09-G3

词典

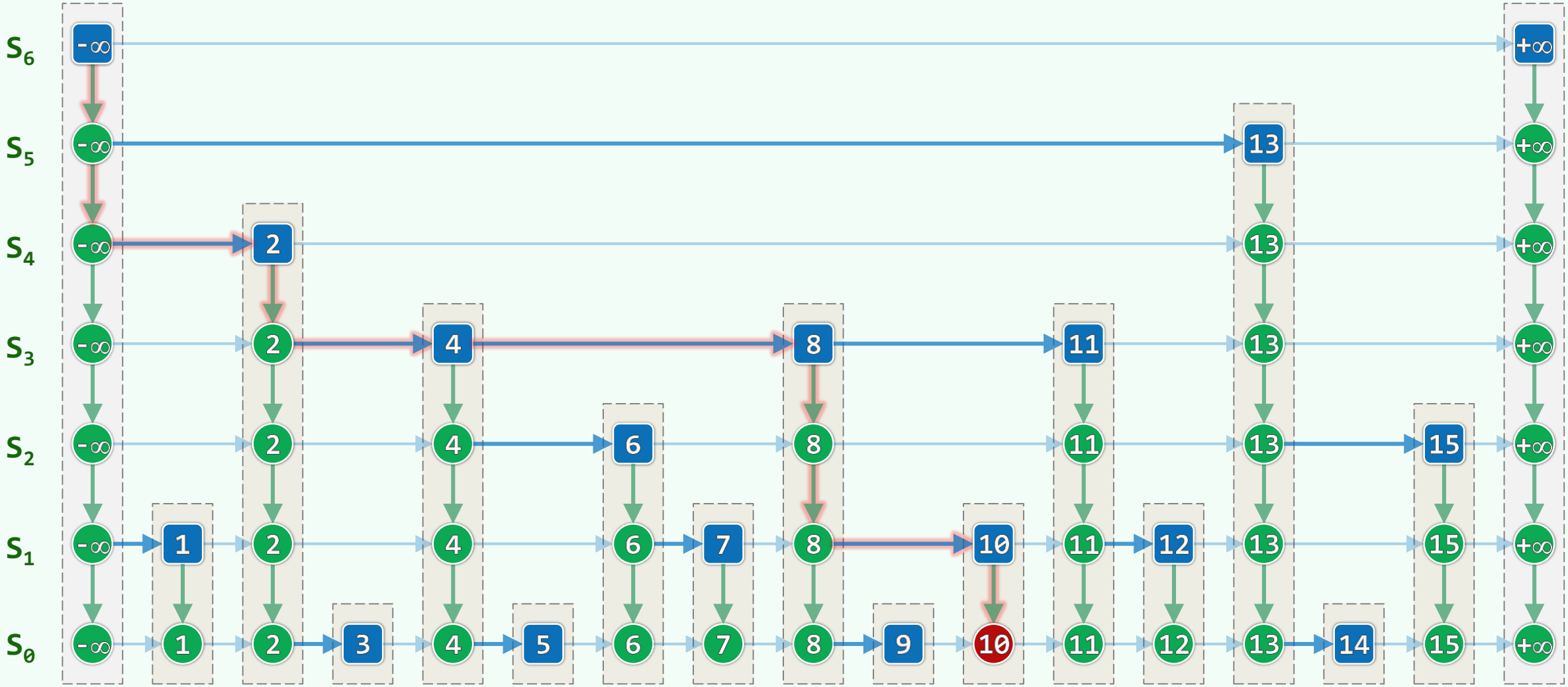
跳转表：查找

她的脚实在娇嫩，从来不下地
而在人们头顶上踱来踱去

只见参仙老怪梁子翁笑嘻嘻的站起身来，向众人拱了拱手，缓步走到庭中，忽地跃起，左足探出，已落在欧阳克插在雪地的筷子之上，拉开架子，...，把一路巧打连绵的“燕青拳”使了出来，脚下纵跳如飞，每一步都落在竖直的筷子之上

