

Hashing

Estrutura de Dados II
Jairo Francisco de Souza

Resolução de colisões

- ✓ Endereçamento aberto
 - ✓ Re-hashing
 - ✓ Encadeamento
 - ✓ Separado
 - ✓ Coalescido
 - ✓ Coalescido com porção
- ✓ Endereçamento em balde

Endereçamento Aberto

- Usar próxima posição vazia
- Inserimos chave u, x, v, w, y e z na tabela:
 - Inserir u e v $\rightarrow h(u)=423$ e $h(x)=425$.

422	423	424	425	426	427	428
	u		x			

- Inserir v $\rightarrow h(v)=423$
 - (colisão com u $\rightarrow h(u)=423$)
 - Próxima posição vazia é 424

422	423	424	425	426	427	428
	u	v	x			

- Inserir w $\rightarrow h(w)=423$ (onde inserir w?)

422	423	424	425	426	427	428
	u	v	x	w		

- Inserir y e z $\rightarrow h(y)=h(z)=425$

422	423	424	425	426	427	428
	u	v	x	w	y	z

Endereçamento Aberto

- Resultado da inserção de u, x, v, w, y e z

422	423	424	425	426	427	428	
	u	v	x	w	y	z	

- Suponha que deseja-se procurar n, onde $h(n)=424$
 - É necessário ir até o 429 para concluir que n não está presente
- Remoção com endereçamento aberto também leva a inconsistência. Se x for removido, acessos a w, y e z estariam perdidos
- Em alguns casos, não há remoção quando se usa “hashing”

Re-hashing

- Se houver colisão, usar outra função para encontrar próxima posição
 - Exemplo: utilizar $r(x)$ para resolver colisão
- Posições para uma chave x são encontradas da seguinte forma:
 - $p_1 = h(x)$
 - $p_2 = h(x) + r(x)$,
 - $p_3 = h(x) + 2 * r(x)$
 - Etc
- De forma mais geral:
 - $h(k, i) = (h_1(k) + i h_2(k)) \bmod m$

Re-hashing

- Considere a tabela abaixo com 13 espaços
- Considere incluir o valor 14 na tabela, com
 - $h_1(k) = k \bmod 13$
 - $h_2(k) = 1 + (k \bmod 11)$
- Em qual posição $k=14$ será inserida?

0	
1	79
2	
3	
4	69
5	98
6	
7	72
8	
9	
10	
11	50
12	

Re-hashing

- Considere a tabela abaixo com 13 espaços
- Considere incluir o valor 14 na tabela, com
 - $h_1(k) = k \bmod 13$
 - $h_2(k) = 1 + (k \bmod 11)$
- Em qual posição $k=14$ será inserida?
 - posição 9!

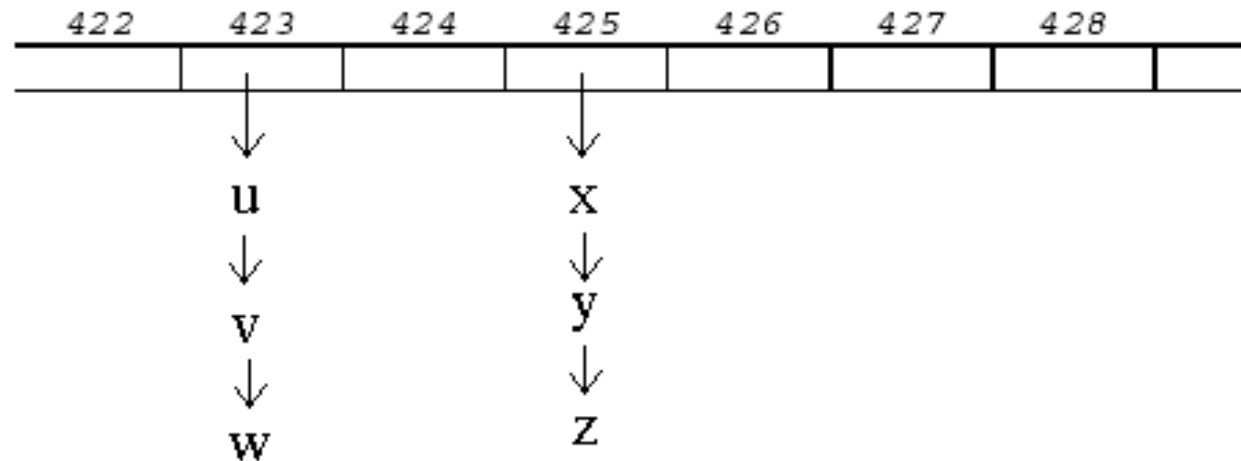
0	
1	79
2	
3	
4	69
5	98
6	
7	72
8	
9	
10	
11	50
12	

