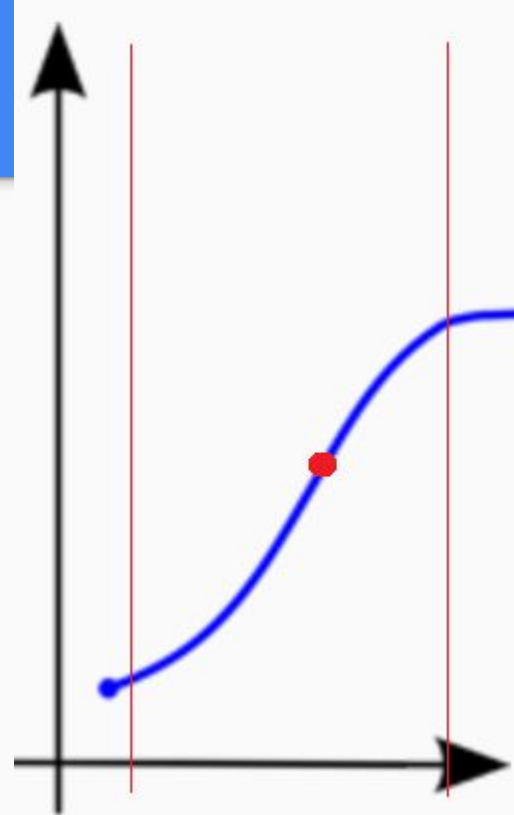
Example

Step 4: Is the answer on the left or the right (or did we find it)?



Step 5: Set $\text{right} = \text{mid}$, or $\text{left} = \text{mid}^*$.

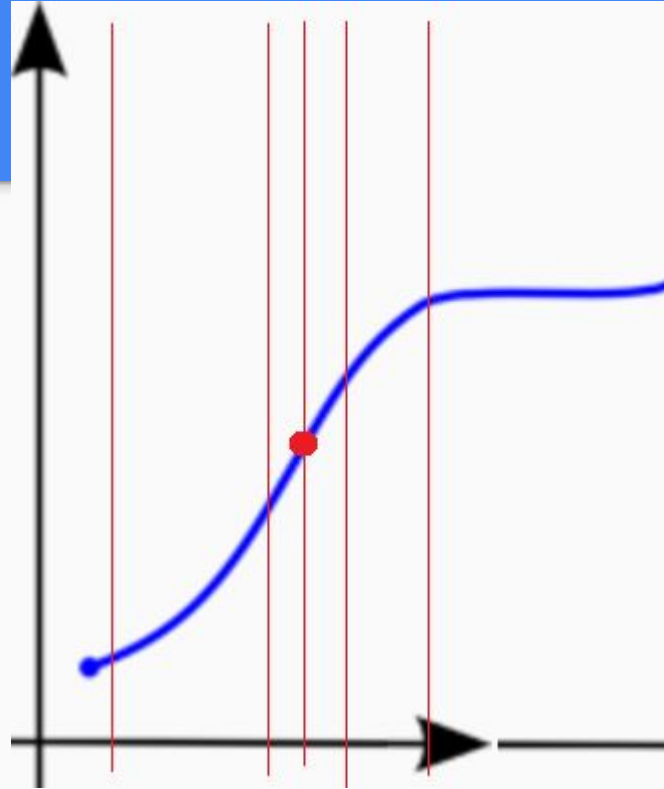
*Slightly different for integer valued search



Example

Step 6: Repeat

Step 7: Make sure to stop



Binary Search over Integers

Code from java.util.Arrays.binarySearch:

```
//Binary Search over array indices
private static int binarySearch(int[] a, int fromIndex, int toIndex, int key) {
    int low = fromIndex;
    int high = toIndex - 1;
    while (low <= high) {
        int mid = (low + high) >>> 1; // avoid overflow
        int midVal = a[mid];
        if (midVal < key)
            low = mid + 1;
        else if (midVal > key)
            high = mid - 1;
        else
            return mid; // key found
    }
    return -(low + 1); // key not found.
}
```

