Fila Duplamente Encadeada (FDE) de Prioridade contendo indicação de *Frente*, *Cauda* e um referencial móvel (*refMovel*)

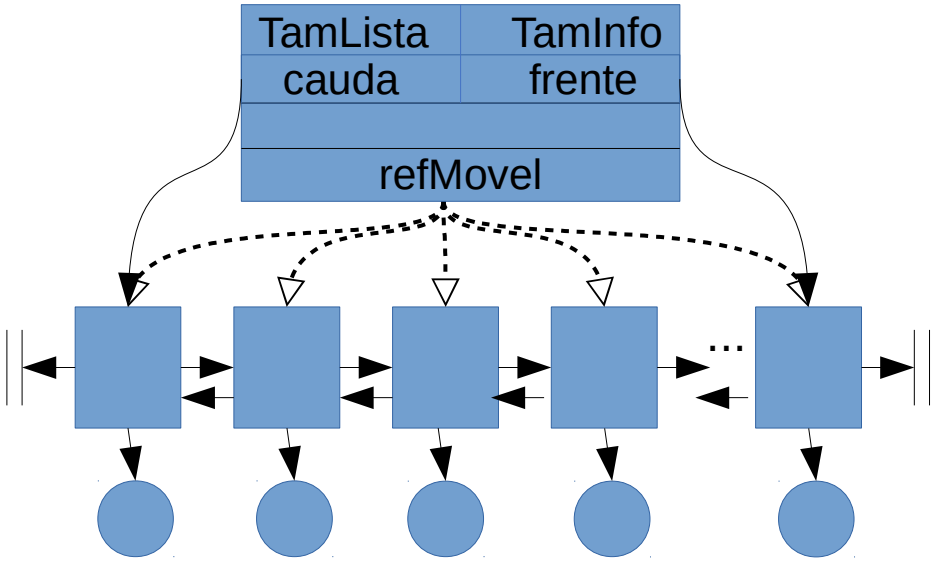
A implementação mais eficiente para uma fila de prioridade consiste no modelo chamado *Heap Priority Queue*;

Heap Priority Queue utiliza uma árvore binária, tema que ainda será abordado;

Futuramente será possível abordar a *Heap Priority Queue;*

A presente proposta tem como objetivo exercitar uma implementação com alguma melhoria de eficiência utilizando os conceitos já abordados na disciplina.

Fila Duplamente Encadeada (FDE) de Prioridade contendo indicação de *Frente*, *Cauda* e um referencial móvel (*refMovel*)



FDE de Prioridade contendo indicação de *Frente*, *Cauda* e um referencial móvel (*refMovel*)

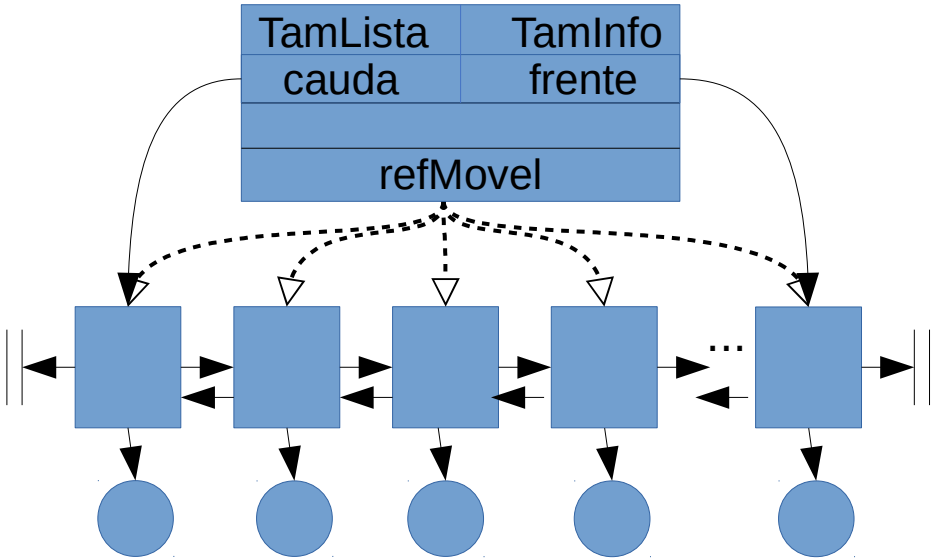
Frente, *Cauda* apontam para as respectivas extremidades da fila;

Considere que o referencial móvel (*refMovel*):

1)É igual a Null para a fila vazia ou…

2)Aponta para o endereço do elemento mais recentemente inserido.