Re-hashing

- Vantagem de usar outra função r diferente de h :
 - se $h(x)=h(y)=h(z)=i$, é pouco provável que $r(y)=r(z)$, logo
 - x fica em i ;
 - y fica em $i+r(y)$; e
 - z em $i+2*r(z)$
 - busca por z é mais rápida, pois evita calcular $r(y)$
- Identificação de que não está presente, quando encontra-se posição vazia
- Tem problemas também para remoção

Encadeamento (separado)

- Cada posição da tabela está associada com uma lista ligada
- Exemplo:



- Este método é chamado de encadeamento separado
- Neste método a tabela nunca irá transbordar
- Aumento do comprimento da lista pode degradar o desempenho da recuperação
- Pode-se ordenar os registros presentes na lista ligada
 - Não será executada busca exaustiva
- Necessidade de espaço adicional para os ponteiros

Encadeamento (coalescido)

- Cada posição na tabela possui espaço para a chave (ou info) e para um ponteiro que aponta para a posição da próxima chave
- Método quando ocorre colisão
 - Encontrar primeira posição disponível
 - Armazenar índice dessa posição com a chave que já está na tabela da lista ligada
 - Exemplo:
 - Inserir A_2 , A_3 , A_5

0		
1		
2	A_2	
3	A_3	
4		
5	A_5	
6		
7		
8		
9		

Encadeamento (coalescido)

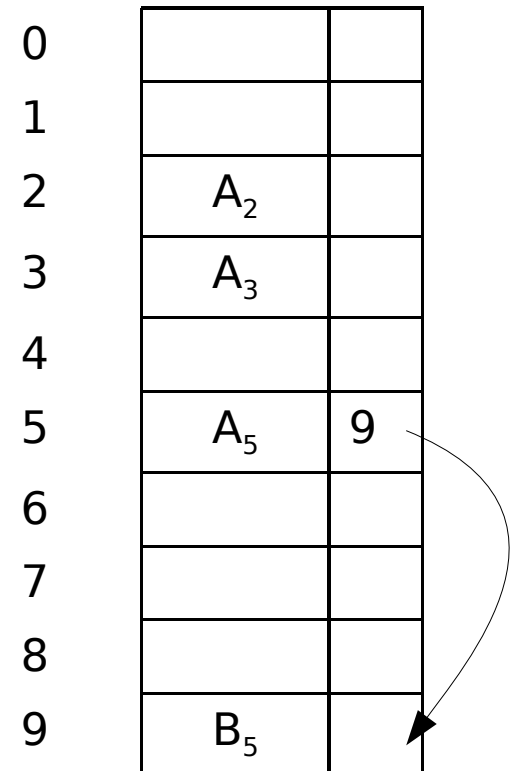
- Cada posição na tabela possui espaço para a chave (ou info) e para um ponteiro que aponta para a posição da próxima chave
- Método quando ocorre colisão
 - Encontrar primeira posição disponível
 - Armazenar índice dessa posição com a chave que já está na tabela da lista ligada
 - Exemplo:
 - Inserir A_2 , A_3 , A_5
 - Inserir B_5
 - Colisão com A_5

0		
1		
2	A_2	
3	A_3	
4		
5	A_5	
6		
7		
8		
9		

Encadeamento (coalescido)

- Cada posição na tabela possui espaço para a chave (ou info) e para um ponteiro que aponta para a posição da próxima chave
- Método quando ocorre colisão
 - Encontrar primeira posição disponível
 - Armazenar índice dessa posição com a chave que já está na tabela da lista ligada
- Exemplo:
 - Inserir A_2 , A_3 , A_5
 - Inserir B_5
 - Colisão com A_5
 - Inserir B_5 na posição vazia a partir do final da tabela

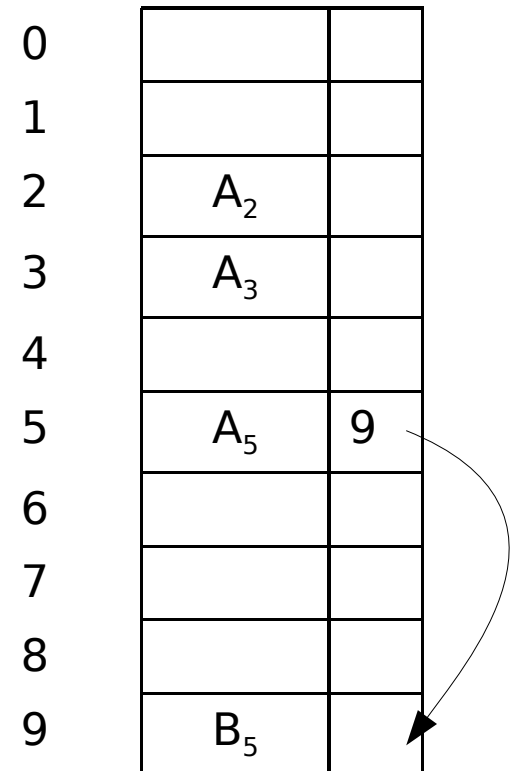
0		
1		
2	A_2	
3	A_3	
4		
5	A_5	9
6		
7		
8		
9	B_5	