Binary Search over doubles (Handbook)

```
/**
 * Return x in [a, b] such that f(x) = y.
 * f() must be monotonic.
 * Also known as bisection method.
 */
static double binarySearch(double low, double high, double y) {
    while ((high - low) > 10 * Math.ulp(high)) {
        double mid = (low + high)/2.0;
        double midVal = f(mid);
        if (midVal < y) // or <=, see below.
            low = mid;
        else
            high = mid;
    }
    return (low + high)/2.0;
}
```

Complexity:

$\log(n) * O(\text{Evaluation Function})$

If you have an $O(1)$ function, it is $O(\log(n))$

If you have an $O(n)$ function, it is $O(n * \log(n))$



```
double EPS = .000000000001;
while ((high - low) > EPS){
    double mid = (low + high)/2.0;
    double midVal = f(mid);
    if (midVal < y)      // or <=, see below.
        low = mid;
    else
        high = mid;
}
/*****
(1000000000.00000000000 + 1000000000.00000000002)/2.0
1000000000.00000000001
1000000000.00000000000
```

This may never end.
Math.ulp(1000000000.00000000000) -> ~1.9073486328125E-6

As much as we love hardcoding random constants, **use ULP.**
*/

Example of Peter messing it up!

| | | | | | | | |
|--------------------------|---------------------|--------------|---|--------|----------------------------------|---------|----------|
| 16818673 | 2016-03-19 15:22:25 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16818658 | 2016-03-19 15:21:42 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20700 KB |
| 16818592 | 2016-03-19 15:18:41 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817947 | 2016-03-19 15:13:03 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20700 KB |
| 16817884 | 2016-03-19 15:10:57 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817540 | 2016-03-19 14:57:06 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817506 | 2016-03-19 14:55:47 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 7 | 2000 ms | 20600 KB |
| 16817483 | 2016-03-19 14:55:14 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 7 | 2000 ms | 20600 KB |
| 16817462 | 2016-03-19 14:54:12 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 1 | 2000 ms | 20400 KB |
| 16812759 | 2016-03-19 13:01:58 | PeterASteele | 653C - Bear and Up-Down | Java 8 | Accepted | 451 ms | 24600 KB |
| 16810600 | 2016-03-19 12:25:10 | PeterASteele | 653D - Delivery Bears | Java 8 | Wrong answer on test 44 | 187 ms | 20600 KB |
| 16810292 | 2016-03-19 12:20:16 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on pretest 7 | 2000 ms | 20600 KB |
| 16810239 | 2016-03-19 12:19:17 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on pretest 4 | 2000 ms | 20500 KB |

```
// binarySearch for the weight each bear can carry
double lowerBound = 0.0;
double upperBound = 100000000.0;
while (upperBound - lowerBound > 1E-15) {
    double test = (lowerBound + upperBound) / 2;
```

Example of Peter messing it up!

| | | | | | | | |
|--------------------------|---------------------|--------------|---|--------|----------------------------------|---------|----------|
| 16818673 | 2016-03-19 15:22:25 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16818658 | 2016-03-19 15:21:42 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20700 KB |
| 16818592 | 2016-03-19 15:18:41 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817947 | 2016-03-19 15:13:03 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20700 KB |
| 16817884 | 2016-03-19 15:10:57 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817540 | 2016-03-19 14:57:06 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817506 | 2016-03-19 14:55:47 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 7 | 2000 ms | 20600 KB |
| 16817483 | 2016-03-19 14:55:14 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 7 | 2000 ms | 20600 KB |
| 16817462 | 2016-03-19 14:54:12 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 1 | 2000 ms | 20400 KB |
| 16812759 | 2016-03-19 13:01:58 | PeterASteele | 653C - Bear and Up-Down | Java 8 | Accepted | 451 ms | 24600 KB |
| 16810600 | 2016-03-19 12:25:10 | PeterASteele | 653D - Delivery Bears | Java 8 | Wrong answer on test 44 | 187 ms | 20600 KB |
| 16810292 | 2016-03-19 12:20:16 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on pretest 7 | 2000 ms | 20600 KB |
| 16810239 | 2016-03-19 12:19:17 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on pretest 4 | 2000 ms | 20500 KB |

```
// binarySearch for the weight each bear can carry
double lowerBound = 0.0;
double upperBound = 1000000.0;
while (upperBound - lowerBound > 1E-13) {
    double test = (lowerBound + upperBound) / 2;
```


