02-D3

向量

有序向量：Fibonacci查找

常伟思微微一笑说：这个比例很奇怪，是吗？

他又想来想去，又想不出好地方，于是终于决心，假定这“幸福的家庭”所在的地方叫做A。

邓俊辉

deng@tsinghua.edu.cn

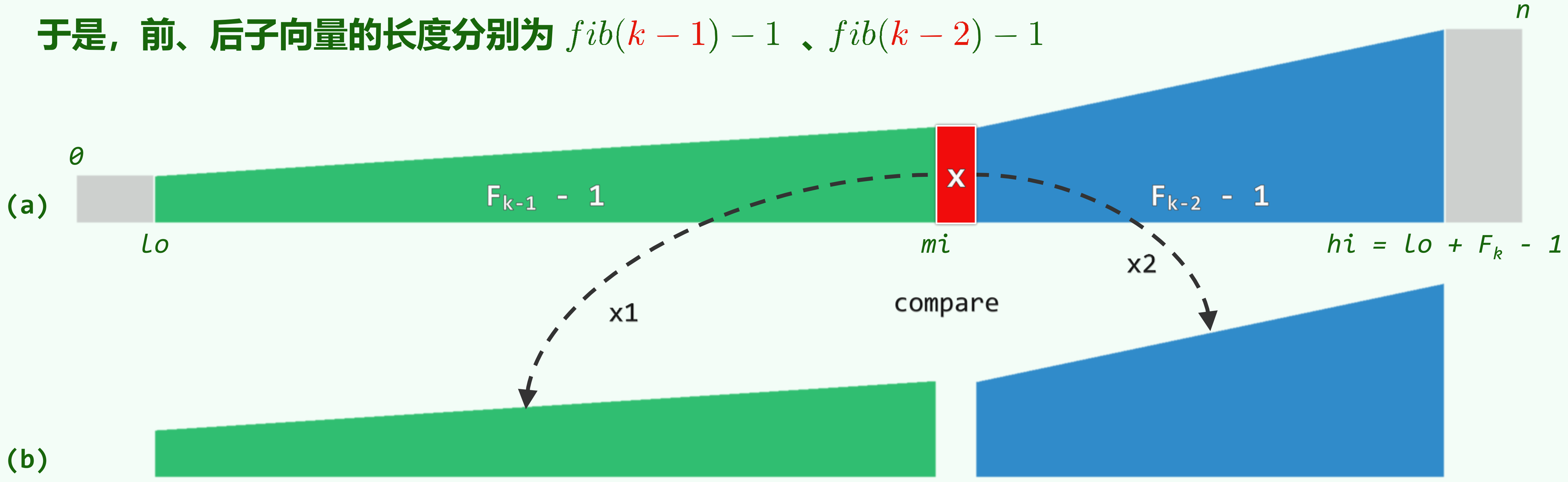
思路及原理

❖ 版本A：转向左、右分支前的关键码比较次数不等，而递归深度却相同

❖ 通过递归深度的不均衡对转向成本的不均衡做补偿，平均查找长度应能进一步缩短！

❖ 比如，若有 $n = fib(k) - 1$，则可取 $mi = fib(k-1) - 1$

于是，前、后子向量的长度分别为 $fib(k-1) - 1$ 、$fib(k-2) - 1$



实现

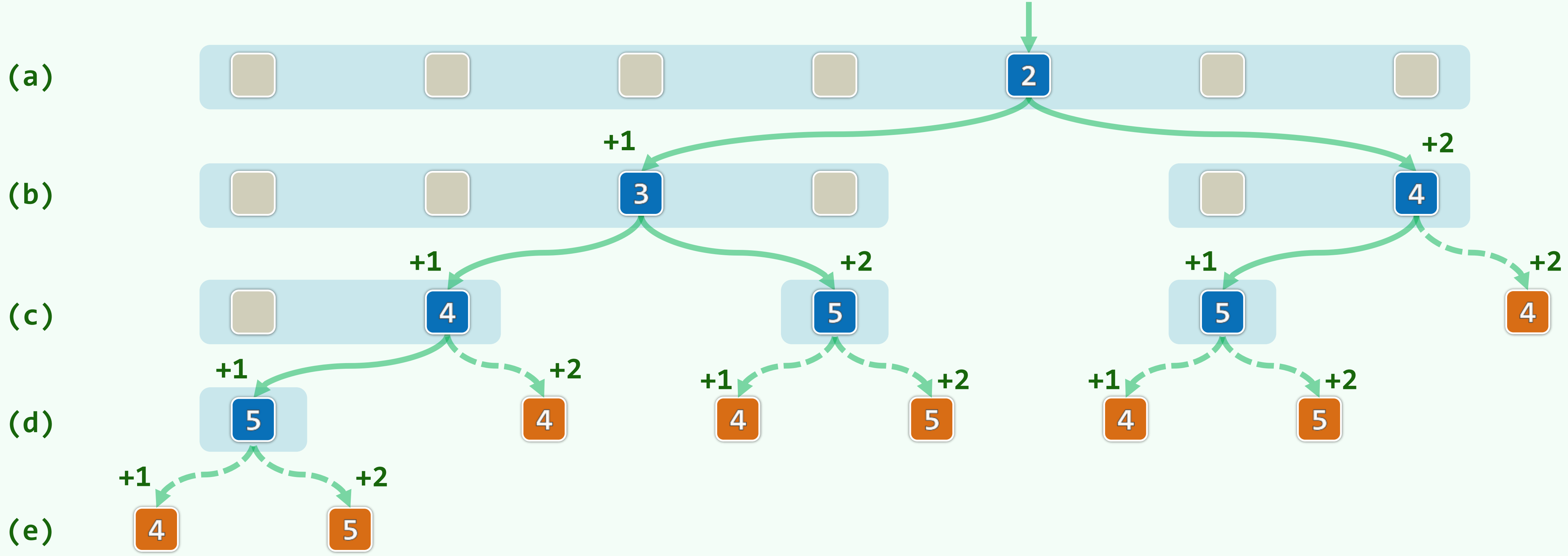
```
template <typename T> //0 <= lo <= hi <= _size
static Rank fibSearch( T * S, T const & e, Rank lo, Rank hi ) {
   for ( Fib fib(hi - lo); lo < hi; ) { //Fib数列制表备查