Encadeamento (coalescido)

- Cada posição na tabela possui espaço para a chave (ou info) e para um ponteiro que aponta para a posição da próxima chave
- Método quando ocorre colisão
 - Encontrar primeira posição disponível
 - Armazenar índice dessa posição com a chave que já está na tabela da lista ligada
- Exemplo:
 - Inserir A_2 , A_3 , A_5
 - Inserir B_5
 - Colisão com A_5
 - Inserir B_5 na posição vazia a partir do final da tabela
 - Inserir A_9 (Colisão com B_5)

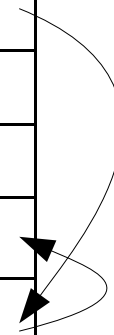
0		
1		
2	A_2	
3	A_3	
4		
5	A_5	9
6		
7		
8		
9	B_5	



Encadeamento (coalescido)

- Cada posição na tabela possui espaço para a chave (ou info) e para um ponteiro que aponta para a posição da próxima chave
- Método quando ocorre colisão
 - Encontrar primeira posição disponível
 - Armazenar índice dessa posição com a chave que já está na tabela da lista ligada
- Exemplo:
 - Inserir A_2 , A_3 , A_5
 - Inserir B_5
 - Colisão com A_5
 - Inserir B_5 na posição vazia a partir do final da tabela
 - Inserir A_9 (Colisão com B_5)
 - Inserir na próxima posição vazia e apontar

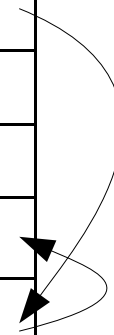
0		
1		
2	A_2	
3	A_3	
4		
5	A_5	9
6		
7		
8	A_9	
9	B_5	8



Encadeamento (coalescido)

- Cada posição na tabela possui espaço para a chave (ou info) e para um ponteiro que aponta para a posição da próxima chave
- Método quando ocorre colisão
 - Encontrar primeira posição disponível
 - Armazenar índice dessa posição com a chave que já está na tabela da lista ligada
 - Exemplo:
 - Inserir A_2 , A_3 , A_5
 - Inserir B_5
 - Colisão com A_5
 - Inserir B_5 na posição vazia a partir do final da tabela
 - Inserir A_9 (Colisão com B_5)
 - Inserir na próxima posição vazia e apontar
 - Inserir B_2 (Colisão com A_2)

0		
1		
2	A_2	
3	A_3	
4		
5	A_5	9
6		
7		
8	A_9	
9	B_5	8



Encadeamento (coalescido)

- Cada posição na tabela possui espaço para a chave (ou info) e para um ponteiro que aponta para a posição da próxima chave
- Método quando ocorre colisão
 - Encontrar primeira posição disponível
 - Armazenar índice dessa posição com a chave que já está na tabela da lista ligada
 - Exemplo:
 - Inserir A_2 , A_3 , A_5
 - Inserir B_5
 - Colisão com A_5
 - Inserir B_5 na posição vazia a partir do final da tabela
 - Inserir A_9 (Colisão com B_5)
 - Inserir na próxima posição vazia e apontar
 - Inserir B_2 (Colisão com A_2)
 - Inserir na próxima posição vazia e apontar

0		
1		
2	A_2	6
3	A_3	
4		
5	A_5	9
6	B_2	
7		
8	A_9	
9	B_5	8

```
graph TD; 6 --> 2; 9 --> 5; 8 --> 9; 6 --> 8;
```


Encadeamento (coalescido)

- Posições disponíveis podem ser marcadas, por exemplo, -2 em next
- -1 pode ser usado para indicar o final de uma cadeia

0		-2
1		-2
2	A_2	6
3	A_3	-1
4		-2
5	A_5	9
6	B_2	-1
7		-2
8	A_9	-1
9	B_5	8

Encadeamento (coalescido)

