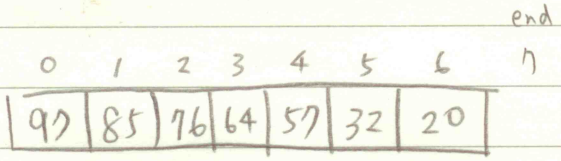
* Binary Search

pick i ← $\lfloor \frac{left + right}{2} \rfloor$ (usually named mid)



note: efficiency relies on "fast" computation of arr[mid]

search (57)

sequential : after 5 iterations

binary : after 3 iterations

* extendable array

- Fixed Len Array (and others) : will be "full" some time
- doesn't matter if you only want, say, top-20 records
- what if storing all record histories ?

Extendable Array [Consecutive]

"insert" grows if end = len

insert (value)

if end = len

allocate new space bigger than len

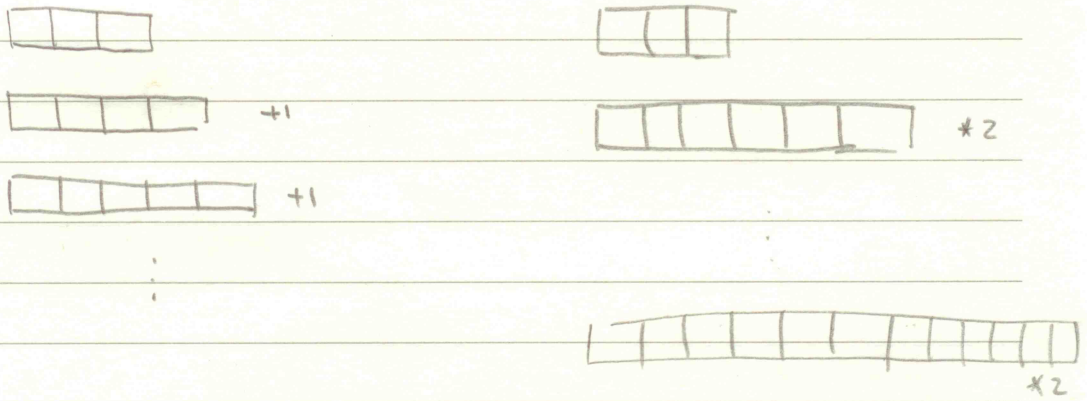
copy original arr[0] ~ arr[end-1] to new space

deallocate old space at arr

link new space to arr

ConsecutiveArray.insert(value)

* "bigger than" ?



"often" needed

"less often", but "costly" ?

vector in Standard Template Library : learn to use it !

```
std::vector<int>  earr;      // # include <vector>
```

std : name space (1.1.4)

vector<int> : template class (2.3)

```
namespace  std {    // 家族
    template <class T>
    class  vector {
        // ... by "T"
    }
}
```