      while ( hi - lo < fib.get() ) fib.prev(); //自后向前顺序查找轴点（分摊O(1)）
      Rank mi = lo + fib.get() - 1; //确定形如Fib(k)-1的轴点
      if ( e < S[mi] ) hi = mi; //深入前半段[lo, mi)
      else if ( S[mi] < e ) lo = mi + 1; //深入后半段(mi, hi)
      else return mi; //命中
   }
   return -1; //失败
} //有多个命中元素时，不能保证返回秩最大者；失败时，简单地返回-1，而不能指示失败的位置
```

平均查找长度：常系数略优

❖ 仍以$n = fib(6) - 1 = 7$为例，在等概率情况下 //详见教材、习题解析

- $ASL_{succ} = (5 + 4 + 3 + 5 + 2 + 5 + 4) / 7 = 28/7 = 4.00$

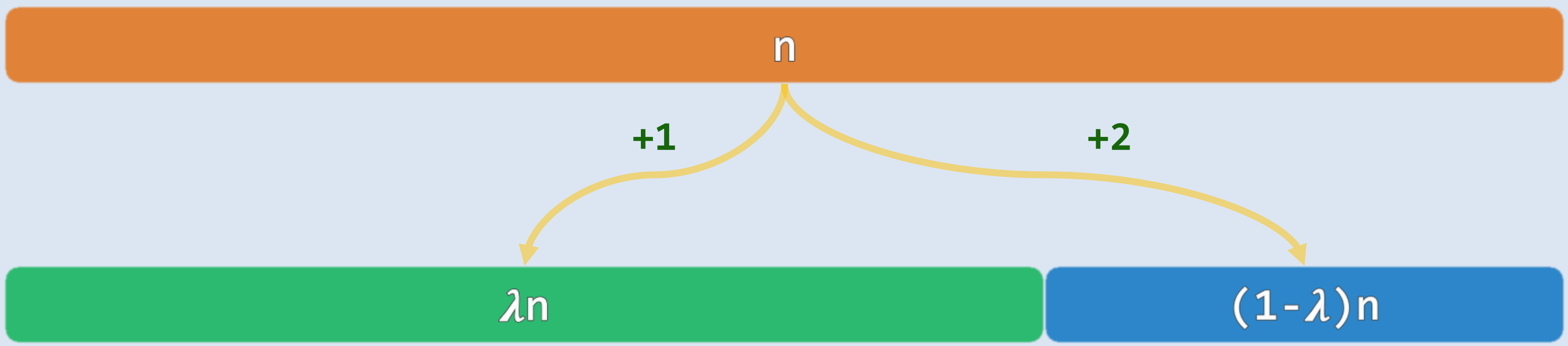
- $ASL_{fail} = (4 + 5 + 4 + 4 + 5 + 4 + 5 + 4) / 8 = 35 / 8 = 4.38$