- Posições disponíveis podem ser marcadas, por exemplo, -2 em next
- -1 pode ser usado para indicar o final de uma cadeia
- Exemplo:
 - Inserir A_7
 - Inserir C_2
 - Colisão com A_2
 - Segue-se para posição 6
 - Posição 6 está ocupada
 - Next indica fim da cadeia (-1)

0		-2
1		-2
2	A_2	6
3	A_3	-1
4		-2
5	A_5	9
6	B_2	-1
7	A_7	-1
8	A_9	-1
9	B_5	8

Encadeamento (coalescido)

- Posições disponíveis podem ser marcadas, por exemplo, -2 em next
- -1 pode ser usado para indicar o final de uma cadeia
- Exemplo:
 - Inserir A_7
 - Inserir C_2
 - Colisão com A_2
 - Segue-se para posição 6
 - Posição 6 está ocupada
 - Next indica fim da cadeia (-1)
 - Achar nova posição vazia
 - a partir do final
 - posição 4

0		-2
1		-2
2	A_2	6
3	A_3	-1
4		-2
5	A_5	9
6	B_2	-1
7	A_7	-1
8	A_9	-1
9	B_5	8

Encadeamento (coalescido)

- Posições disponíveis podem ser marcadas, por exemplo, -2 em next
- -1 pode ser usado para indicar o final de uma cadeia

- Exemplo:

- Inserir A_7
- Inserir C_2
- Colisão com A_2
- Segue-se para posição 6
- Posição 6 está ocupada
- Next indica fim da cadeia (-1)
- Achar nova posição vazia
 - a partir do final
 - posição 4
- Inserir C_2
- Montar cadeia (posição 6 aponta para

0		-2
1		-2
2	A_2	6
3	A_3	-1
4	C_2	-1
5	A_5	9
6	B_2	4
7	A_7	-1
8	A_9	-1
9	B_5	8

Encadeamento (coalescido com porão)

- Uso de área de transbordamento para armazenar chaves quando não há mais espaço na tabela.
 - Esta área pode ser alocada dinamicamente
- Exemplo:
 - Inserir A_2 , A_3 e A_5

0		
1		
2	A_2	
3	A_3	
4		
5	A_5	
6		
7		
8		
9		
10		
11		
12		

Porão {

Encadeamento (coalescido com porão)

- Uso de área de transbordamento para armazenar chaves quando não há mais espaço na tabela.
 - Esta área pode ser alocada dinamicamente
- Exemplo:
 - Inserir A_2 , A_3 e A_5
 - Inserir B_5 (colisão com A_5)
 - Achar posição no porão

0		
1		
2	A_2	
3	A_3	
4		
5	A_5	
6		
7		
8		
9		
10		
11		
12		

Porão {

Encadeamento (coalescido com porão)

- Uso de área de transbordamento para armazenar chaves quando não há mais espaço na tabela.
 - Esta área pode ser alocada dinamicamente
- Exemplo:
 - Inserir A_2 , A_3 e A_5
 - Inserir B_5 (colisão com A_5)
 - Achar posição no porão
 - Incluir link entre as posições

0		
1		
2	A_2	
3	A_3	
4		
5	A_5	12
6		
7		
8		
9		
10		
11		
12	B_5	

Porão

Encadeamento (coalescido com porão)

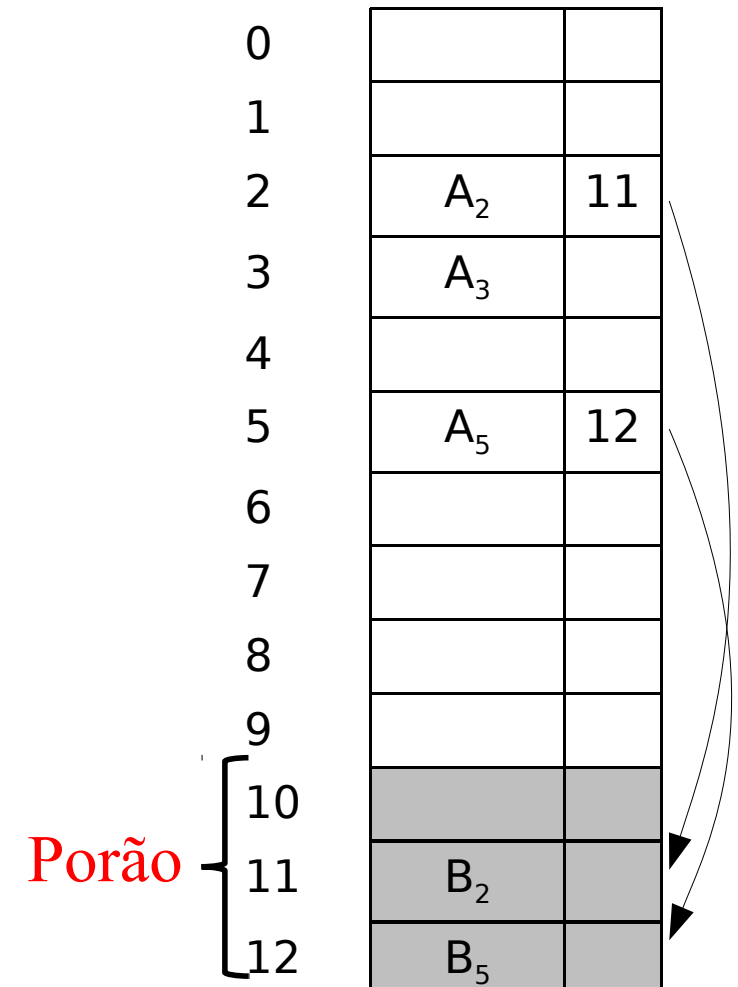
- Uso de área de transbordamento para armazenar chaves quando não há mais espaço na tabela.
 - Esta área pode ser alocada dinamicamente
- Exemplo:
 - Inserir A_2 , A_3 e A_5
 - Inserir B_5 (colisão com A_5)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir B_2 (colisão com A_2)