FDE estiver vazia: *cauda* == *frente*== *refMovel*==*NULL*;



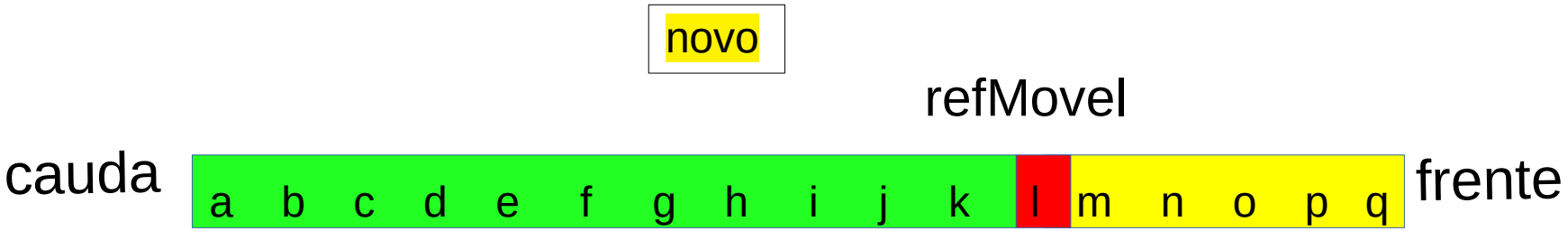
Explorando a vantagem do RefMovel na FDE de prioridade:

A ideia é **tirar proveito** do referencial móvel.

Discutiremos os casos possíveis para a inserção, considerando a idade de um indivíduo como campo chave de prioridade (quanto maior a prioridade mais à frente será a inserção).

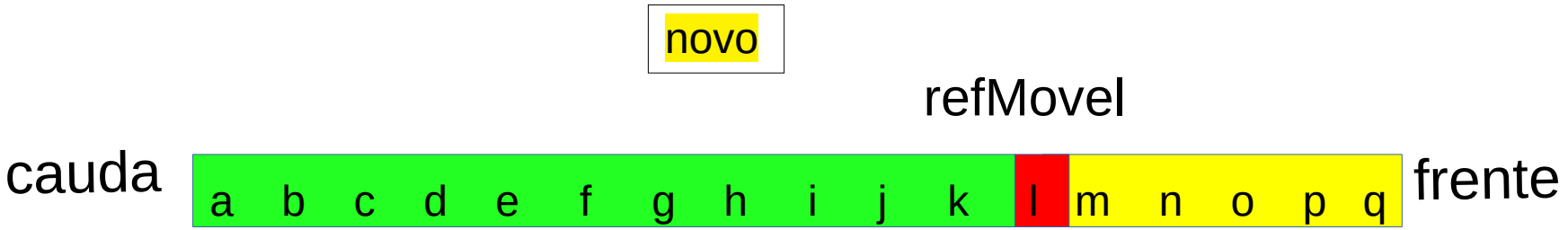
A função idade( ) retorna a idade armazenada no nó da fila.

O **desenho** é uma mera ilustração colorida, as implementações são realizadas por encadeamento, não por um arranjo do tipo vetor.



Explorando a vantagem do RefMovel na FDE de prioridade:

A ideia é **tirar proveito** do referencial móvel.



Caso #1

Se (idade(novo) ≤ idade(cauda))