Example of Peter messing it up!

| | | | | | | | |
|--------------------------|---------------------|--------------|---|--------|----------------------------------|---------|----------|
| 16818673 | 2016-03-19 15:22:25 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16818658 | 2016-03-19 15:21:42 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20700 KB |
| 16818592 | 2016-03-19 15:18:41 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817947 | 2016-03-19 15:13:03 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20700 KB |
| 16817884 | 2016-03-19 15:10:57 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817540 | 2016-03-19 14:57:06 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817506 | 2016-03-19 14:55:47 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 7 | 2000 ms | 20600 KB |
| 16817483 | 2016-03-19 14:55:14 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 7 | 2000 ms | 20600 KB |
| 16817462 | 2016-03-19 14:54:12 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 1 | 2000 ms | 20400 KB |
| 16812759 | 2016-03-19 13:01:58 | PeterASteele | 653C - Bear and Up-Down | Java 8 | Accepted | 451 ms | 24600 KB |
| 16810600 | 2016-03-19 12:25:10 | PeterASteele | 653D - Delivery Bears | Java 8 | Wrong answer on test 44 | 187 ms | 20600 KB |
| 16810292 | 2016-03-19 12:20:16 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on pretest 7 | 2000 ms | 20600 KB |
| 16810239 | 2016-03-19 12:19:17 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on pretest 4 | 2000 ms | 20500 KB |

```
double lowerBound = 0.0;
double upperBound = 1000000.0;
while (upperBound - lowerBound > 1E-9) {
    double test = (lowerBound + upperBound) /
```

Example of Peter messing it up!

| | | | | | | | |
|--------------------------|---------------------|----------------|---|--------|----------------------------------|---------|----------|
| 16818673 | 2016-03-19 15:22:25 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16818658 | 2016-03-19 15:21:42 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20700 KB |
| 16818592 | 2016-03-19 15:18:41 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817947 | 2016-03-19 15:13:03 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20700 KB |
| 16817884 | 2016-03-19 15:10:57 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817540 | 2016-03-19 14:57:06 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817506 | 2016-03-19 14:55:47 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 7 | 2000 ms | 20600 KB |
| 16817483 | 2016-03-19 14:55:14 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 7 | 2000 ms | 20600 KB |
| 16817462 | 2016-03-19 14:54:12 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 1 | 2000 ms | 20400 KB |
| 16812759 | 2016-03-19 13:01:58 | PeterAS Steele | 653C - Bear and Up-Down | Java 8 | Accepted | 451 ms | 24600 KB |
| 16810600 | 2016-03-19 12:25:10 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Wrong answer on test 44 | 187 ms | 20600 KB |
| 16810292 | 2016-03-19 12:20:16 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on pretest 7 | 2000 ms | 20600 KB |
| 16810239 | 2016-03-19 12:19:17 | PeterAS Steele | 653D - Delivery Bears | Java 8 | Time limit exceeded on pretest 4 | 2000 ms | 20500 KB |

```
// binarySearch for the weight each bear can c
double lowerBound = 0.0;
double upperBound = 1000000.0;
while (upperBound - lowerBound > 1E-16) {
    double test = (lowerBound + upperBound
```

Example of Peter messing it up!