邓俊辉
deng@tsinghua.edu.cn

算法：由粗到细 = 由高到低



算法：实现

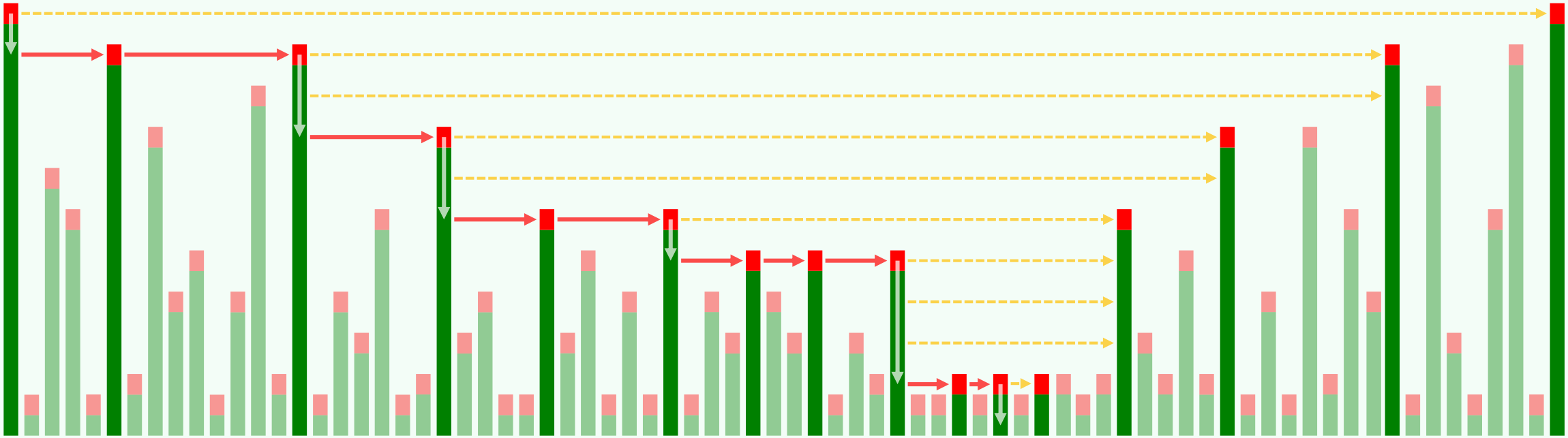
```
template <typename K, typename V> //关键码不大于k的最后一个词条（所对应塔的基座）
QNodePosi< Entry<K, V> > Skiplist<K, V>::search( K k ) {

   for ( QNodePosi< Entry<K, V> > p = first()->data->head; ; ) //从顶层的首节点出发
      if ( (p->succ->succ) && (p->succ->entry.key <= k) )  p = p->succ; //尽可能右移
      else if ( p->below )  p = p->below; //水平越界时，下移
      else  return p; //验证：此时的p符合输出约定（可能是最底层列表的head）

} //体会：得益于哨兵的设置，哪些环节被简化了？
```

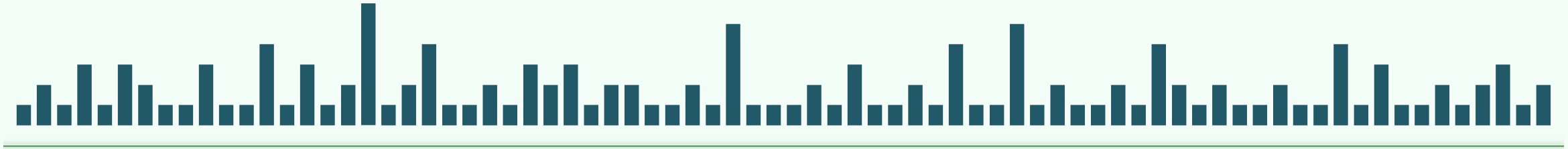
算法：总览

- 查找时间取决于横向、纵向的累计跳转次数
- 会否因层次过多，首先导致纵向跳转过多？



纵向跳转 ~ 层高

❖ 观察：一座塔高度不低于 k 的概率为 p^k；低于 k 的概率为 $1-p^k$



❖ 引理：随着 k 的增加，S_k 为空的概率急剧上升，非空的概率急剧下降

$$Pr(|S_k|=0) \;=\; (1-p^k)^n \;\geq\; 1-n\cdot p^k \qquad\qquad Pr(|S_k|>0) \;\leq\; n\cdot p^k$$