Então o novo elemento será inserido convencionalmente como nova cauda

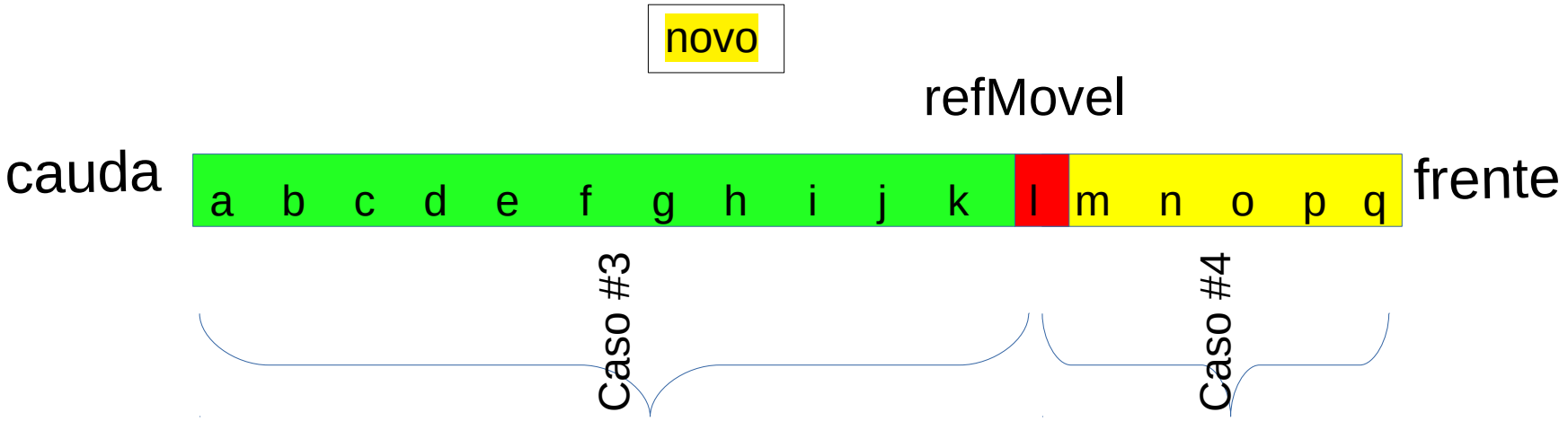
Caso #2

Se (idade(*idade(frente) < idade(novo))*

Então o novo elemento será inserido como novo item de frente

Explorando a vantagem do RefMovel na FDE de prioridade:

A ideia é **tirar proveito** do referencial móvel.



Caso #3

Se (idade(cauda) < idade(novo) ≤ idade(refMovel))

