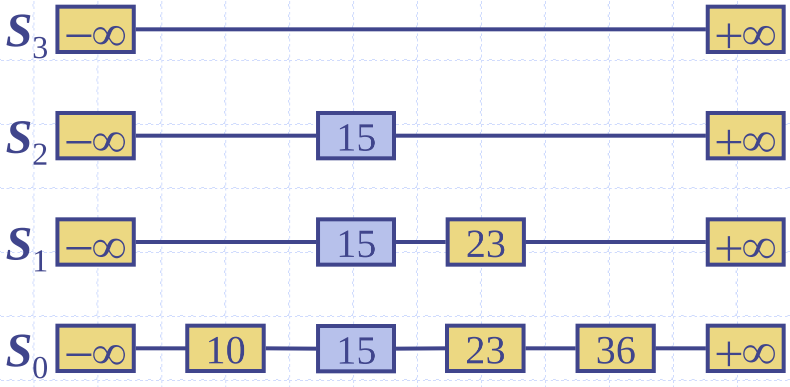
Skip Lists



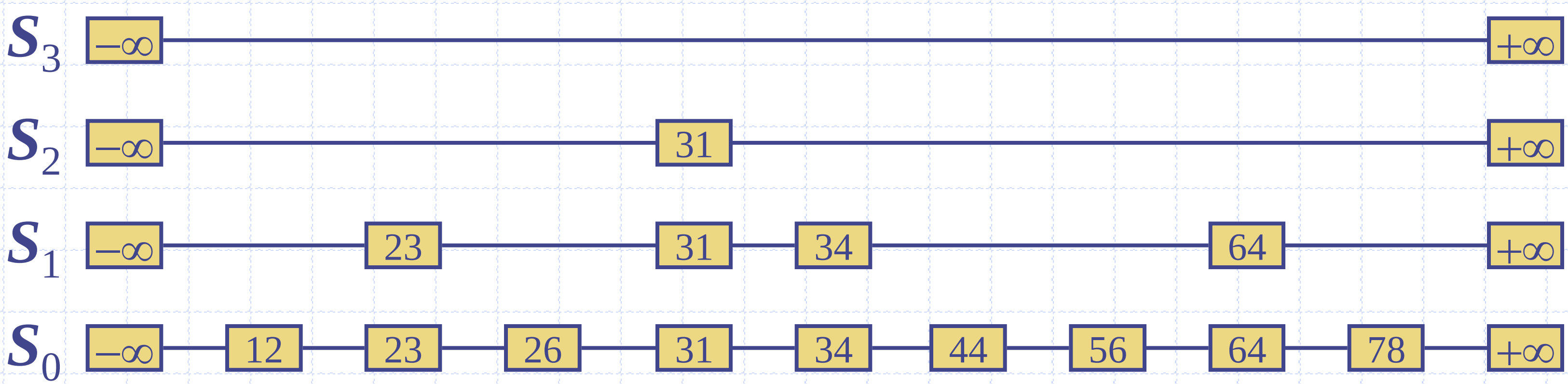
Outline and Reading

- O que é um skip list
- Operations
  - Busca
  - Inserção
  - Remoção
  - Estrutura da dados concreta
- Analise
  - Espaço usado
  - Busca e tempo de atualização

O que é um Skip List

- Um skip list para um conjunto de itens distintos (key, element) é uma série de listas $S_0, S_1, \ldots, S_h$ de modo que:
  - Cada Lista S_i contém uma chave especial $+\infty$ and $-\infty$
  - A Lista S_0 contém chaves de S em ordem não-decrescente
  - Cada Lista é uma subseqüência da anteiror, i.e.,

    $$S_0 \supseteq S_1 \supseteq \ldots \supseteq S_h$$
  - A Lista S_h contém somente duas chaves especiais
- Nós Mostraremos como usar uma skip list para implementar o TAD dicionário