| | | | | | | | |
|--------------------------|---------------------|--------------|---|--------|----------------------------------|---------|----------|
| 16818673 | 2016-03-19 15:22:25 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16818658 | 2016-03-19 15:21:42 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20700 KB |
| 16818592 | 2016-03-19 15:18:41 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817947 | 2016-03-19 15:13:03 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20700 KB |
| 16817884 | 2016-03-19 15:10:57 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817540 | 2016-03-19 14:57:06 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 16 | 2000 ms | 20600 KB |
| 16817506 | 2016-03-19 14:55:47 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 7 | 2000 ms | 20600 KB |
| 16817483 | 2016-03-19 14:55:14 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 7 | 2000 ms | 20600 KB |
| 16817462 | 2016-03-19 14:54:12 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on test 1 | 2000 ms | 20400 KB |
| 16812759 | 2016-03-19 13:01:58 | PeterASteele | 653C - Bear and Up-Down | Java 8 | Accepted | 451 ms | 24600 KB |
| 16810600 | 2016-03-19 12:25:10 | PeterASteele | 653D - Delivery Bears | Java 8 | Wrong answer on test 44 | 187 ms | 20600 KB |
| 16810292 | 2016-03-19 12:20:16 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on pretest 7 | 2000 ms | 20600 KB |
| 16810239 | 2016-03-19 12:19:17 | PeterASteele | 653D - Delivery Bears | Java 8 | Time limit exceeded on pretest 4 | 2000 ms | 20500 KB |

```
// binarySearch for the weight each bear
double lowerBound = 0.0;
double upperBound = 1000000.0;
while (upperBound - lowerBound > 1E-13)
    double test = (lowerBound + upperBound) / 2;
```

You get the idea

| | | | | | | | |
|--------------------------|------------------------|--------------|---------------------------------------|-----------|----------|--------|----------|
| 16821549 | 2016-03-19 17:38:13 | PeterASteele | 653D - Delivery Bears | Java 8 | Accepted | 171 ms | 20800 KB |
|--------------------------|------------------------|--------------|---------------------------------------|-----------|----------|--------|----------|

Fix:

```
// binarySearch for the weight each bear can carry.
double lowerBound = 0.0;
double upperBound = 1000000.0;
double maxFlow = 100;
while (upperBound - lowerBound > (lowerBound + (upperBound-lowerBound)*.5)/10E9) {
    double test = lowerBound + (upperBound-lowerBound)*.5;
```

When can you use binary search?

Use binary search when it is possible to tell if an answer is too high or too low.

It's OK if this process is $O(n)$; Runtime will still be $O(n \cdot \log(n))$, which may be good enough.

Typical binary search type question statements:

Find the lowest value x such that $\text{STATEMENT}(X)$ is true.

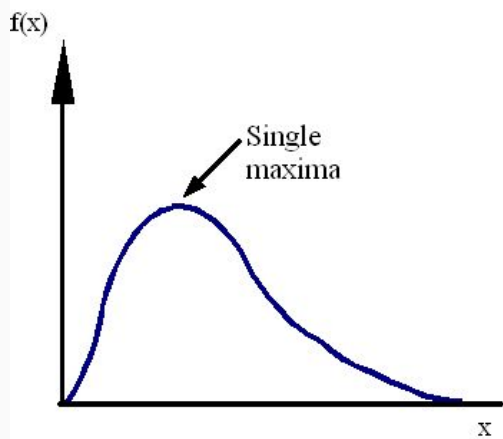
Find the highest value x such that $\text{STATEMENT}(X)$ is true.

What's the smallest number of partitions such that a list can be divided into y parts meeting a certia?

How do I find it?

If you were given a potential answer, could you check if it was correct? Too high? Too low? If the answer is yes, consider binary search.