Então a posição do novo elemento estará entre *cauda* e *refMovel*

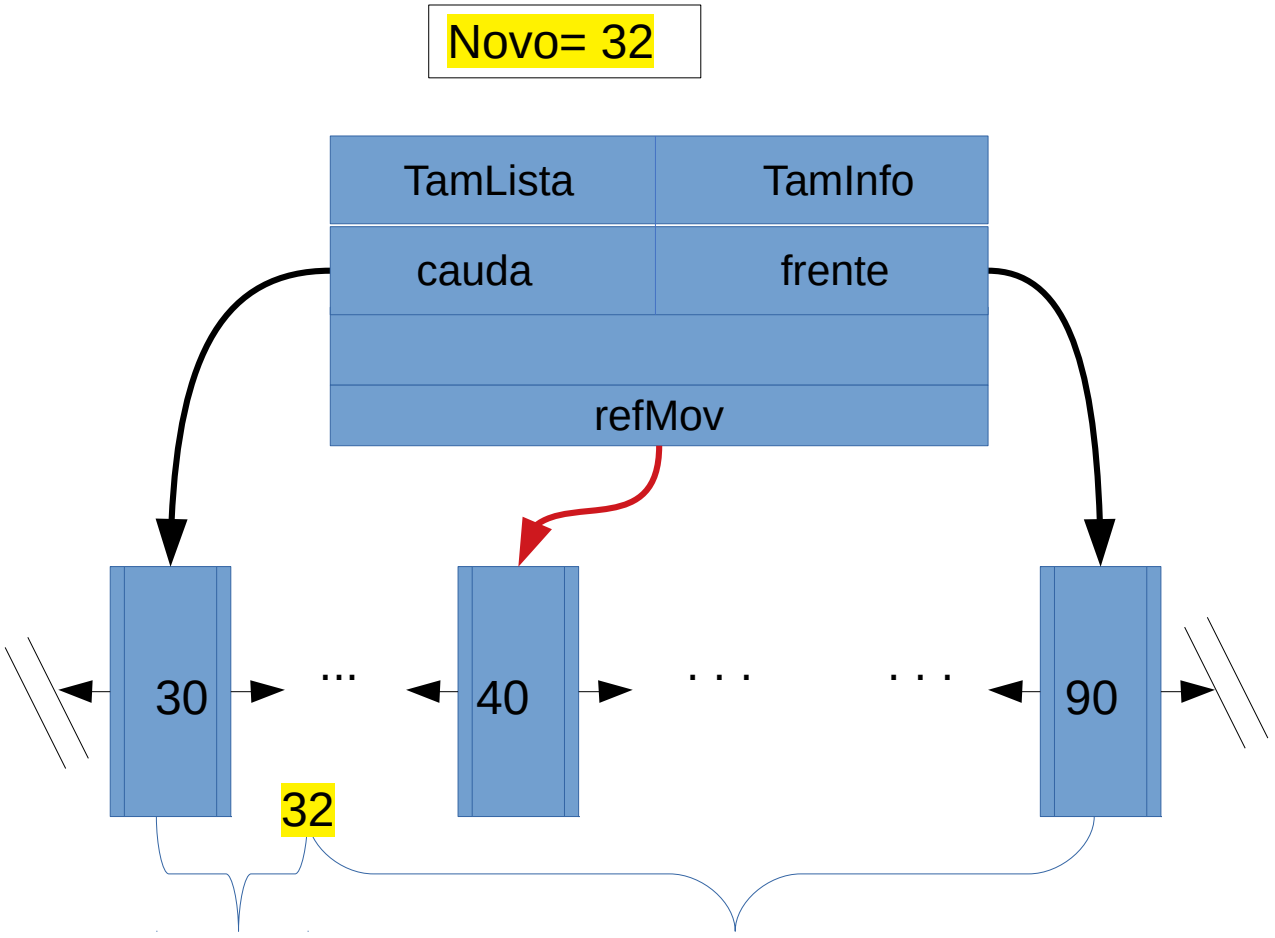
Caso #4

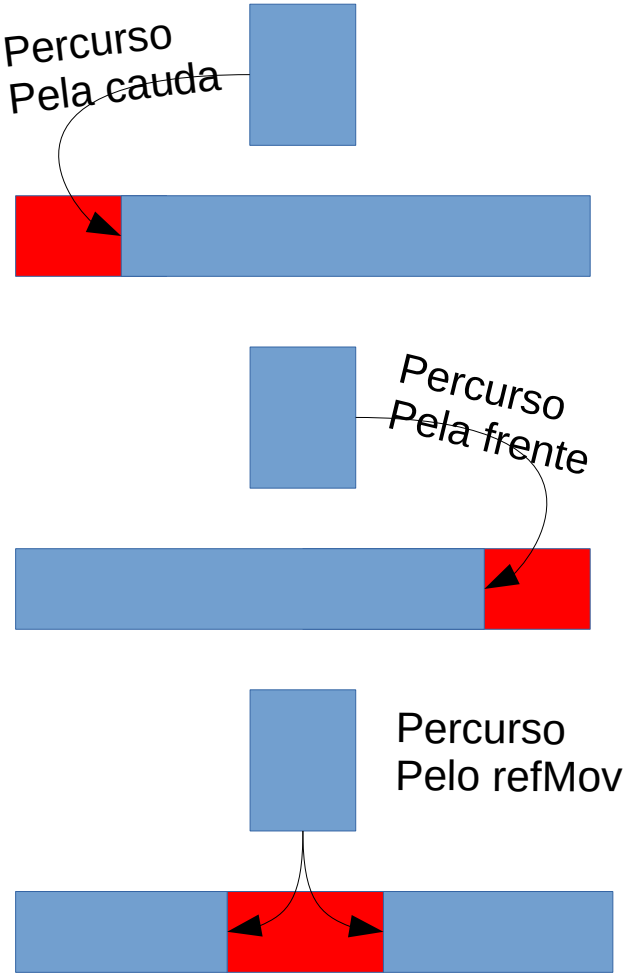
Se (idade(refMovel) < idade(novo) ≤ idade(frente))

Então a posição do novo elemento estará entre *refMovel* e *frente*

Explorando a vantagem do RefMovel na FDE de prioridade:

Aprimorando um pouco mais: o caminho de busca é determinado pela proximidade entre a chave de prioridade do *novo* e a chave de prioridade de algum dos referenciais (*cauda*, *refMov* ou *frente*).