❖ 推论：跳转表高度 $h \;=\; \mathcal{O}(\log n)$ 的概率极大

❖ 比如：若 $p=1/2$，则第 $k=3\cdot\log_{1/p} n$ 层非空（当且仅当 $h\geq k$）的概率为

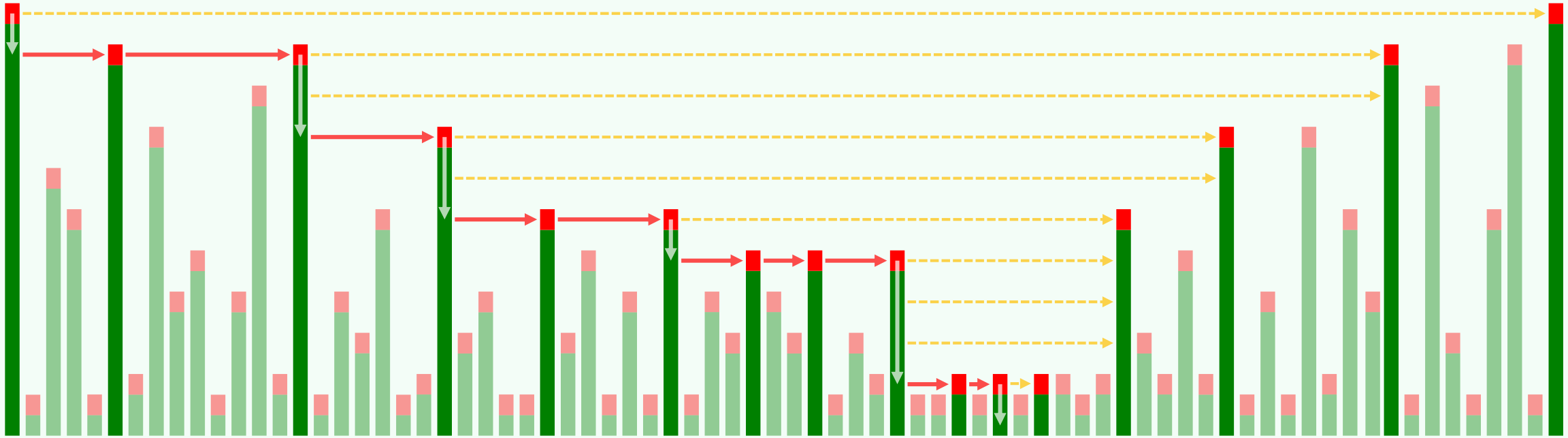
$$Pr(|S_k|>0) \;\leq\; n\cdot p^k \;=\; n\cdot n^{-3} \;=\; 1/n^2 \longmapsto 0 \qquad //w.h.p.$$

❖ 结论：查找过程中，纵向跳转的次数，累计不过 expected-$\mathcal{O}(\log n)$

横向跳转 ~ 紧邻塔顶

❖ 那么：横向跳转是否可能很多次？比如 $\omega(\log n)$，甚至 $\Omega(n)$？

❖ 观察：在同一水平列表中，横向跳转所经节点必然依次紧邻，而且每次抵达都是塔顶

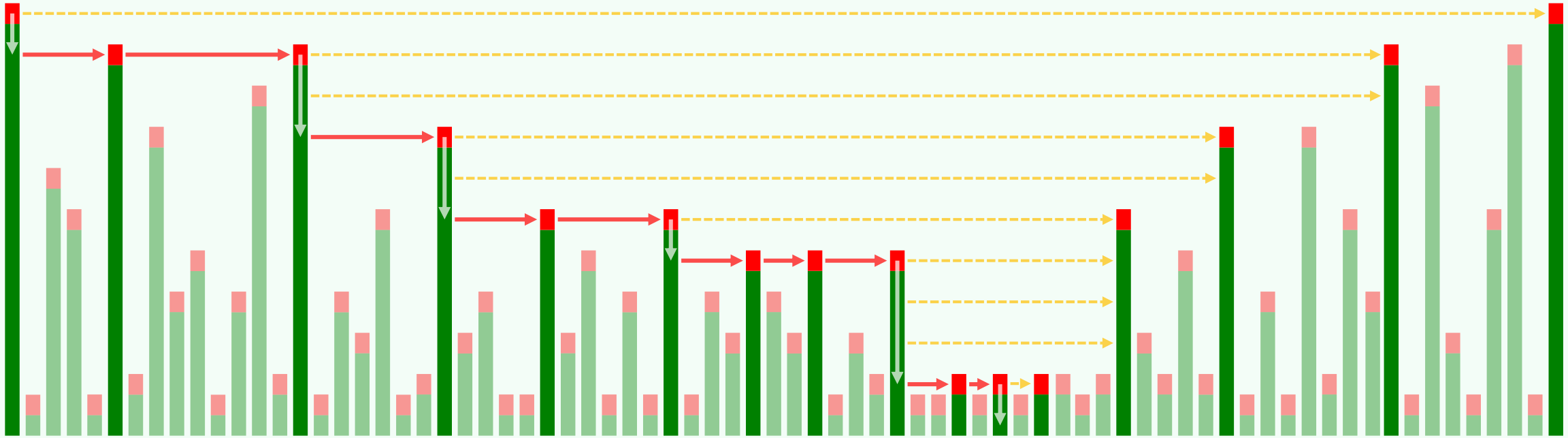


❖ 于是：若将沿同一层列表跳转的次数记作 Y，则它符合几何分布 $Pr(Y = k) = (1-p)^k \cdot p$

横向跳转 ~ 期望时间成本

❖ **定理：** $\mathbb{E}(Y) = (1-p)/p = (1-0.5)/0.5 = 1$ **次**

❖ **因此：在同一层列表中连续跳转的期望时间成本 = 1次跳转 + 1次驻足 = 2**



❖ **结论：跳转表的每次查找，都可在** $\leq$ expected-$(2h)$ $=$ expected-$\mathcal{O}(\log n)$ **时间内完成**