



WIKIPEDIA
The Free Encyclopedia

[Main page](#)

[Contents](#)

[Featured content](#)

[Current events](#)

[Random article](#)

[Donate to Wikipedia](#)

[Wikipedia store](#)

Interaction

[Help](#)

[About Wikipedia](#)

[Community portal](#)

[Recent changes](#)

[Contact page](#)

Tools

[What links here](#)

[Related changes](#)

[Upload file](#)

[Special pages](#)

[Permanent link](#)

[Page information](#)

[Wikidata item](#)

[Cite this page](#)

Print/export

[Create a book](#)

[Download as PDF](#)

[Printable version](#)

Languages

[فارسی](#)

[ไทย](#)

 [Edit links](#)

Article [Talk](#)

[Read](#)

[Edit](#)

[View history](#)



Uniform binary search

From Wikipedia, the free encyclopedia

Uniform binary search is an optimization of the classic [binary search](#) algorithm invented by [Donald Knuth](#) and given in Knuth's *[The Art of Computer Programming](#)*. It uses a [lookup table](#) to update a single array index, rather than taking the midpoint of an upper and a lower bound on each iteration; therefore, it is optimized for architectures (such as Knuth's [MIX](#)) on which

- a table lookup is generally faster than an addition and a shift, and
- many searches will be performed on the same array, or on several arrays of the same length

C implementation [\[edit\]](#)

The uniform [binary search algorithm](#) looks like this, when implemented in [C](#).

```
#define LOG_N 4

static int delta[LOG_N];

void make_delta(int N)
{
    int power = 1;
    int i = 0;
    do {
        int half = power;
        power <= 1;
        delta[i] = (N + half) / power;
    } while (delta[i++] != 0);
}

int unisearch(int *a, int key)
{
    int i = delta[0]-1; /* midpoint of array */
    int d = 0;

    while (1) {
        if (key == a[i]) {
            return i;
        } else if (delta[d] == 0) {
            return -1;
        } else {
            if (key < a[i]) {
                i -= delta[++d];
            } else {
                i += delta[++d];
            }
        }
    }
}

/* Example of use: */
#define N 10
int main(void)
{
    int i, a[N] = {1,3,5,6,7,9,14,15,17,19};
    make_delta(N);
    for (i=0; i < 20; ++i)
        printf("%d is at index %d\n", i, unisearch(a, i));
    return 0;
}
```

References [\[edit\]](#)

- Knuth. *The Art of Computer Programming*, Volume 3. Page 412, Algorithm C.

External links [\[edit\]](#)

- [An implementation of Knuth's algorithm](#) in [Pascal](#), by Han de Bruijn

Categories: [Search algorithms](#)

This page was last modified on 8 October 2014, at 00:20.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

