第六讲作业

1 上下界包含的二分查找

```
1  int binsearch_inclusive(int x, int[ ] A, int n)
2  //@requires 0 <= n && n <= \length(A);
3  //@requires is_sorted(A, 0, n);
4  /*@ensures (-1 == \result && !is_in(x, A, 0, n))
5        || ((0 <= \result && \result < n) && A[\result] == x);
6    @*/
7  {
8      int lower = 0;
9      int upper = n - 1;
10     while (lower <= upper)
11     //@loop_invariant 0 <= lower && lower <= upper + 1 && upper < n;
12     //@loop_invariant (lower == 0 || A[lower - 1] < x);
13     //@loop_invariant (upper == n - 1 || A[upper + 1] > x);
14     {
15         int mid = lower + (upper - lower) / 2;
16         //@assert lower <= mid && mid <= upper;
17         if (A[mid] == x) return mid;
18         else if (A[mid] < x) lower = mid + 1;
19         else /*@assert(A[mid] > x);@*/
20             upper = mid - 1;
21     }
22     //@assert upper + 1 == lower;
23     return -1;
24 }
```

证明：

- 第 16 行 $lower \leq mid \leq upper$：用第 10, 11, 15 行和数学知识可证，由此可推出 $0 \leq mid < n$，下标合法
- 循环不变量
  - 初始情况：$lower = 0,\ upper = n - 1 \geq -1$，（由第 2, 8, 9 行推出），成立
  - 假设循环不变量成立，证明下次循环时仍成立：

$$0 \leq lower \leq upper + 1 < n$$
$$lower \leq mid \leq upper$$

1. `A[mid] == x`：不进行判断，退出
2. `A[mid] < x`：

$$\begin{aligned}
&lower' = mid + 1 \\
&A[lower' - 1] < x\ (Line\ 12\ 成立) \\
&upper' = upper\ (Line\ 13\ 成立) \\
&lower < mid + 1 \leq upper + 1 \\
&0 \leq lower < lower' \leq upper' + 1 < n\ (Line\ 11\ 成立)
\end{aligned}$$

3. `A[mid] > x`：

$$\begin{aligned}
&upper' = mid - 1 \\
&A[upper' + 1] > x\ (Line\ 13\ 成立) \\
&lower' = lower\ (Line\ 12\ 成立) \\
&lower - 1 \leq mid - 1 \\
&0 \leq lower' \leq upper' + 1 \leq upper < n\ (Line\ 11\ 成立)
\end{aligned}$$

综上，循环不变量成立

- 循环可终止证明：

  记 $length = upper - lower + 1$，则

  - 若$upper' = mid - 1$, $lower' = lower$，那么
    $length' = upper' - lower' + 1 = mid - lower < mid - lower + 1 \leq upper - lower + 1 = length$
  - 若$upper' = upper$, $lower' = mid + 1$，那么
    $length' = upper' - lower' + 1 = upper - mid < upper - mid + 1 \leq upper - lower + 1 = length$

  所以 $length$ 严格单调递减，当 $length < 1$ 时循环终止

- 后置条件证明：

  由于 `is_sorted(A,0,n)`，那么 $A[0, lower) < x$，$A[upper + 1, n) > x$，当循环结束时，$lower > upper$，因此 $upper + 1 = lower$，$x \notin A[0, n)$

2 `is_in` 函数作循环不变量

```
1  int search(int x, int[] A, int n)
2  //@requires n == \length(A);
3  //@requires is_sorted(A, 0, n);
4  /*@ensures (\result == -1 && !is_in(x, A, 0, n))
5          || (0 <= \result && \result < n && A[\result] == x); @*/
6  {
7      int lower = 0;
8      int upper = n;
9      while (lower < upper)
10     //@loop_invariant 0 <= lower && lower <= upper && upper <= n;
11     //@loop_invariant !is_in(x, A, 0, lower);
12     //@loop_invariant !is_in(x, A, upper, n);
13     {
14         int mid = lower + (upper - lower)/2;
15         //@assert lower <= mid && mid < upper;
16         if (A[mid] == x) return mid;
17         if (A[mid] < x) {
18             //@assert mid + 1 <= upper;
19             //@assert !is_in(x, A, 0, mid + 1);
20             lower = mid + 1;
21         } else { //@assert A[mid] > x;
22             //@assert !is_in(x, A, mid, n);
23             upper = mid;
24         }
25     }
26     //@assert lower == upper;
27     //@assert !is_in(x, A, 0, n);
28     return -1;
29 }
```

证明：

- 第 15 行 $lower \leq mid < upper$：用第 9, 10, 14 行和数学知识可证，由此可推出 $0 \leq mid < n$，下标合法
- 第 18 行 `mid + 1 <= upper`，由 $mid < upper$ 推出
- 第 19 行，由于 `is_sorted(A, 0, n)`，A单调上升，`A[mid] < x`，则 $A[0, mid + 1) < x$，所以 `!is_in(x, A, 0, mid + 1)`
- 第 22 行，由于 `is_sorted(A, 0, n)`，A单调上升，`A[mid] > x`，则 $A[mid, n) > x$，所以 `!is_in(x, A, mid, n)`