Ternary Search



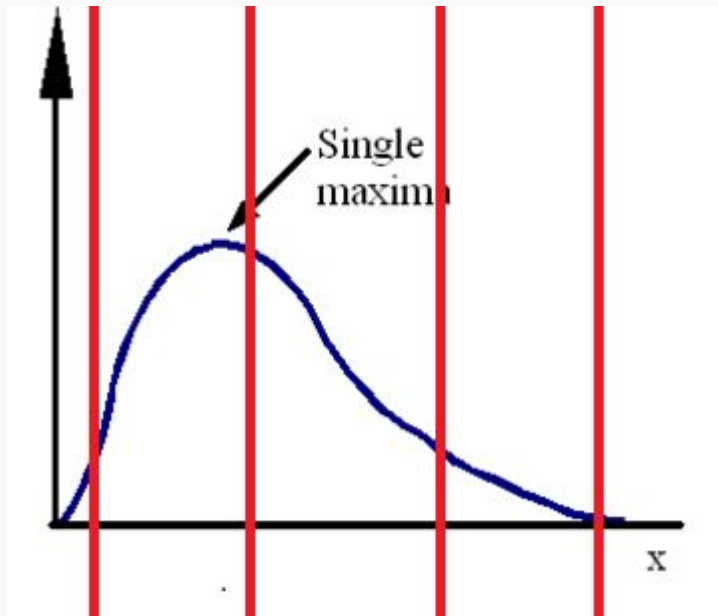
Must be UNIMODAL.

Maximum one change in sign.

Process

Step 1: Pick a left and right.

Step 2: Calculate thirds

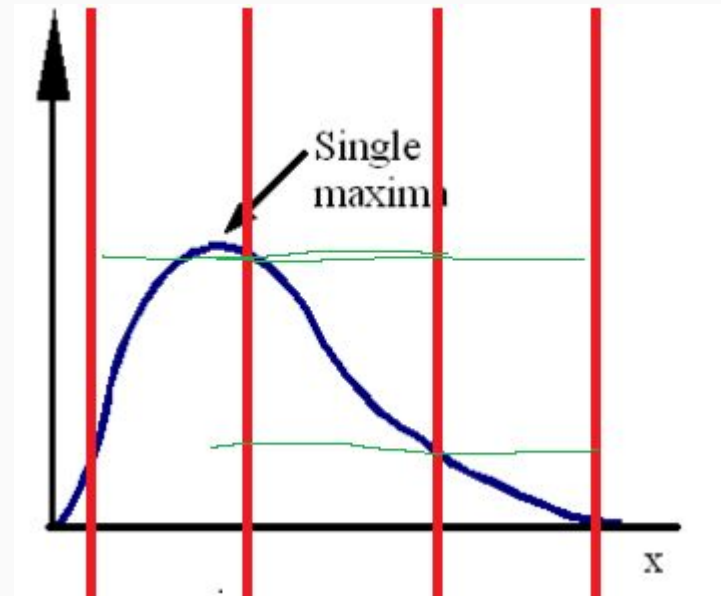


Step 3: Look at the left and right height.

If left is higher than right, the answer CANNOT be in the far right third.

If right is higher than left, the answer CANNOT be in the far left third.

Each iteration removes 1/3rd; $\log(n)$ iterations.

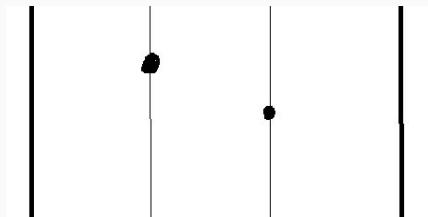


Step 4: Repeat until precision. Use ULP like we did in BinarySearch to stop.

Why does this work?

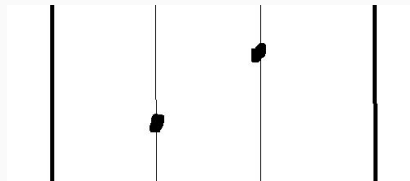
There are two cases:

Case 1: $f(\text{left}) > f(\text{right})$



We can safely remove right, because if the max was on the right, there would be 2 sign changes.

Case 2: $f(\text{left}) < f(\text{right})$



We can safely remove right, because if the max was on the right, there would be 2 sign changes.

Problems!

Binary Search:

- <https://pcs.spruett.me/problems/32>
- <https://pcs.spruett.me/problems/16>
- <https://open.kattis.com/contests/naipc16-p10/problems/primal>
- <http://codeforces.com/problemset/problem/653/D> (Hard)

Ternary Search

- <http://codeforces.com/problemset/problem/613/A>
- <https://open.kattis.com/contests/naipc16-p08/problems/reconnaissance>