0		
1		
2	A_2	
3	A_3	
4		
5	A_5	12
6		
7		
8		
9		
10		
11		
12	B_5	

Porão

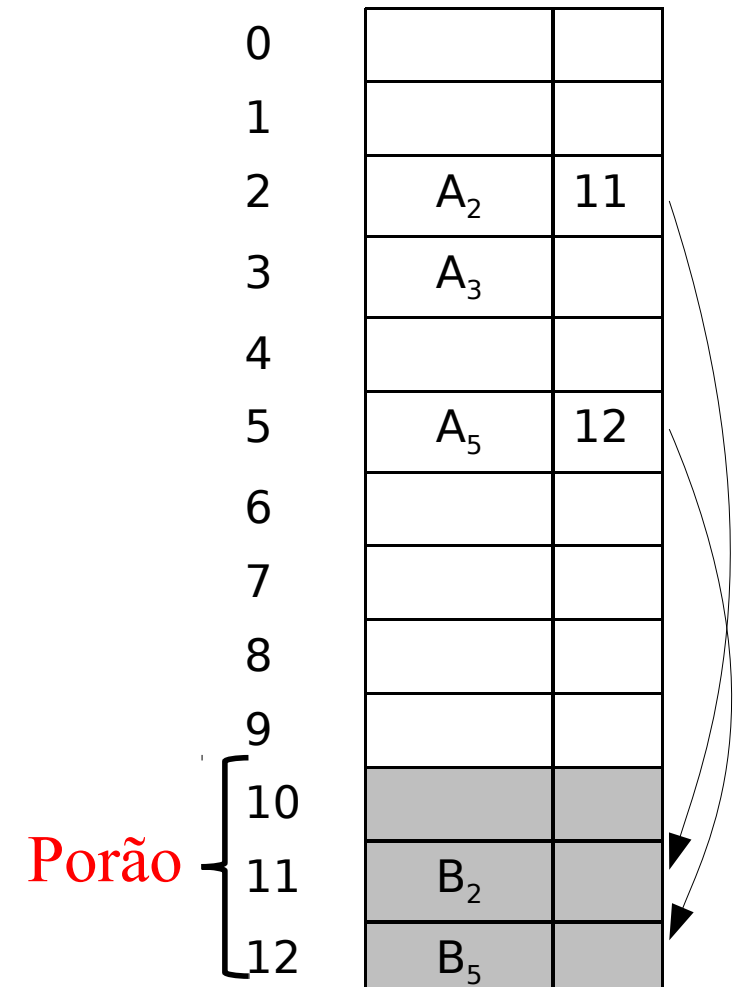
Encadeamento (coalescido com porão)

- Uso de área de transbordamento para armazenar chaves quando não há mais espaço na tabela.
 - Esta área pode ser alocada dinamicamente
- Exemplo:
 - Inserir A_2 , A_3 e A_5
 - Inserir B_5 (colisão com A_5)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir B_2 (colisão com A_2)
 - Achar posição no porão
 - Incluir link entre as posições



Encadeamento (coalescido com porão)

- Uso de área de transbordamento para armazenar chaves quando não há mais espaço na tabela.
 - Esta área pode ser alocada dinamicamente
- Exemplo:
 - Inserir A_2 , A_3 e A_5
 - Inserir B_5 (colisão com A_5)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir B_2 (colisão com A_2)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir A_9 e B_9