- 循环不变量
  - 初始情况：$lower = 0$, $upper = n$，成立
  - 假设循环不变量成立，证明下次循环时仍成立：

$$0 \le lower \le upper \le n$$
$$lower \le mid < upper$$

    1. `A[mid] == x`：不进行判断，退出
    2. `A[mid] < x`：仅 `lower` 改变，$lower' = mid + 1$，由第 18 行 `mid + 1 <= upper` 可知，第 10 行成立，由第 19 行 `!is_in(x, A, 0, mid + 1)` 可知，第 11 行成立
    3. `A[mid] > x`：仅 `upper` 改变，$upper' = mid$，由第 15 行 $lower \le mid < upper$ 可知，第 10 行成立，由第 22 行 `!is_in(x, A, mid, n)` 可知，第 12 行成立

  综上，循环不变量成立
- 循环可终止证明：

  记 $length = upper - lower$，则
  - 若$upper' = mid$, $lower' = lower$，那么
    $length' = upper' - lower' = mid - lower < upper - lower = length$
  - 若$upper' = upper$, $lower' = mid + 1$，那么
    $length' = upper' - lower' = upper - mid - 1 \le upper - lower - 1 < upper - lower = length$

  所以 $length$ 严格单调递减，当 $length \le 0$ 时循环终止
- 第 26 行，由第 9, 10 行可知
- 后置条件证明：

  当 `\result != -1` 时是显然的。

  当 `\result == -1` 时，由第 11, 12, 26 行，`!is_in(x, A, 0, lower) && !is_in(x, A, upper, n) && lower == upper` 可以推出 `!is_in(x, A, 0, n)`

3 查找最左侧的x

- 例子：长度为5，全为1的数组，查找1，会返回2，而非0

```
1  int search(int x, int[] A, int n)
2  //@requires n == \length(A);
3  //@requires is_sorted(A, 0, n);
4  /*@ensures (\result == -1 && !is_in(x, A, 0, n))
5          || (0 <= \result && \result < n && A[\result] == x && !is_in(x, A, 0,
   \result)); @*/
6  {
7      int lower = 0;
8      int upper = n;
9      while (lower < upper)
10     //@loop_invariant 0 <= lower && lower <= upper && upper <= n;
11     //@loop_invariant (lower == 0 || A[lower - 1] < x);
12     //@loop_invariant (upper == n || A[upper] >= x);
13     {
14         int mid = lower + (upper - lower)/2;
15         //@assert lower <= mid && mid < upper;
16         if (A[mid] < x) lower = mid + 1;
17         else /*@assert A[mid] >= x; @*/
18             upper = mid;
19     }
20     //@assert lower == upper;
21     if (lower < n && A[lower] == x)
```

```
22          return lower;
23      return -1;
24  }
```

证明：

- 第 15 行 $lower \le mid < upper$：用第 9, 10, 14 行和数学知识可证，由此可推出 $0 \le mid < n$，下标合法
- 循环不变量
  - 初始情况：$lower = 0,\ upper = n$，成立
  - 假设循环不变量成立，证明下次循环时仍成立：

$$0 \le lower \le upper \le n$$
$$lower \le mid < upper$$

1. `A[mid] < x`：

$$\begin{aligned}
&lower' = mid + 1 \\
&A[lower' - 1] < x\ (Line\ 11\ 成立) \\
&upper' = upper\ (Line\ 12\ 成立) \\
&mid + 1 \le upper \\
&0 \le lower' = mid + 1 \le upper = upper' \le n\ (Line\ 10\ 成立)
\end{aligned}$$

2. `A[mid] >= x`：

$$\begin{aligned}
&upper' = mid \\
&A[upper'] \ge x\ (Line\ 12\ 成立) \\
&lower' = lower\ (Line\ 11\ 成立) \\
&0 \le lower' = lower \le mid = upper' \le n\ (Line\ 10\ 成立)
\end{aligned}$$

  综上，循环不变量成立

- 循环可终止证明：

  记 $length = upper - lower$，则

  - 若$upper' = mid,\ lower' = lower$，那么
    $length' = upper' - lower' = mid - lower < upper - lower = length$
  - 若$upper' = upper,\ lower' = mid + 1$，那么
    $length' = upper' - lower' = upper - mid - 1 \le upper - lower - 1 < upper - lower = length$

  所以 $length$ 严格单调递减，当 $length \le 0$ 时循环终止

- 第 20 行，由第 9, 10 行可知
- 后置条件证明：

  由于 `is_sorted(A,0,n)`，结合循环不变量，那么 $A[0, lower) < x$，$A[upper, n) \ge x$。当循环结束时，$lower = upper$，因此只需要验证 `A[lower]` 是否为 `x` 即可，同时，此处需注意验证下标合法。

4 整型溢出

如果改为 `int mid = (lower + upper)/2;`，那么在计算 `lower + upper` 的时候可能会发生溢出，变为负数，最直接的是 `//@assert lower <= mid && mid < upper;` 会报错。而且，在访问 `A[mid]` 时，也会因为下标非法原因有异常产生。

5 死循环

- 证明循环不变量
  - 初始情况显然满足
  - 归纳的时候，由于 `lower` 和 `upper` 均未改变，所以在新一次循环中依然满足
- 假如初始时，$n > 0$，那么循环无法终止，循环终止条件 $lower \geq upper$ 不可能满足