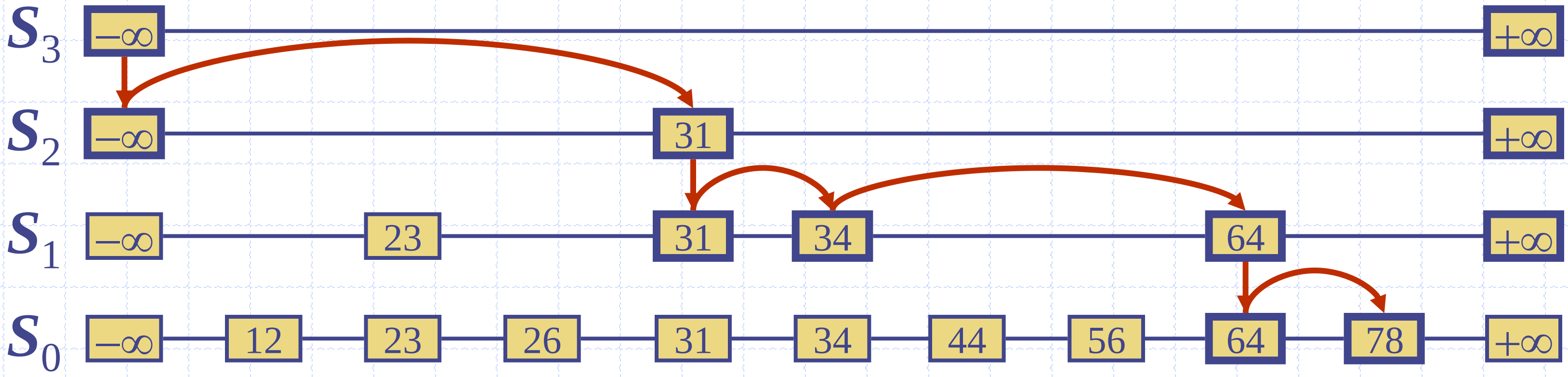
Busca

- Procuramos por uma chave x na skip list abaixo:
  - Iniciamos na primeira posição no topo da lista
  - Na posição p, comparamos x with $y \leftarrow key(after(p))$

    $x = y$: retornamos $element(after(p))$

    $x > y$: fazemos "scan forward"

    $x < y$: fazemos "drop down"
  - Se o final inferior da lista for atingido, retornamos *NO_SUCH_KEY*
- Exemplo: buscar 78



Algoritmos Randômicos

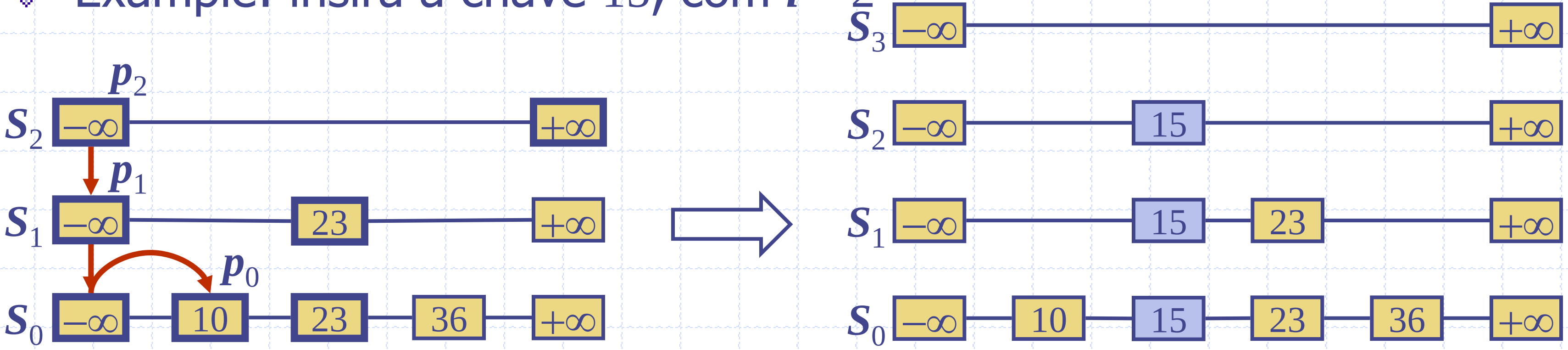
- Um algoritmo randômico faz uso de um fator aleatório para controlar sua execução
- Ele contém um código do tipo