通用策略

❖ 在任何区间 $[0, n)$ 内，总是选取 $[\lambda \cdot n]$ 作为轴点， $0 \le \lambda < 1$

比如：二分查找对应于 $\lambda = 0.5$ ，Fibonacci查找对应于 $\lambda = \phi = 0.6180339\ldots$

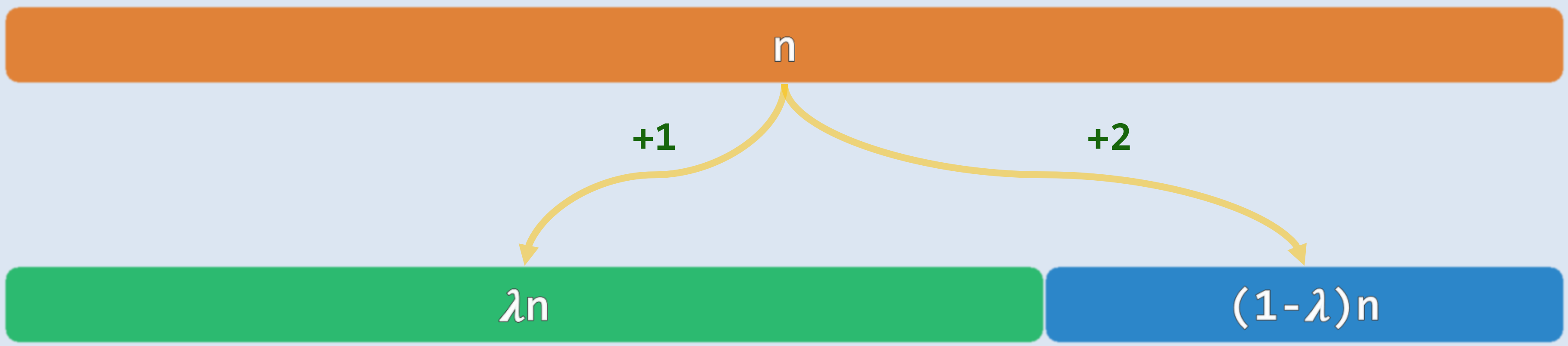


❖ 这类查找算法的渐近复杂度为 $\alpha(\lambda) \cdot \log_2 n \; = \; \mathcal{O}(\log n)$

❖ 常系数 $\alpha(\lambda)$ 何时达到最小...

ϕ = 0.6180339...

❖ **递推式：** $\alpha(\lambda) \cdot \log_2 n \;=\; \lambda \cdot [\underline{1 + \alpha(\lambda) \cdot \log_2 (\lambda n)}] \;+\; (1-\lambda) \cdot [\underline{2 + \alpha(\lambda) \cdot \log_2 ((1-\lambda)n)}]$



❖ **整理后：** $\dfrac{-\ln 2}{\alpha(\lambda)} = \dfrac{\lambda \cdot \ln \lambda + (1-\lambda) \cdot \ln(1-\lambda)}{2-\lambda}$

❖ **当** $\lambda = \phi = (\sqrt{5}-1)/2$ **时，** $\alpha(\lambda) = 1.440420\ldots$ **达到最小**