Encadeamento (coalescido com porão)

- Uso de área de transbordamento para armazenar chaves quando não há mais espaço na tabela.
 - Esta área pode ser alocada dinamicamente
- Exemplo:
 - Inserir A_2 , A_3 e A_5
 - Inserir B_5 (colisão com A_5)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir B_2 (colisão com A_2)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir A_9 e B_9

0		
1		
2	A_2	11
3	A_3	
4		
5	A_5	12
6		
7		
8		
9	A_9	10
Porão	B_9	
	B_2	
	B_5	

Encadeamento (coalescido com porão)

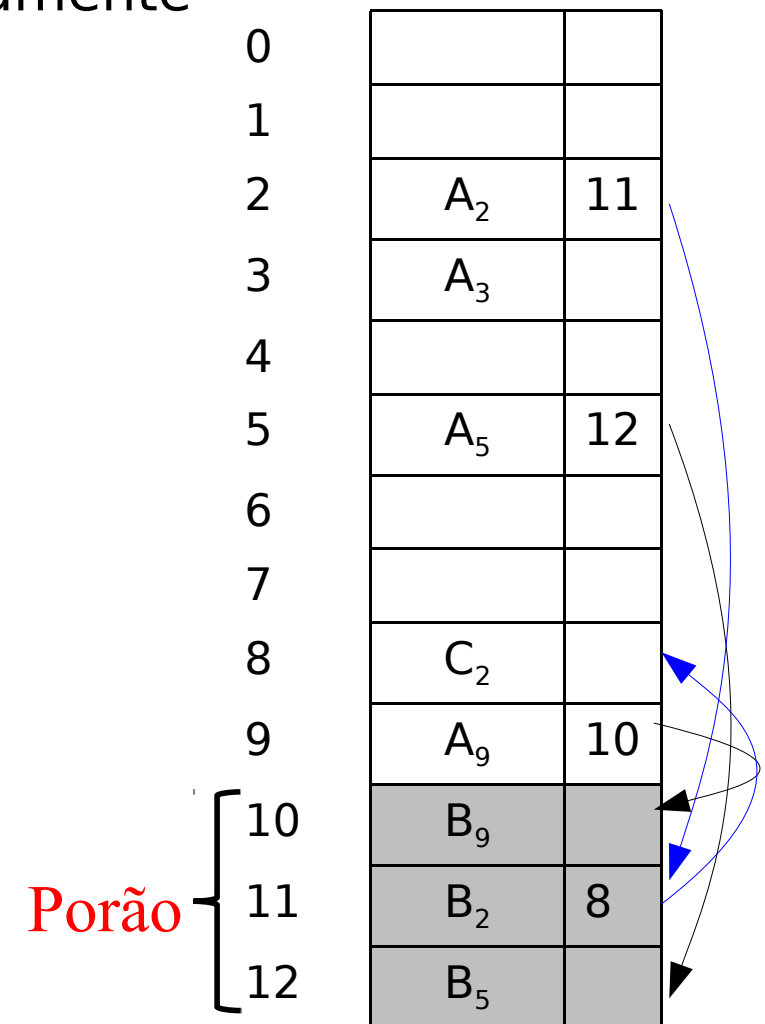
- Uso de área de transbordamento para armazenar chaves quando não há mais espaço na tabela.
 - Esta área pode ser alocada dinamicamente
- Exemplo:
 - Inserir A_2 , A_3 e A_5
 - Inserir B_5 (colisão com A_5)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir B_2 (colisão com A_2)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir A_9 e B_9
 - Inserir C_2

0		
1		
2	A_2	11
3	A_3	
4		
5	A_5	12
6		
7		
8		
9	A_9	10
10	B_9	
11	B_2	
12	B_5	

Porão

Encadeamento (coalescido com porão)

- Uso de área de transbordamento para armazenar chaves quando não há mais espaço na tabela.
 - Esta área pode ser alocada dinamicamente
- Exemplo:
 - Inserir A_2 , A_3 e A_5
 - Inserir B_5 (colisão com A_5)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir B_2 (colisão com A_2)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir A_9 e B_9
 - Inserir C_2



Encadeamento (coalescido com porão)

- Uso de área de transbordamento para armazenar chaves quando não há mais espaço na tabela.
 - Esta área pode ser alocada dinamicamente
- Exemplo:
 - Inserir A_2 , A_3 e A_5
 - Inserir B_5 (colisão com A_5)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir B_2 (colisão com A_2)
 - Achar posição no porão
 - Incluir link entre as posições
 - Inserir A_9 e B_9
 - Inserir C_2