  $b \leftarrow random()$
  if $b = 0$
  do A …
  else { $b = 1$}
  do B …

- O tempo de execução depende dos resultados do fator aleatório (*random()*)
- A análise do tempo de execução de um algoritmo randômico basea-se em um fator aleatório honesto
- No pior caso o tempo de execução do algoritmo randômico é frequentemente alto, mas tem uma probabilidade muito baixa de ocorrer (isso ocorre quando *b,* em todos os sorteios, é zero)
- O algoritmo randômico é utilizado na inserção de itens em um skip list

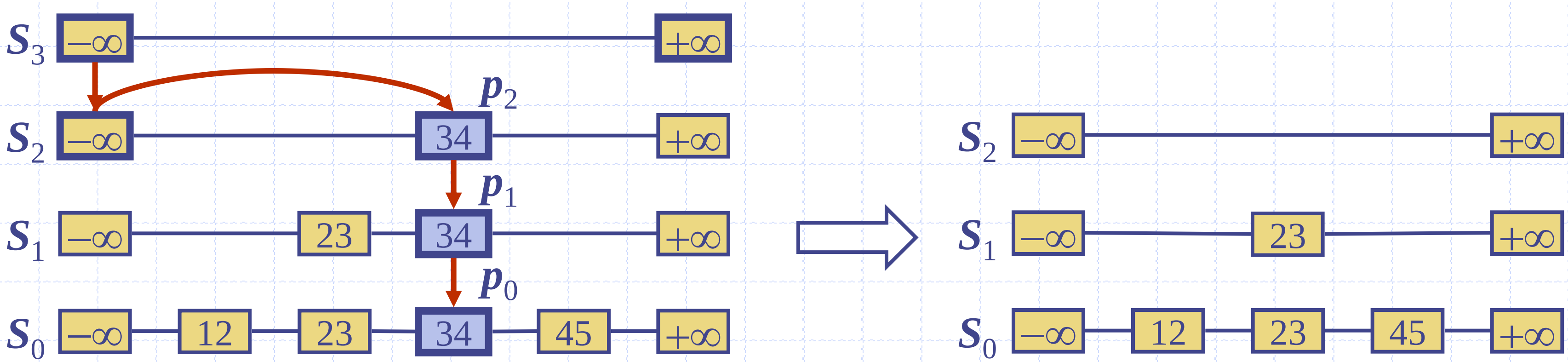
Inserção

- Para inserir um item (x, o) em um skip list, usa-se um algoritmo randômico:
- Repetidamente faz-se o sorteio (0 ou 1) enquanto for zero soma i mais 1 (i denota o número de vezes que o zero foi sorteado)
  - Se $i \geq h$, adicionar ao the skip uma nova lista $S_{h+1}, \ldots, S_{i+1}$, contendo somente as duas chaves especiais
  - Procurar por x no skip list e encontrar a posição $p_0, p_1, \ldots, p_i$ dos itens com a maior chave menor do que x em cada list $S_0, S_1, \ldots, S_i$
  - Para $j \leftarrow 0, \ldots, i$, insira o item (x, o) na lista S_j após a posição p_j
- Example: insira a chave 15, com $i = 2$