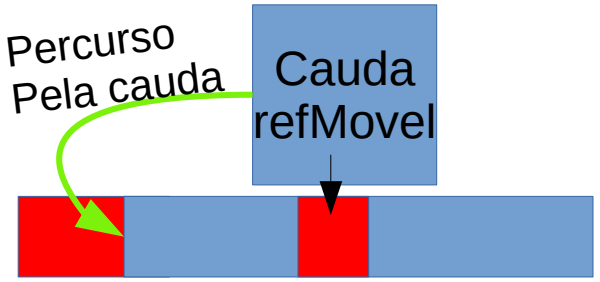




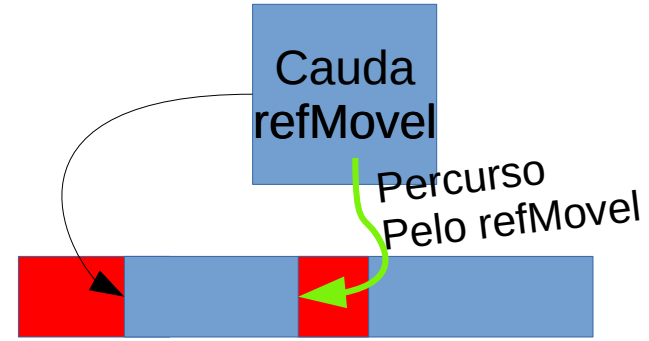
Explorando a vantagem do RefMovel na FDE de prioridade:

Caso#3:
Se (idade(cauda) < idade(novo) ≤ idade(refMovel))
Δ_a = | idade(cauda) – idade(novo) |
Δ_b = | idade(refMovel) – idade(novo) |

Se $\Delta_a < \Delta_b$: localize a posição do novo pela cauda



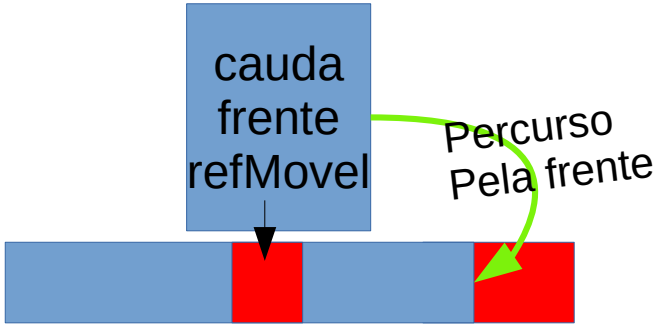
Senão:



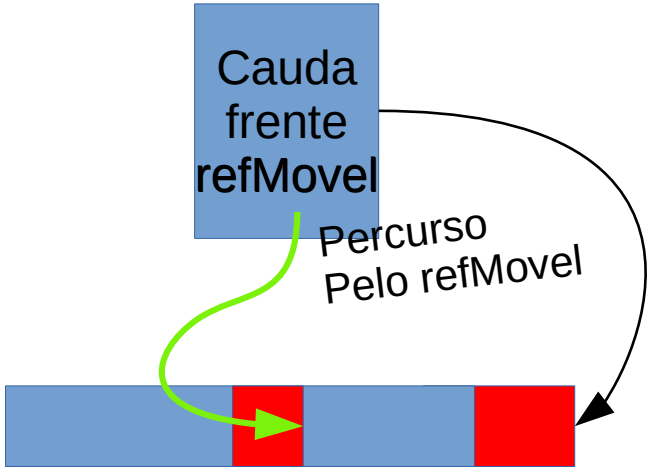
Explorando a vantagem do RefMovel na FDE de prioridade:

Caso#4:
Se (idade(refMovel) < idade(novo) ≤ idade(frente))
Δ_c = | idade(frente) – idade(novo) |
Δ_d = | idade(refMovel) – idade(novo) |

Se $\Delta_c < \Delta_d$: localize a posição do novo pela frente



Senão:



Utilize o código da FDE de prioridade discutido em sala e, junto com o seu colega de equipe, busque projetar as operações básicas sobre a FDE de prioridade com referencial móvel.

Considere que o referencial móvel (*refMovel*):

- É igual a Null para a fila vazia ou...
- Aponta para o endereço do elemento mais recentemente inserido.

O que acontece com a atualização do *refMovel* se ocorrer alternadamente: a inserção de um novo item mais idoso que todos seguida da inserção de um novo item mais jovem do que todos (ou vice-versa)? Esse seria um pior caso para a inserção em questão?

A próxima tarefa versará sobre uma FDE de prioridade com referencial móvel.

O modelo (FDE de prioridade) foi tratado em sala e é descrito no pdf publicado no Moodle.