Usando:

-2 para posição vazia
-1 para fim da cadeia

0		-2
1		-2
2	A_2	11
3	A_3	-1
4		-2
5	A_5	12
6		-2
7		-2
8	C_2	-1
9	A_9	10
10	B_9	-1
11	B_2	8
12	B_5	-1

Porão

Endereçamento em balde

- Armazenar na mesma posição os elementos que colidem
- Associação de um balde a cada endereço
- Balde
 - Bloco de espaço grande o suficiente para armazenar múltiplos itens
- Não evita colisão
- Se o balde está cheio
 - Próximo elemento pode ser armazenado no próximo balde vazio
 - Área de transbordamento também pode ser utilizada
 - Neste caso, cada balde deve ser marcado para indicar se a busca deve ser continuada nessa área
 - Marcador sim/não
 - Em conjunto com o marcador pode-se utilizar um ponteiro para indicar onde a lista inicia na área de transbordamento

Endereçamento em balde

- Exemplo: Balde e sondagem linear pela próxima posição vazia
 - Inserir $A_5, A_2, A_3, B_5, A_9, B_2, B_9, C_2$

Endereçamento em balde

- Exemplo: Balde e sondagem linear pela próxima posição vazia
 - Inserir A_5 , A_2 , A_3 , B_5 , A_9 , B_2 , B_9 , C_2

0		
1		
2	A_2	
3	A_3	
4		
5	A_5	
6		
7		
8		
9		

Endereçamento em balde

- Exemplo: Balde e sondagem linear pela próxima posição vazia
 - Inserir A_5 , A_2 , A_3 , B_5 , A_9 , B_2 , B_9 , C_2

0		
1		
2	A_2	
3	A_3	
4		
5	A_5	B_5
6		
7		
8		
9		

Endereçamento em balde

- Exemplo: Balde e sondagem linear pela próxima posição vazia
 - Inserir A_5 , A_2 , A_3 , B_5 , A_9 , B_2 , B_9 , C_2

0		
1		
2	A_2	
3	A_3	
4		
5	A_5	B_5
6		
7		
8		
9	A_9	

Endereçamento em balde

- Exemplo: Balde e sondagem linear pela próxima posição vazia
 - Inserir $A_5, A_2, A_3, B_5, A_9, B_2, B_9, C_2$

0		
1		
2	A_2	B_2
3	A_3	
4		
5	A_5	B_5
6		
7		
8		
9	A_9	

Endereçamento em balde

- Exemplo: Balde e sondagem linear pela próxima posição vazia
 - Inserir $A_5, A_2, A_3, B_5, A_9, B_2, B_9, C_2$