Remoção

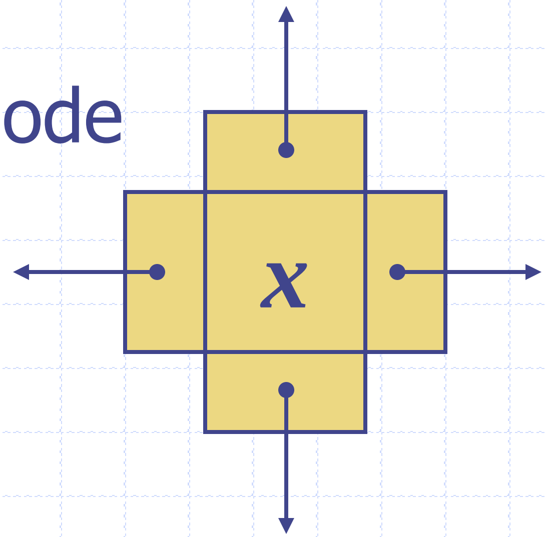
- Para remover um item com a chave x de uma skip list, faz o seguinte:
  - Procurar por x na skip list e encontrar a posição $p_0, p_1, \ldots, p_i$ dos itens com chave x, onde a posição p_j está na lista S_j
  - Removas posições $p_0, p_1, \ldots, p_i$ das listas $S_0, S_1, \ldots, S_i$
  - Remova todas as listas com somente duas chaves especiais (excetuando-se uma)
- Example: remova a chave 34



Implementação

- Pode-se implementar a skip list com um quad-nodes
- Um quad-node armazena:
  - item
  - link para o nó anterior
  - link para o nó posterior
  - link para o nó abaixo
  - link para o nó acima
- Também, define-se chaves especiais para MAIS_INF e MENOS_INF, e deve-se realizar as modificações necessárias no comparator

quad-node



Espaço Usado

- O espaço usado por um skip list depende do sorteio realizado por cada inserção do algoritmo
- Usam-se os seguintes fatos probabilísticos:

  Fato 1: A probabilidade de se ter i consecutivos zeros quando se faz o sorteio é $1/2^i$

  Fato 2: Se cada um dos n itens é apresentado em um conjunto de probabilidade p, o tamanho esperado do conjunto é np

- Considere uma skip list com n itens
  - Pelo fato 1, é inserido um item na lista S_i com probabilidade $1/2^i$
  - Pelo Fato 2, o tamanho esperado da lista S_i é $n/2^i$
- O número esperado de nós usado por um skip list é

$$\sum_{i=0}^{h} \frac{n}{2^i} = n \sum_{i=0}^{h} \frac{1}{2^i} < 2n$$

- Desta forma, o espaço usado de um skip list com n itens é $O(n)$

Altura

- O tempo de execução da busca no algoritmo de inserção é afetado pela altura h do skip list
- Com alta probabilidade, um skip list com n itens tem altura $O(\log n)$
- Usa-se o seguinte fato probabilístico adicional:

  Fato 3: Se cada um dos n eventos tem probabilidade p, a probabilidade de que no mínimo um evento ocorra é no máximo np

- Considere um skip list com n itens
  - Pelo Fato 1, insere-se um item na lista S_i com probabilidade $1/2^i$
  - Pelo Fato 3, a probabilidade de que a lista S_i tenha ao menos um item é no máximo $n/2^i$
- Com $i = 3\log n$, A probabilidade de que $S_{3\log n}$ tanha ao menos um item é no máximo

$$n/2^{3\log n} = n/n^3 = 1/n^2$$

- Um skip list com n itens tem altura no máximo $3\log n$ com probabilidade no mínimo $1 - 1/n^2$

Tempos de Busca e Atualização

- O tempo de busca em um skip list é proporcional ao
  - Número de passos drop-down, mais
  - O número de passos scan-forward
- Os passos drop-down são aproximados pela altura do skip list e é $O(\log n)$ com alta probabilidade
- Para analisar o passo scan-forward, deve-se usar outro fato probabilístico:

  Fato 4: O número de vezes esperado para se conseguir 1 é 2

- Assim, o número de passos scan-forward é $O(\log n)$
- Logo, uma busca em um skip list leva $O(\log n)$ (tempo esperado)
- A análise da inserção e remoção tem resultados similares

Resumo

- Um skip list é uma estrutura de dados para dicionários que usa um Algoritmo de inserção randomizado
- Em uma skip list com n itens
  - O espaço usado é $O(n)$
  - O tempo esperado para busca, inserção e remoção é $O(\log n)$
- Usando análise complexa de probabilidade pode-se mostrar que essas performances aproximam-se com alta probabilidade do que foi mostrado
- Skip lists são rápidas e simples de implementar