Meta: strategies

Jill-Jênn Vie

Sep 13, 2024

Finding the minimum x such that $f(x)$ is `true` (f monotonic)

```
// dernier 0 de predicat[l..r[ en supposant croissance
if (predicat(l))
        return l - 1;
while (r - l > 1) {
        int m = (l + r) / 2;
        if (predicat(m))
                r = m;
        else
                l = m;
}
return l;
```

Application: binary search on the answer

Finding the minimum x such that $v[x] \geq$ `val` (v sorted array)

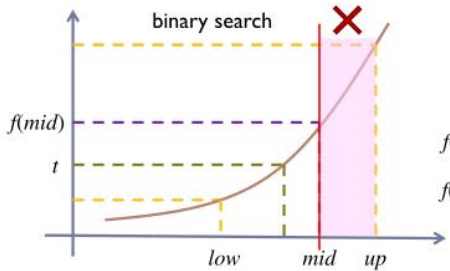
```
std::sort(v.begin(), v.end());
// First element in [first, last) which is not less than val
low = std::lower_bound(v.begin(), v.end(), val);
low - v.begin();

// First element in [first, last) which is greater than val
up = std::upper_bound(v.begin(), v.end(), val);
up - v.begin();
```

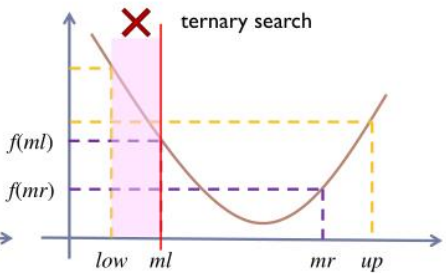
Example for `val` $= 20$

```
          | low = 3
10 10 10 20 20 20 30 30
                  | up = 6
```

Finding the Minimum

- 給予 convex function f，並已知 f 在 (low, up) 之間出現最小值，求值為何
- 像在具有單調性的函數上 binary search 的方式，不斷丟棄不可能存在答案的區間來逼進答案
  - ternary search 也是 divide and conquer





Finding the minimum of a convex function

```
//strictly increasing then strictly decreasing function f
double ternary_search(double l, double r) {
    double eps = 1e-9; //set the error limit here
    while (r - l > eps) {
        double m1 = l + (r - l) / 3;
        double m2 = r - (r - l) / 3;
        double f1 = f(m1); //evaluates the function at m1
        double f2 = f(m2)  //evaluates the function at m2
        if (f1 > f2)
            l = m1;
        else
            r = m2;
    }
    return f(l); //return the maximum of f(x) in [l, r]
}
```

Enumeration

- All subsets of $\{1, \ldots, n\}$
- All subsets of a subset
- All permutations
- All combinations
- All combinations such that adjacent combinations differ by one element
- All ways to put n identical objects into k labeled boxes
- All sets of positive integers that sum to N
- Backtracking: choose / explore / unchoose

References

- Competitive Programmer's Handbook book by Antti Laaksonen
- Stanford course
- algo-en by Donglai Fu