0		
1		
2	A_2	B_2
3	A_3	
4		
5	A_5	B_5
6		
7		
8		
9	A_9	B_9

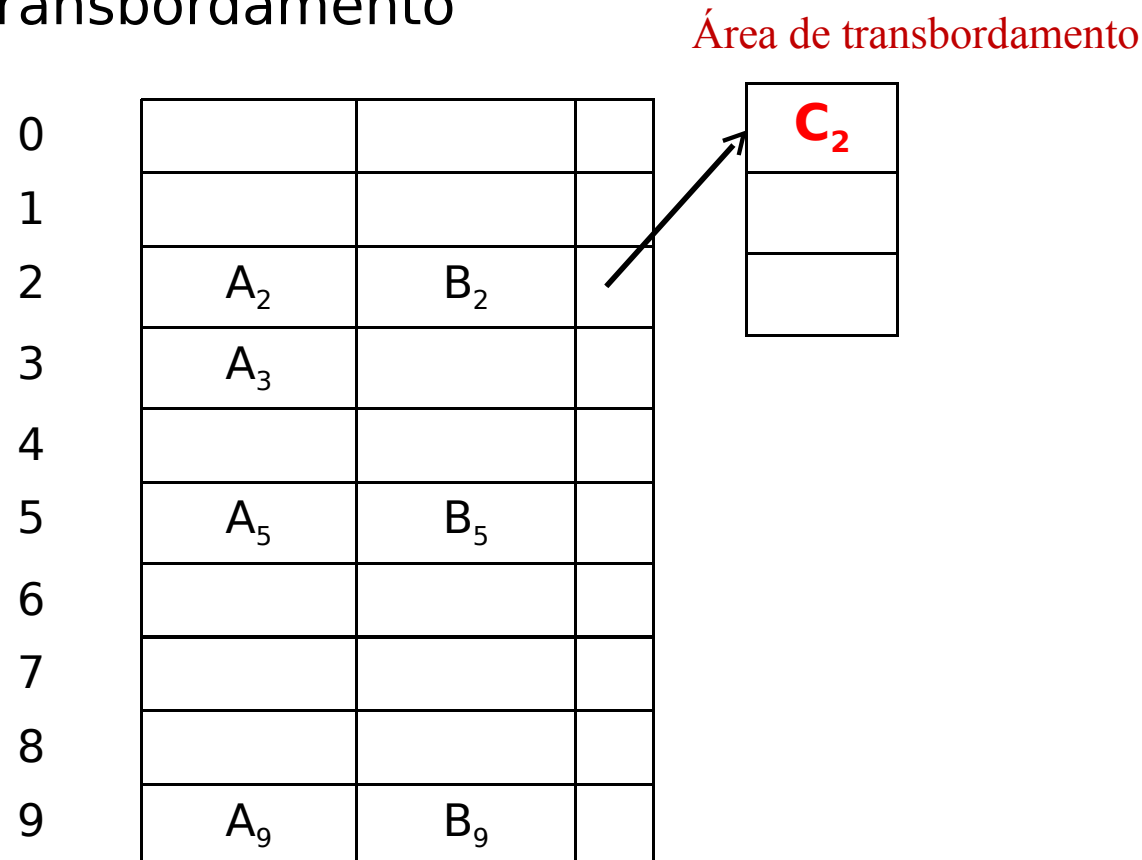
Endereçamento em balde

- Exemplo: Balde e sondagem linear pela próxima posição vazia
 - Inserir $A_5, A_2, A_3, B_5, A_9, B_2, B_9, C_2$

0		
1		
2	A_2	B_2
3	A_3	C_2
4		
5	A_5	B_5
6		
7		
8		
9	A_9	B_9

Endereçamento em balde

- Exemplo: Balde e transbordamento
 - Inserir $A_5, A_2, A_3, B_5, A_9, B_2, B_9, C_2$
 - Ao inserir C_2 o balde está cheio, logo este é inserido na área de transbordamento



Remoção

- Com um método de encadeamento
 - Remover elemento leva à remoção de um nó da lista ligada que contém o elemento
- Para outros métodos
 - Remoção pode exigir tratamento mais cuidadoso da resolução de colisões, exceto no caso raro de uma função hashing perfeita ser utilizada

- Exemplo:

0	
1	A_1
2	A_2
3	B_1
4	A_4
5	B_4
6	
7	
8	
9	

Remover A_4



0	
1	A_1
2	A_2
3	B_1
4	
5	B_4
6	
7	
8	
9	

Pesquisar B_4
leva a resposta
errada

Remoção

- Com um método de encadeamento
 - Remover elemento leva à remoção de um nó da lista ligada que contém o elemento
- Para outros métodos
 - Remoção pode exigir tratamento mais cuidadoso da resolução de colisões, exceto no caso raro de uma função hashing perfeita ser utilizada

- Exemplo:

0	
1	A_1
2	A_2
3	B_1
4	
5	B_4
6	
7	
8	
9	

Remover A_2



0	
1	A_1
2	
3	B_1
4	
5	B_4
6	
7	
8	
9	

Pesquisar B_1
leva a resposta
errada

Remoção

- Possível solução
 - Manter chaves removidas, marcando as posições como elementos não válidos
 - Busca não termina prematuramente
 - Ao inserir nova chave, ela sobrescreve posição com elemento não válido

0		
1	A_1	
2	A_2	x
3	B_1	
4	A_4	x
5	B_4	
6		
7		
8		
9		

Remoção

- Possível solução
 - Manter chaves removidas, marcando as posições como elementos não válidos
 - Busca não termina prematuramente
 - Ao inserir nova chave, ela sobrescreve posição com elemento não válido

0		
1	A_1	
2	A_2	x
3	B_1	
4	A_4	x
5	B_4	
6		
7		
8		
9		

Para muitas remoções e poucas inserções

- A tabela se torna sobrecarregada de registros removidos
- Necessidade de testar os elementos removidos

Registros devem ser expurgados depois de certo tempo e tabela reorganizada

Remoção

- Possível solução
 - Manter chaves removidas, marcando as posições como elementos não válidos
 - Busca não termina prematuramente
 - Ao inserir nova chave, ela sobrescreve posição com elemento não válido

0		
1	A_1	
2	B_1	
3		
4	B_4	
5		
6		
7		
8		
9		

Para muitas remoções e poucas inserções

- A tabela se torna sobrecarregada de registros removidos
- Necessidade de testar os elementos removidos

Registros devem ser expurgados depois de certo tempo e tabela reorganizada